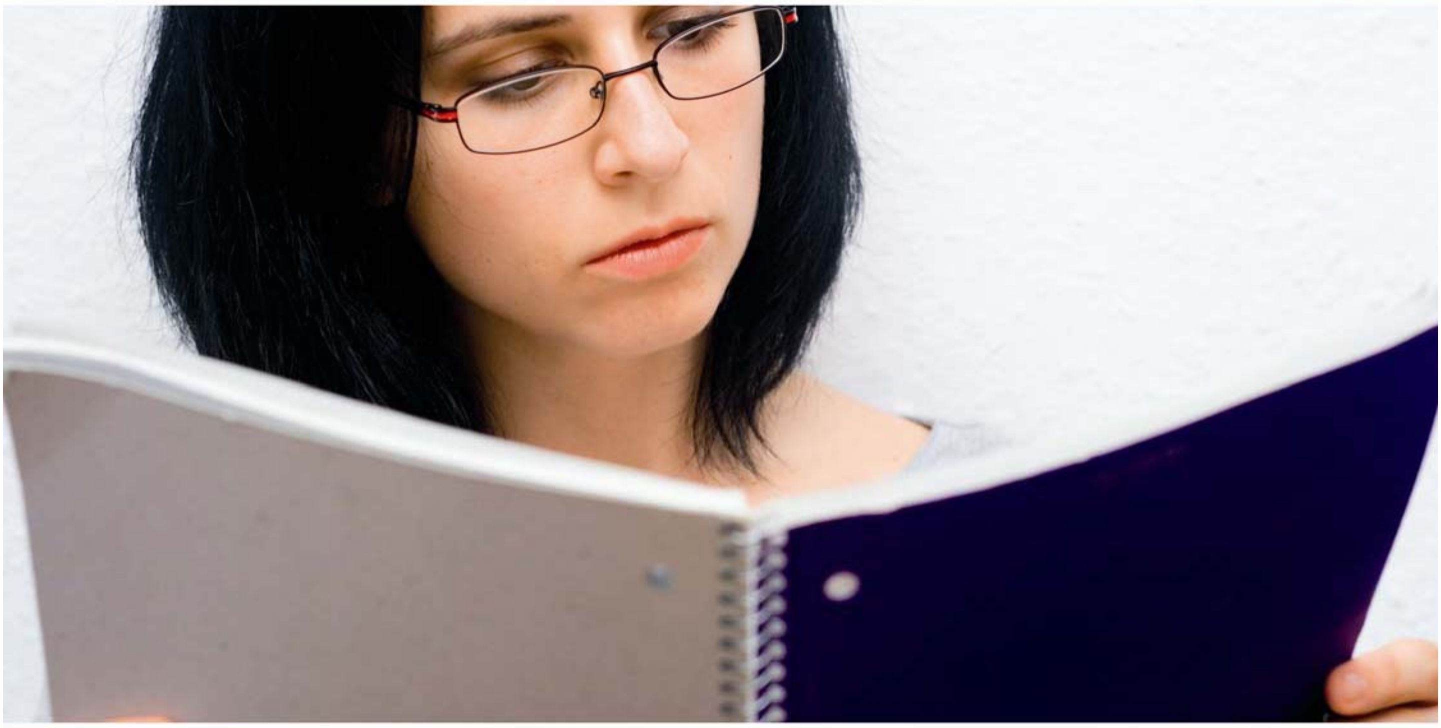
Supporting achievement

Exemplar work

GCE Computing COMP4



Project 9

Please note: every effort has been made to respect the right of anonymity of the source of this coursework and of those who have contributed to it, without compromising the work or invalidating it for the purpose to which it is to put. Some personal details of contributors may therefore remain. We ask the reader to respect their right to privacy.

		AQA COMP4 COMMENTARY SHEET v4.0			
Level of Complexity:		**A Complex Project**			
EXEMPLAR PROJECT No.		**PROJECT 9**			
Programming Language:		**VB6 - Fully Coded**			
Title/Type of System:		**CAL - Vectors Solved**			
Analysis	**Band**	**Comments**	**Page**	**Mark**	**Max**
Background & Problem identification					
Description of current system					
User(s) identification					
User needs & limitations					
Data source(s) and destination(s)					
Data volumes					
Analysis data dictionary					
DFDs existing & proposed					
Objectives					
Complexity					
Potential solutions					
Proposed solution					
Use of formal method(s)					
E-R model / objects etc (if appropriate)					**12**
Design					
Overall system design					
Modular structure					
Record or d'base structure					
Validation incl errors					
File org & processing					
D'base design + E-R model					
Storage media & format					
Algs for data transformation					
User interface (I/O)					
Security & integrity of data					
System security					
Test strategy					**12**
Technical Solution					
Technical competence					
Complex tasks					
Annotated "listing"					
Samples of screens and design views					
Customisation					**20**
System Testing		**THIS SECTION NOT RELATED TO COMPLEXITY**			
Design of test plan					
Minimal test data cross referenced					
Typical data					
Erroneous data					
Extreme (boundary) data					
Annotated results for above					**8**
System Maintenance					
System overview					
Algorithms or alternative					
Annotated listings/screens					
Proc & VAR lists (prog)					
Forms, macros etc (package)					**7**
User Manual + QWC		**FULL USER MANUAL REQUIRED; NOT JUST ONE SECTION OF SYSTEM**			
Contents page					
Introduction & how to install					
Screen displays explained					
Error handling					
Appropriate level for user					
Enables easy use of system					
Quality of Written Communication					**10**
Appraisal					
Objectives evaluated					
Further development?					**6**
User feedback & analysis of this					
					75
Total					
NB: some of these sections will NOT be appropriate for certain types of project, especially non-Data Processing ones. Not all of them need to be completed. In all cases, you MUST refer to the full assessment criteria as given in the specification, if using this grid.					

Copyright © 2011 AQA and its licensors. All rights reserved.

AQA

Centre-assessed work
Project log
2012 (version 1.1)

A-level Computing (2510)
Unit 4 The Computing Practical Project (COMP4)

Centre number

Centre name

Candidate's full name
PROJECT 9

Candidate number

Section 1 - The Project

To be completed by the candidate and returned to the teacher for approval before the project is started

Project title VECTORS SOLVED

Outline description

IT IS A PROGRAM USED TOHELP STUDENTS' LEARNING OF THE C4 TOPIC VECTORS, IT ENABLES THEM TO ANSWER QUESTIONS ON VECTOR GRAPHS AND PRINT HARD COPIES OF RESULTS.

Section 2 - Project development

To be completed by the candidate and teacher.

The candidate **(C)** and the teacher **(T)** should indicate which items are present in each section by selecting/ticking the appropriate boxes, providing the related page reference

Analysis	**C**	**T**	**Page**
Background to/identification of problem	□	□	
Description of the current system	□	□	
Identification of the prospective user(s)	□	□	
Identification of user needs and acceptable limitations	□	□	
Data source(s) and destination(s)	□	□	
Data volumes	□	□	
Analysis Data Dictionary	□	□	
Data flow diagrams (DFDs) (existing and proposed systems)	□	□	
Entity-relationship (E-R) model (if appropriate), E-R diagrams, entity descriptions	□	□	
Object analysis diagrams - inheritance, aggregation (if appropriate)	□	□	
Numbered general and specific objectives of the project	□	□	

Candidate's full name

PROJECT 9

Candidate number

Analysis (continued)	C	T	Page
Realistic appraisal of the feasibility of potential solutions	□	□	
Justification of chosen solution	□	□	
Evidence of use of appropriate analysis techniques	□	□	

Comment	Maximum mark	Mark awarded
	12	

Design	C	T	Page
Overall system design	□	□	
Description of modular structure of system	□	□	
Definition of data requirements (Design Data Dictionary)	□	□	
Description of record structure (if appropriate)	□	□	
Validation required	□	□	
File organisation and processing (if appropriate) or database design including normalised relations (if appropriate)	□	□	
Sample of planned SQL queries (if appropriate)	□	□	
Identification of storage media	□	□	
Identification of suitable algorithms for data transformation, pseudocode of these algorithms	□	□	
Class definitions (diagrams) and details of object behaviours and methods (if appropriate)	□	□	
User interface design (HCI) rationale	□	□	
UI sample of planned data capture and entry designs	□	□	
UI sample of planned valid output designs	□	□	
Description of measures planned for security and integrity of data	□	□	
Description of measures planned for system security	□	□	
Overall test strategy	□	□	

Comment	Maximum mark	Mark awarded
	12	

Candidate's full name PROJECT 9	Candidate number

Technical Solution
Evidence for this section may come from copies of code listings in the appendix and/or details of software tailoring included in the systems maintenance section. It is not expected that candidates will supply multiple copies of listings, systems or algorithm design documentation.

Comment	Maximum mark	Mark awarded
	20	

System testing	C	T	Page
Design of test plan	□	□	
A minimal set of test data	□	□	
Expected results for typical, erroneous and extreme (boundary) data where appropriate	□	□	
Annotated hard copy of samples of actual test runs	□	□	
Cross-referenced to the test plan	□	□	

Comment	Maximum mark	Mark awarded
	8	

System maintenance	C	T	Page
System overview	□	□	
A sample of detailed algorithm design	□	□	
Procedure and variable lists/descriptions for programs OR list of package items developed	□	□	
Annotated listings of program code/macro code and tailoring	□	□	

Comment	Maximum mark	Mark awarded
	7	

Candidate's full name

PROJECT 9

Candidate number

User manual, including Quality of Written Communication	C	T	Page
A brief introduction and installation instructions	□	□	
Detailed description of the use of the full system	□	□	
Samples of actual screen displays in situ	□	□	
Samples of error messages and error recovery procedures	□	□	
Quality of Written Communication	□	□	

Comment	Maximum mark	Mark awarded
	10	

Appraisal	C	T	Page
Comparison of project performance against numbered general and specific objectives	□	□	
User feedback authenticated by assessor	□	□	
Analysis of user feedback	□	□	
Possible extensions	□	□	

Comment	Maximum mark	Mark awarded
	6	

This form should be attached to the candidate's work and retained at the centre or sent to the moderator as required.

To see how AQA complies with the Data Protection Act 1988 please see our Privacy Statement at aqa.org.uk

Copyright © 2011 AQA and its licensors. All rights reserved.

The Assessment and Qualifications Alliance (AQA) is a company limited by guarantee registered in England and Wales (company number 3644723). Registered address: AQA, Devas Street, Manchester M15 6EX

AQA

Centre-assessed work
Candidate record form
2012 (version 1.2)

A-level Computing (2510)
Unit 4 The Computing Practical Project (COMP4)

Centre number ________ Centre name ________

Candidate's full name PROJECT 9

Candidate number ________

Notice to candidate The work you submit for assessment must be your own. If you copy from someone else or allow another candidate to copy from you, or if you cheat in any other way, you may be disqualified.

To be completed by the candidate

1. Have you received any help or information from anyone other than your subject teacher(s) in the production of this work?

Yes □ No ☒

2. If you have answered yes, give details below and on a separate sheet if necessary.

3. Any books, leaflets or other materials (eg DVDs, software packages, Internet information) used to help you complete this work and not clearly acknowledged in the work itself must be listed below. Presenting materials copied from books or other sources without acknowledgement will be regarded as deliberate deception.

Candidate declaration I have read and understood the above and can confirm that I have produced the attached work without assistance other than that which is acceptable under the scheme of assessment.

As part of AQA's commitment to assist students, AQA may make your coursework available on a strictly anonymous basis to teachers, examining staff and students in paper form or electronically, through the Internet or other means, for the purpose of indicating a typical mark or for other educational purposes. In the unlikely event that your coursework is made available for the purposes stated above, you may object to this at any time and we will remove the work on reasonable notice. If you have any concerns, please contact AQA.

PROJECT 9

Date ________

Teacher declaration I confirm that the candidate's work was conducted under the conditions laid out by the specification. I have authenticated the candidate's work and am satisfied that to the best of my knowledge the work produced is solely that of the candidate.

Teacher signature

Date ________

This form should be attached to the candidate's work and retained at the centre or sent to the moderator as required.

Turn over ▸

Candidate's full name

PROJECT 9

Candidate number

Section A – Project background

To be completed by the candidate and returned to the teacher for approval before the project is started

Project title VECTORS SOLVED

Implementation language and/or proposed packages to be used VB6

Name of End User and key role (ie job title) in relation to the project

Section B – Summary of marks

To be completed by the teacher

Mark scheme	**Maximum mark**	**Mark awarded**
Analysis	12	
Design	12	
Technical solution	20	
System testing	8	
System maintenance	7	
User manual, including Quality of Written Communication	10	
Appraisal	6	
Total	**75**	

Concluding comments

I regard this project as being complex/ of adequate complexity/ of limited complexity/ simple[1]. My reasons for this decision are given briefly below. (You should refer to the *'Definitions of problem types for projects'* document in the Teacher Resource Bank.)

If you have reason to doubt the authenticity or completeness of acknowledgement, or where there has been little or no opportunity for personal supervision, you must take steps (for example an interview) to confirm that the candidate has done the work and is the author of the project. The steps taken and their outcome must be given on a separate sheet, signed by you and attached to this Record Form.

Details of additional assistance given (if any) Record here details of any assistance given to this candidate which is beyond that given to the class as a whole and beyond that described in the specification. Continue on a separate sheet if necessary.

[1] *Please delete whichever does not apply.*

To see how AQA complies with the Data Protection Act 1988 please see our Privacy Statement at aqa.org.uk

Copyright © 2011 AQA and its licensors. All rights reserved.

The Assessment and Qualifications Alliance (AQA) is a company limited by guarantee registered in England and Wales (company number 3644723). Registered address: AQA, Devas Street, Manchester M15 6EX

AQA – COMP4

Candidate Name :	[redacted]
Candidate Number :	[redacted]
Marker :	J [redacted]

General Comments

A high standard project which has been conducted with continued consultation of the end user.

Total Mark	[redacted] / 75

Analysis [redacted]	**Max** 12

Comments

There is evidence that the student has analysed a complex problem which involves creating a computer aided learning package for use by A2 Level Maths students, in order to help understand vectors in the Core 4 module (p12 -13). The client/user is the pupil's maths teacher; the student has evidence of an interview p10 but did spend time discussing the various requirements. The candidate also discussed preferences with pupils p3. There is a realistic appreciation of system requirements and the student has clearly interviewed the user Mr Mason and talked to fellow pupils in order to establish them. All sections of the analysis are covered and it is well structured.

Design [redacted]	**Max** 12

Comments

The student demonstrates clearly that they grasp the overall design of the system and has used a range of tools to describe the modular structure of the system. There is evidence that a sensible HCI rationale has been considered for both input screens and printed reports p21 -26. The student has discussed data storage and file organisation at a level which is appropriate for this application (ie. stores very little as the main application is the simulation of the vectors). Algorithms for data transformation are given in pseudocode as well as explanation of their purpose p31 – 35. Use of recursion is evidenced showing the complexity of the proposed solution p32. Test strategy is sensible.

Technical Solution [redacted]	**Max** 20

Comments

The student has used VB6 to produce a robust, fully working solution. The code is well annotated and sensible features of the language haven been utilised especially the graphics capabilities. The program is generally well structured and uses a range of procedures and functions and employs sensible use of parameter passing when appropriate. There is also the use of recursion to achieve the animation of the vectors and some basic file handling has been used to store the student data. Screen shots help illustrate the look and feel of the system as well as the functionality. There has been a real attempt by the student to produce an effective, user-friendly solution which meets the original objectives of a fully analysed complex problem.

System Testing	**Max** 8

Comments

The candidate has produced a well-designed test plan and has employed a sensible approach to testing as is appropriate to the problem. Test data has been utilised to ensure that the vectors are plotted as expected (p 60, 62, 67 -70) and there is a good range of cross-referenced evidence provided to prove the reliability of the solution. There has been some attempt to provoke error and amendments to the code have been carried out and properly explained.

Systems Maintenance	**Max** 7

Comments

A real attempt has been made to explain the technicalities of the system, especially the difficult to understand parts of the code. The procedure and variable lists are comprehensive and the program listing is self-documenting and well annotated and easy to follow. The candidate has also detailed the addition of a new algorithm which has changed slightly from the design stage.

User Manual	**Max** 7

Comments

A well presented user manual, appropriate to the user.

Appraisal	**Max** 6

Comments

The student has gained real feedback from her maths teacher as well as from some of the pupils. She has given a reasoned and detailed explanation of how the SMART objectives have been met and where there is room for improvement/extensions and takes on board the comments from the user.

Quality of Comm	**Max** 3

Comments

The project has been well written and presented with some features of word-processing exploited.

Contents

Analysis

Analysis

Identification

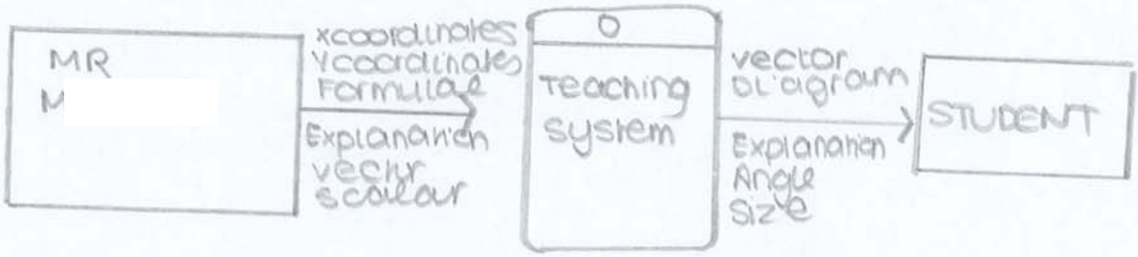
The [redacted] School, located in Newcastle [redacted], is comprised of numerous subject related departments; the maths department within the school is one of the largest; with pupils being taught from Year 7 – Year 13, meaning that teachers constantly have so much teaching to carry out; notes to photocopy, tests to mark, questions to set and through all of this to make sure that the pupils they are teaching have a strong understanding of the mathematical based problems they are given in order to pass their exams and achieve their desired grades.

After studying A-Level Maths last year my maths teacher, who was aware I was studying computing, Mr M[redacted] recently approached me to discuss ways in which he thinks he could improve the teaching for the students; knowing that I was both a computing & a maths student he thought this may be of interest to me. He believes that the way forward is to create an interactive learning environment that will help teach the students on one of the most complex areas of the Core 4 maths syllabus, vectors, which are most problematic for teachers to teach as it involves numerous diagrams that need to be drawn by teachers in order to provide students with a clear understanding of what is occurring within each question. Mr M[redacted] thinks that an interactive teaching method will help to cut teaching time on this difficult topic and also allow students to create a firmer grasp on their understanding of vectors.

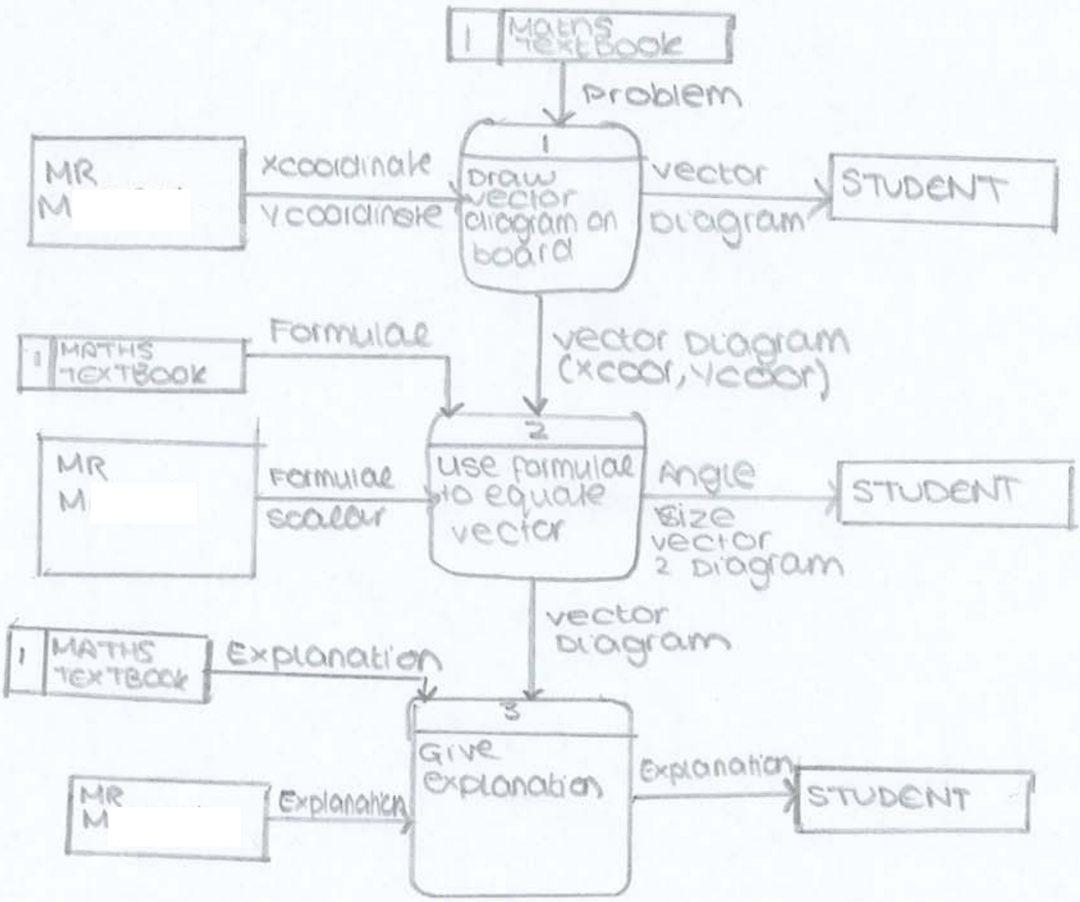
Description of the Current System.

At the moment Mr M[redacted] teaches his students by copying notes and questions out of the Edexcel A Level Core 4 maths book, he then draws diagrams on the whiteboard or on pieces of paper and hands them out to students for them to appreciate & interpret the vector problems. However this is extremely time consuming and means Mr M[redacted] constantly has to refer to the text book whilst trying to explain the concept at the same time; this means that sometimes the problem can sound more confusing than it actually is and cause students to not fully appreciate the concept they are trying to learn.
Mr M[redacted] hopes that the new interactive system will be able to produce clearer diagrams and explanations of vector problems, as he believes that sometimes his drawings can be messy and that students don't fully understand what is going on.

Data Flow Diagram of the Current System
Context



Level 1



Analysis

Before I obtained the objectives of what Mr M[redacted] required for the system I felt that it was important to talk to the students who are taking A Level Maths and have them fill in a questionnaire to find out what they are finding difficult in the C4 Vectors topic. Below is a sample of one the questionnaires I obtained from a pupil:

1) When studying vector problems what are the 3 things you find most difficult about them?

ANGLES AND CALCULATION OF THEM | DISTANCE BETWEEN 2 VECTORS | PARALLEL VECTORS

2) What about the current system you have of learning about vectors do you think could be improved?

WOULD LIKE IT TO BE MORE FUN AND INTERACTIVE

3) What do you think would help to facilitate additional learning of vector problems?

AN INTERACTIVE SYSTEM OF ANSWERING QUESTIONS

4) What 3 topics on vectors would you most like to see featured on a Computer Aided Learning Package?

AS NO. 1.

5) Do you feel that learning on a computer would be more fun and beneficial than hand written work on vectors?

YES

6)When using a system to learn about vectors would you like it if there was also questions involved for you to answer?

DEFINITELY AS I THINK IT WOULD HELP MY LEARNING

Jat 15/09/10

Users of the new system

Of course the new system will benefit Mr M[redacted] and his methods of teaching, but the main users of the new interactive system will be the A Level pupils of Mr M[redacted]'s class, he has at least 20 pupils per class apart from his Further Maths class which is smaller and consists of only 10 pupils.

All students will potentially use the system and it will therefore have to be clear and easy for everyone to understand and use or else it will just further complicate the problems of teaching the topic of teaching vectors.

Mr M[redacted] himself will not be using the system in itself in terms of the problems pages, but will oversee it as I implement it so the correct topics are covered for the students; however he will be using the Main Menu form in order to access the command button that will enable him to print out a hard copy of the students' results.

User Needs

The interactive learning system will need to be available on every computer throughout the school so the pupils have access to it at any time and the pupils will be provided with a copy of it to place on their memory stick so they can use the program at home for homework and revision purposes.

It will incorporate the topics of chapter 5 in the Core 4 maths text book, vectors, it will need to include diagrams, explanations but also an element of colour etc. to make sure that it doesn't seem too drab and boring to put the students off, to make sure the students actually interact within the system they will have the opportunity to enter the vector coordinates of their chosen question & the opportunity to input their answer , to see how they would actually go about working out the correct answer & if their answer is correct.

General Objectives

Mr M[redacted] has asked me if it would be possible for me to produce a Computer Aided Learning package which will help A Level maths students to learn and interact about vector problems, located in Chapter 5 of the C4 A Level textbook. The package that I will create should allow students to not only see their vectors being drawn out and explained but also provide them with the opportunity to answer questions and obtain feedback on their problems. The system should also be able to print hard copies of both the vector graphs/explanations and the results of the students' scores.

Objectives for the proposed interactive system

1. The learning system must cover the 3 most difficult topics of vectors: working out if vectors are parallel to one another, working out angles between vectors and working out the distance between non-parallel vectors.
2. The proposed system must include diagrams of the vectors drawn in real time along with an explanation of what is occurring to help the students.
3. The proposed system must then produce transformation in real time of these vectors for the topic of if vectors are parallel to one another.
4. The proposed system must also draw a line between the 2 non-parallel vectors to show students what distance the computer is working out for them.
5. The student will be able to input the coordinates of the vectors, the computer will then draw these out and show the student how they would work out the given problems.
6. The system must be easy to navigate around and will consist of 3 main sections for

each of the 3 main topics.

7. Colour must be incorporated to make sure that the system isn't too mundane, as vectors aren't particularly a fun subject.
8. Have an option for students to print out certain vector diagrams/explanations.
9. For the student be able to quit the system at any time if they do not wish to continue.
10. For the student to be able to input their answer to the question and be provided feedback if their answer is wrong/right.
11. For the scores the students score on the program to be stored to a small file so Mr M[redacted] can see how each pupil is performing on the different topics.
12. For Mr M[redacted] to able to obtain a hard copy of the student's results.

Limitations

1. The system will not be able to incorporate every topic of vectors within Chapter 5 of C4, but only the 3 main topics that the students are struggling with.
2. The vector coordinates that the students will input in the system will be within a finite range.
3. The vector coordinates that will be drawn by the system will again be within a finite range.

Data Sources & Destinations

After talking this through with Mr M[redacted] I believe the following data will be necessary for the system to work correctly :

Data Source (Input)	Process	User	Output	Destination
Vector Coordinates	Student inputs coordinates	Student	Drawings of vector.	Output as vector on page.
Choose what topic they wish to cover.	Student clicks button on home page to select topic.	Student	Takes the student to the desired form.	Output as page on screen.
Formulae within the system.	Formulae is placed within the system during implementation.	Student	The answer to a given problem once the formulae is applied.	Answer & diagrams on page.
Explanations	Explanations of the given topics are placed within the system during implementation.	Student	Written explanation of questions/answer.	Written output on the page.
Navigational Buttons	Once clicked takes the students to the desired pages, e.g. back etc.	Student	Displays the desired page.	Displayed page.
Student Answers	Student inputs their answer	Student	Shown if wrong or right and result written to file.	Text File

Student Name	Student inputs their name	Student	Student Name written to file	Text file

Analysis Data Dictionary & Volumes

The formulae that I will be using within the system will consist of 3 main formulas :

Calculate If Vectors are parallel
$a+b=x(a+b)$ where x is a non-zero scalar

Calculate the angle between 2 vectors
$COSAOB = a.b / |a| |b|$

Calculate the distance between 2 non parallel vector points
The distance between the points (x1, y1) and (x2,y2) =
$\surd\,(x1-x2)^2 +(y1-y2)^2$

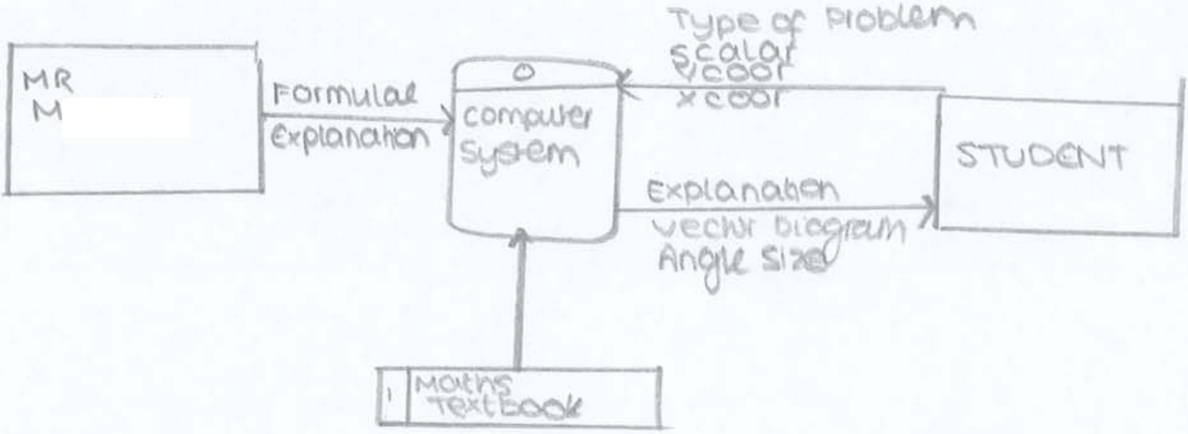
Variable Name	Data Type	Format/Size (Bytes)	Description
XCoordinate1 (i)	Integer	4	Will vary between -15 and 15 on the x and y axis.
YCoordinate1 (j)	Integer	4	Will vary between -15 and 15 on the x and y axis.
Xcoordinate2 (i)	Integer	4	Will be used when working out distance between 2 non-parallel vector points. Again it will vary between -15 and 15.
Ycoordinate2 (j)	Integer	4	Will be used when working out distance between 2 non-parallel vector points. Again it will vary between -15 and 15.
Explanation	String	A few sentences to explain the problem and solution. About 500 bytes.	To explain to the student the problem and how they would go about solving it.
Formulae	String	200 bytes.	Formulae that will be displayed to show the students what they would need to use.
Scalar	Integer	4	This will be used when

			vectors are being tested to see if they are parallel and will be multiplied by a certain vector.
Angle Size	Integer	5	This will display the given angle size of two vectors using the specified formulae.
Student Answer	Integer	4	Will be inputted by the student to compare to the actual answer & result written to file
Student Name	String	20	Will be inputted by the student & written to file along with results.

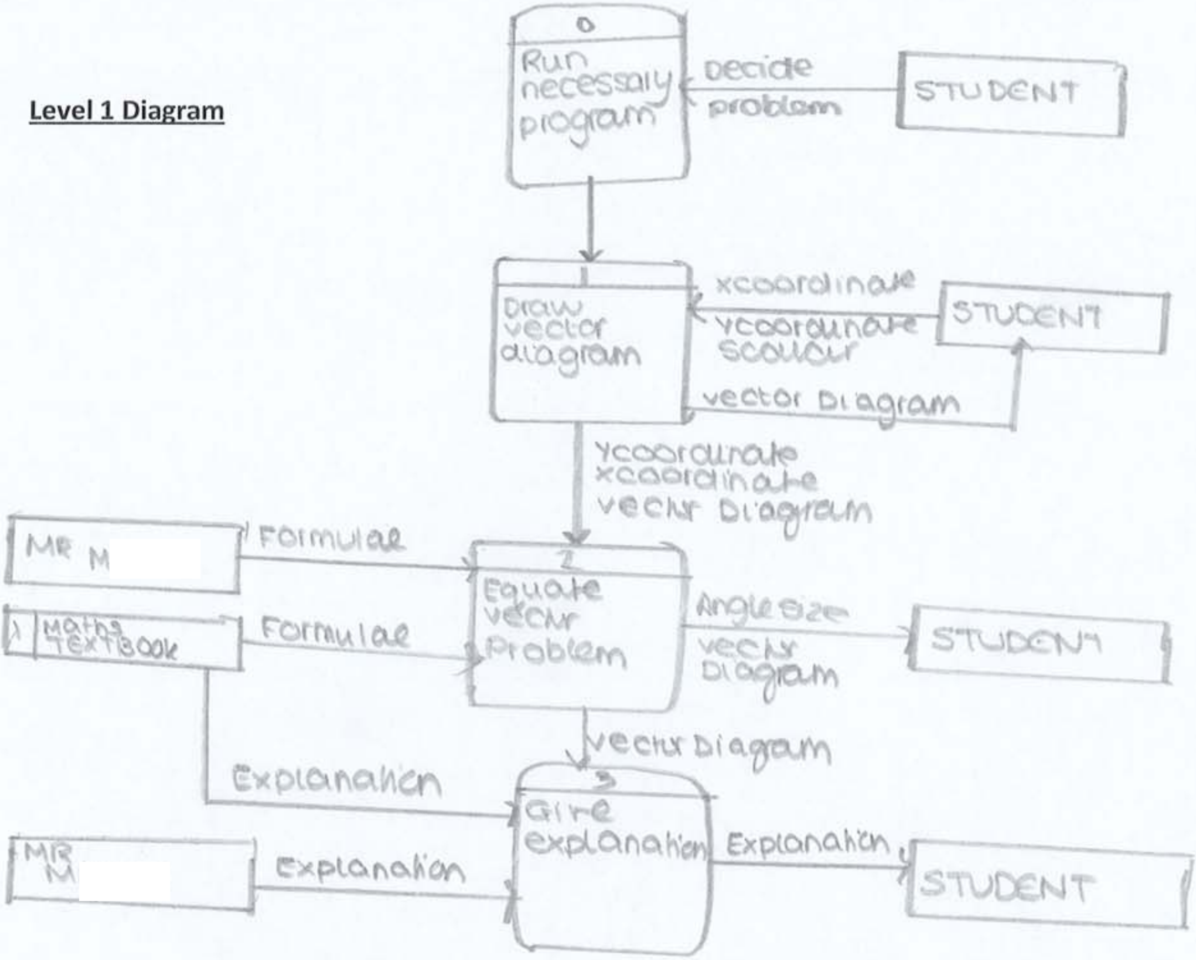
Of course there won't be large amounts of data constantly being entered into the system, all that will be input by the student is the coordinates of the vectors they wish to see, their answer & their name. This means that the program won't have to handle large amounts of data being input into it but will have to be able to create drawings, graphs etc at run time.

Data Flow Diagram of the Proposed System

Context Diagram



Level 1 Diagram



Realistic Appraisal of the Feasibility of Potential Solutions

Manual Solution

It would be extremely difficult for Mr M[redacted] to carry on drawing and teaching multiple vector problems; it is extremely time consuming and results in him having to photocopy numerous notes and draw various vector diagrams on the whiteboard throughout his lessons. This takes up valuable teaching time that could be better spent and doesn't provide pupils with a clear understanding of the topic due to the fact that the diagrams can often become unclear and confusing and once removed from the whiteboard the student has nothing to refer back to and often misplaces their notes.

Off the Shelf Packages

I have not yet come across a computer software package that solely deals with vector problems at C4 level, meaning that there is a very limited option for students. Although there are other packages that deal with various problems in C4 Mr M[redacted] only wants vectors to be covered as he feels more than able to fully teach other topics that don't require large amounts of time to be spent drawing. Not only this but off the shelf packages are usually relatively expensive and although continually photocopying notes is expensive the school budget cannot really afford to shed out large sums of money to pay for new computer software programs at this given time.

Bespoke Software

As I have stated above the school doesn't have a large amount of money to spend on software at the moment after recently spending large sums of money on a rebuild. Therefore professional bespoke software wouldn't seem a realistic option, although it is catered to the users need it is often very expensive as it is sold only to the one client, not publically. However it makes sense to have bespoke software solution in the form that I would create as it would not only be free but catered towards the end users' needs.

Programmed in Visual Basic

This seems to be the most possible option for the school and Mr M[redacted] at this current time, Visual Basic is one of the simplest of languages to program in for beginner programmers and has a large variety of drawing features that will be able to be used for the vector diagrams. It will not cost the school a large sum of money but at the same time will be able to incorporate everything that Mr M[redacted] requires.

Justification of Chosen Solution

After conducting an interview with Mr M[redacted], which is shown below and analysing the current system I have decided to implement the interactive system in Visual Basic 6. It seems to me that the other possible options were just not plausible for the school at this current time.

The off the shelf and bespoke software packages were simply too expensive or didn't fully cover the topic of vectors in enough depth without addressing other unnecessary topics that Mr M[redacted] didn't want covered. Therefore the most promising solution seems to be Visual Basic 6, which will allow me to program a system that will be able to incorporate various drawings and explanations to help aid the students learning and will specifically be produced for Mr M[redacted]'s requirements. Visual Basic 6 is also extremely flexible allowing both the use of graphics and mathematical formulae which will allow me to represent graphs effectively. Not only this but I also have some previous experience in programming in Visual Basic meaning that I should be able to create a system fairly quickly for Mr M[redacted] and his students During the program implementation I will also be using a piece of software called Corel Paint Shop Pro X to design the splash page for the system as the start up screen.

M

Interview with Mr B[redacted]

What is the current system you use? As with all other topics, I give notes concerning the basic vector methods, and work through several examples on the board to cover exam-type questions.

Are they any more additional problems with your current system?

Well, vectors are 3-Dimensional, which makes realistic diagrams difficult to draw – so most get projected onto 2-Dimensional space.

If a new system was created for you is there anything specific you want to be included? Students need to be able to use the software to enhance their understanding, and to take hard copies of useful examples.

What about the layout, colours etc of the system?

Any use of colour must not distract from the teaching content, rather it should enhance their understanding of the basics

Who exactly will be using the system?

Upper Sixth students doing A2 Mathematics. I would need to trial the system before handing over to them.

How often would you expect students to use the system?

For a couple of weeks in Upper Sixth intensively, and then for it to be available for revision etc.

Oh right, that's fine and do you want any security on the system, to stop other pupils accessing it?

No — most other students would not know what to do with it. They would, however, need to save onto their own user area.

Is there any main topics of vectors you want to cover, it is quite a large topic?

① Modulus of a vector
② Parallel vectors
③ Unit vectors
④ Vector eq^{n} of a line
⑤ Use of column vectors.
⑥ Distance between 2 3D points
⑦ Scalar product of 2 vectors
⑧ Angle between 2 vectors
⑨ Does a point lie on a line?
⑩ Perpendicular vectors.

Ok, sure, no problem, is there anything else I can do for you within the system?

Yes, let me try it out as it develops, and offer mathematical advice

Appendix C4 – Formulae

This appendix lists formulae that candidates are expected to remember and that may not be included in formulae booklets.

Integration

function	integral
$\cos kx$	$\frac{1}{k}\sin kx + c$
$\sin kx$	$-\frac{1}{k}\cos kx + c$
e^{kx}	$\frac{1}{k}\mathrm{e}^{kx} + c$
$\frac{1}{x}$	$\ln\lvert x\rvert + c,\ x \neq 0$
$\mathrm{f}'(x) + \mathrm{g}'(x)$	$\mathrm{f}(x) + \mathrm{g}(x) + c$
$\mathrm{f}'(\mathrm{g}(x))\,\mathrm{g}'(x)$	$\mathrm{f}(\mathrm{g}(x)) + c$

Vectors

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} \bullet \begin{pmatrix} a \\ b \\ c \end{pmatrix} = xa + yb + zc$$

S	Numerical integration of functions.	Application of the trapezium rule to functions covered in C3 and C4. Use of increasing number of trapezia to improve accuracy and estimate error will be required. Questions will not require more than three iterations. Simpson's Rule is *not* required.

6. Vectors

N	Vectors in two and three dimensions.	
N	Magnitude of a vector.	Candidates should be able to find a unit vector in the direction of **a**, and be familiar with $\vert a \vert$.
N	Algebraic operations of vector addition and multiplication by scalars, and their geometrical interpretations.	
N	Position vectors. The distance between two points.	$\overrightarrow{OB} - \overrightarrow{OA} = \overrightarrow{AB} = \mathbf{b} - \mathbf{a}$. The distance d between two points (x_1 , y_1 , z_1) and (x_2 , y_2 , z_2) is given by $d^2 = (x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2$.
N	Vector equations of lines.	To include the forms $\mathbf{r} = \mathbf{a} + t\mathbf{b}$ and $\mathbf{r} = \mathbf{c} + t(\mathbf{d} - \mathbf{c})$. Intersection ,or otherwise, of two lines.
N	The scalar product. Its use for calculating the angle between two lines.	Candidates should know that for $\overrightarrow{OA} = \mathbf{a} = a_1\mathbf{i} + a_2\mathbf{j} + a_3\mathbf{k}$ and $\overrightarrow{OB} = \mathbf{b} = b_1\mathbf{i} + b_2\mathbf{j} + b_3\mathbf{k}$ then $\mathbf{a} \,.\, \mathbf{b} = a_1b_1 + a_2b_2 + a_3b_3$ and $\cos \angle AOB = \dfrac{\mathbf{a}.\mathbf{b}}{\vert\mathbf{a}\vert\vert\mathbf{b}\vert}$. Candidates should know that if $\mathbf{a} \,.\, \mathbf{b} = 0$, and that **a** and **b** are non-zero vectors, then **a** and **b** are perpendicular.

Design

Design

Aim of the Project

The overall aim of the project I will be producing for Mr M will be to produce a computer aided learning system which will help students to learn about various vector problems it will show them if vectors are parallel; the angle size between 2 vectors & the distance between 2 vectors.
I will program it in Visual Basic 6.0 and it will be used by the students of the A Level Maths classes, they will be able to see the transformation of their vectors, the drawn angle sizes & the distance between their vector coordinates -these will all be drawn on a set of axis, the x axis (i) and the y axis (j). ; however they will also have the opportunity to input their answers and be provided with feedback as to if they are right/wrong, these results will then be written to a small text file so Mr M can see how each pupil is performing in the separate sections. The students will have the option to print off each specific page and problem that they carry out and will also be able to clear the vector diagrams from the screen and reset the program in order to enter new vector coordinates. To be able to produce this program I need to produce a thorough design in order to make sure that it is satisfactory and meets all of the necessary objectives.

Overall System Design

Main Menu

INPUT	**PROCESS**
Student Name Choice of Vector Form Option to Print Scores	Student Name Written to File Vector Form Loaded Scores Printed
OUTPUT	**HARD COPY OUTPUT**
Correctly Loaded Vector Form	Report of all Student Scores

Parallel Vectors Form

INPUT	**PROCESS**
Vector Coordinates Answer Scalar Option to print graph	Graph Drawn Answer Calculated Answer compared with Student Answer Results written to file
OUTPUT	**HARD COPY OUTPUT**
Drawn graph Explanation + Formuale Answer Details whether the student is correct	Printed output of graph along with explanation + answer

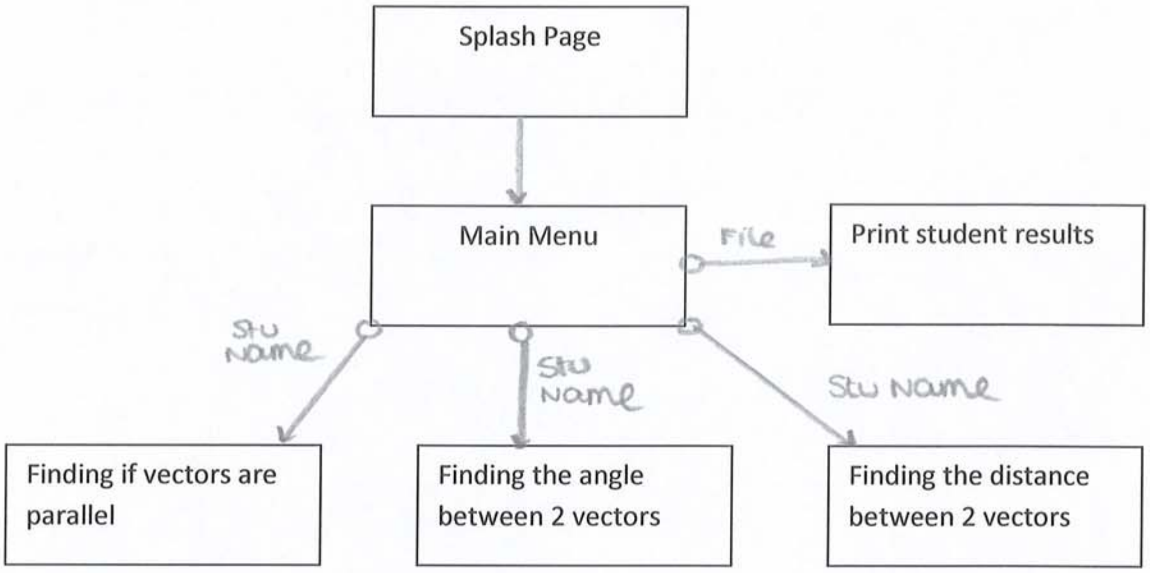
Angle Vectors Form

INPUT Vector Coordinates Answer Option to print graph	**PROCESS** Graph Drawn Angle Size Calculated Answer compared with Student Answer Results written to file
OUTPUT Drawn graph Explanation + Formuale Answer Details whether the student is correct	**HARD COPY OUTPUT** Printed output of graph along with explanation + answer

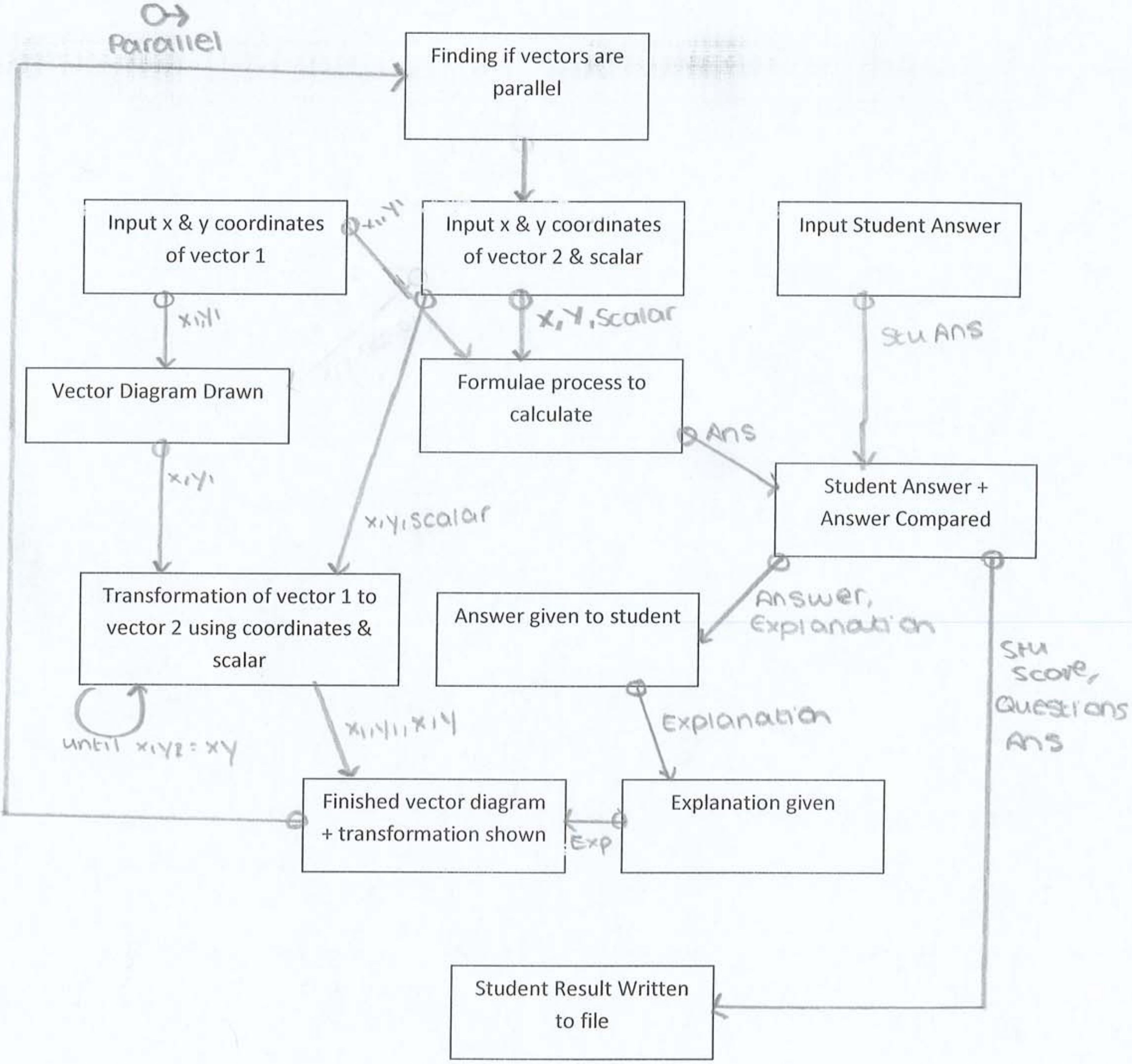
Distance Vectors Form

INPUT Vector Coordinates Answer Scalar Option to print graph	**PROCESS** Graph Drawn Answer Calculated Answer compared with Student Answer Results written to file
OUTPUT Drawn graph Explanation + Formuale Answer Details whether the student is correct	**HARD COPY OUTPUT** Printed output of graph along with explanation + answer

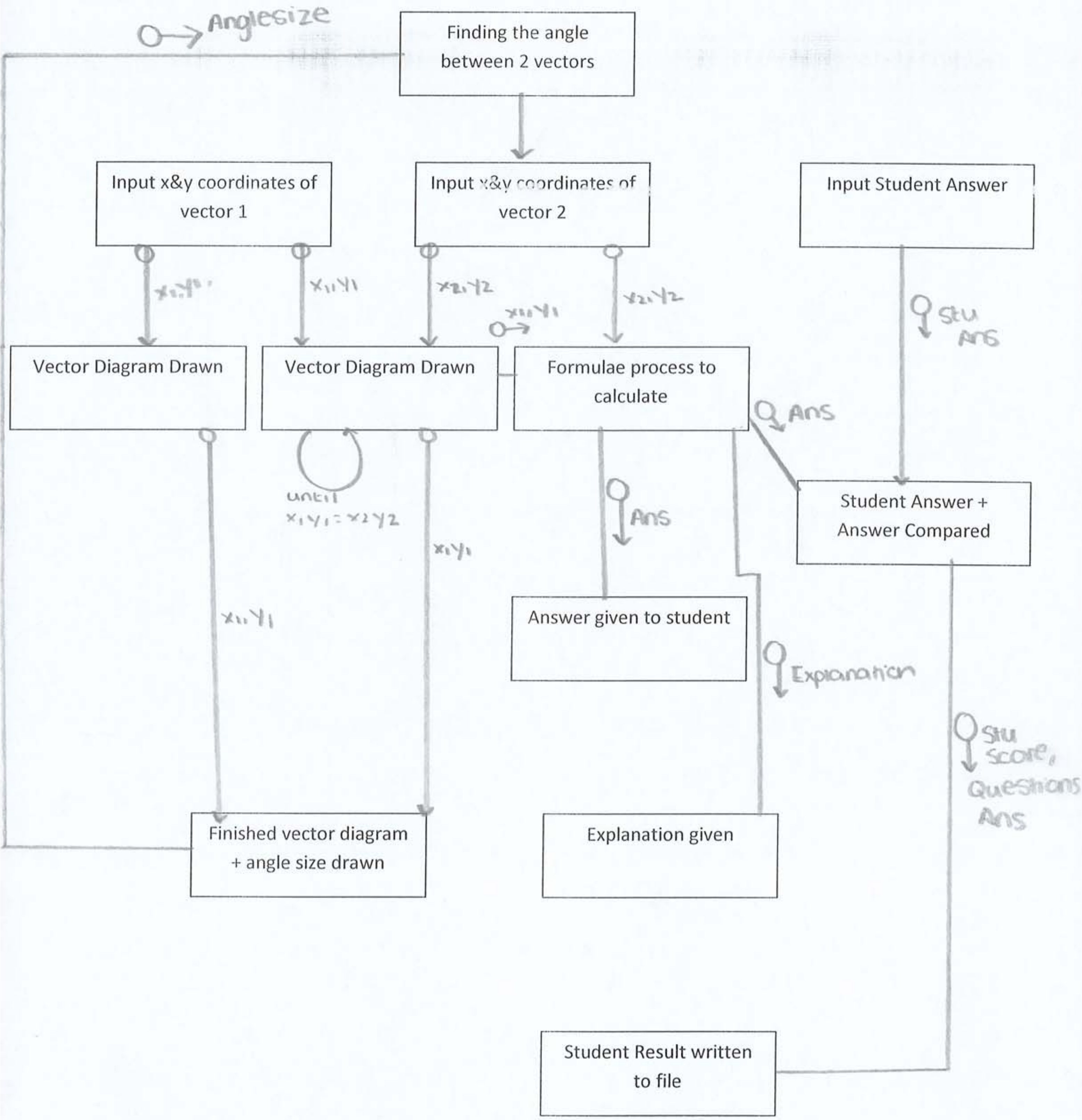
Description of the Modular Structure System


Splash Page
Main Menu
File
Print student results
Stu Name
Stu Name
Stu Name
Finding if vectors are parallel
Finding the angle between 2 vectors
Finding the distance between 2 vectors

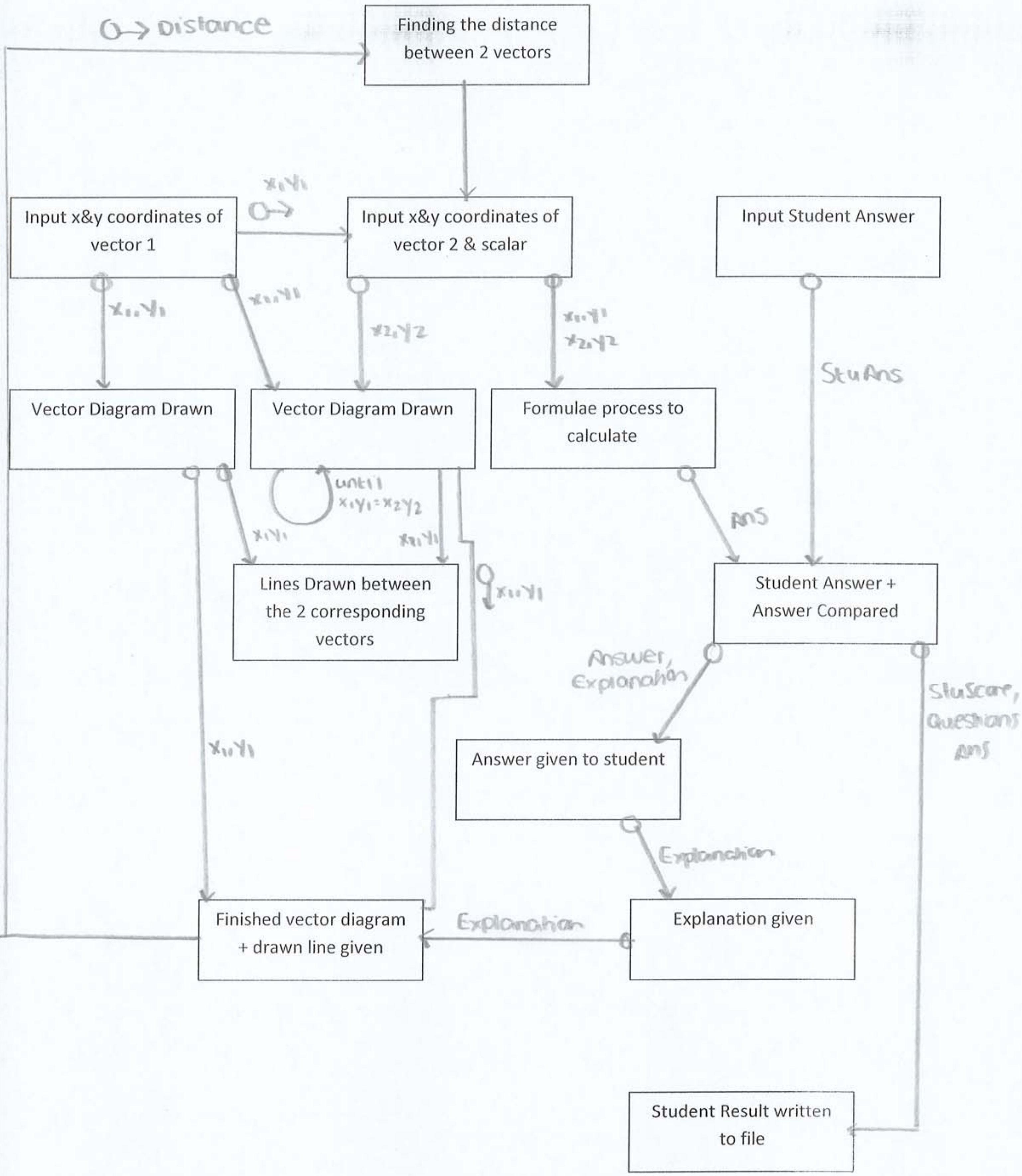
Finding if vectors are parallel


Parallel
Finding if vectors are parallel
Input x & y coordinates of vector 1
Input x & y coordinates of vector 2 & scalar
Input Student Answer
x,y'
x,y,Scalar
Stu Ans
Vector Diagram Drawn
Formulae process to calculate
Ans
x,y'
x,y,scalar
Student Answer + Answer Compared
Transformation of vector 1 to vector 2 using coordinates & scalar
Answer given to student
Answer, Explanation
Stu score, Questions Ans
until x,y2 = xy
x,y,,x,y
Explanation
Finished vector diagram + transformation shown
Exp
Explanation given
Student Result Written to file

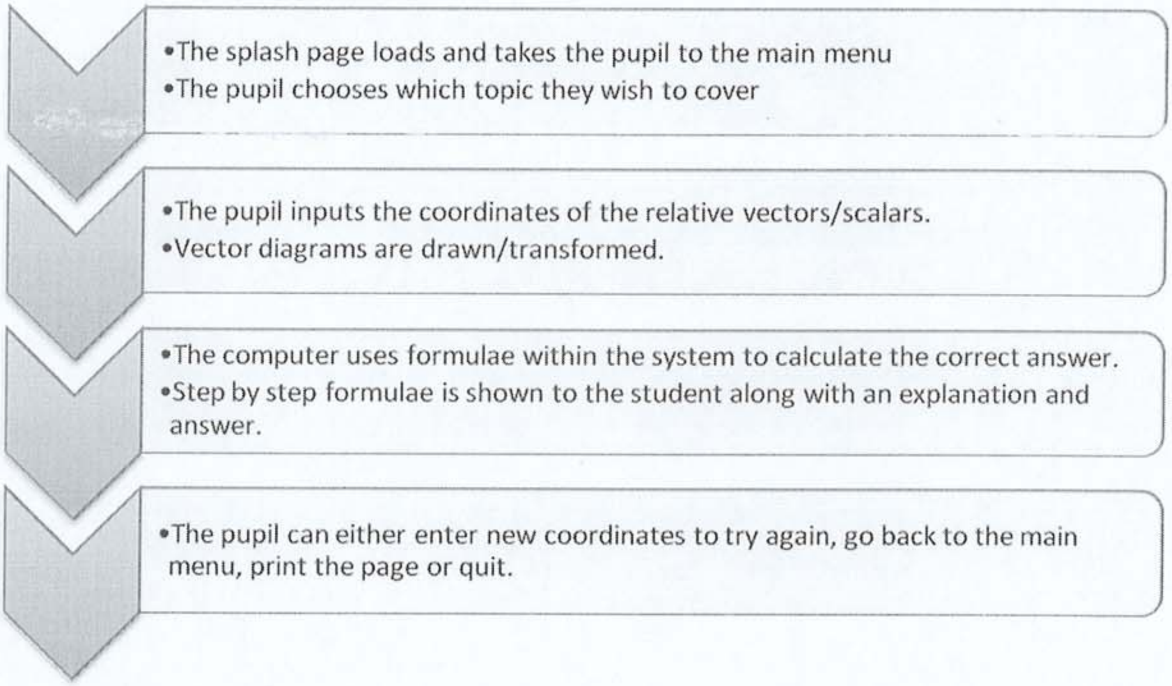
Finding the angle between 2 vectors


Anglesize
Finding the angle between 2 vectors
Input x&y coordinates of vector 1
Input x&y coordinates of vector 2
Input Student Answer
x1,y1
x1,y1
x2,y2
x2,y2
x1,y1
Stu Ans
Vector Diagram Drawn
Vector Diagram Drawn
Formulae process to calculate
Ans
until x1y1=x2y2
Student Answer + Answer Compared
Ans
x1y1
x1,y1
Answer given to student
Explanation
Stu Score, Questions Ans
Finished vector diagram + angle size drawn
Explanation given
Student Result written to file

Finding the distance between 2 vectors


O→ Distance
Finding the distance between 2 vectors
x1,y1
O→
Input x&y coordinates of vector 1
Input x&y coordinates of vector 2 & scalar
Input Student Answer
x1,y1
x1,y1
x2,y2
x1,y1
x2,y2
StuAns
Vector Diagram Drawn
Vector Diagram Drawn
Formulae process to calculate
until x1y1=x2y2
x1y1
x2,y1
x1,y1
Ans
Lines Drawn between the 2 corresponding vectors
Student Answer + Answer Compared
Answer, Explanation
StuScore, Questions Ans
x1,y1
Answer given to student
Explanation
Explanation
Finished vector diagram + drawn line given
Explanation given
Student Result written to file

Decomposition Diagram



- The splash page loads and takes the pupil to the main menu
- The pupil chooses which topic they wish to cover

- The pupil inputs the coordinates of the relative vectors/scalars.
- Vector diagrams are drawn/transformed.

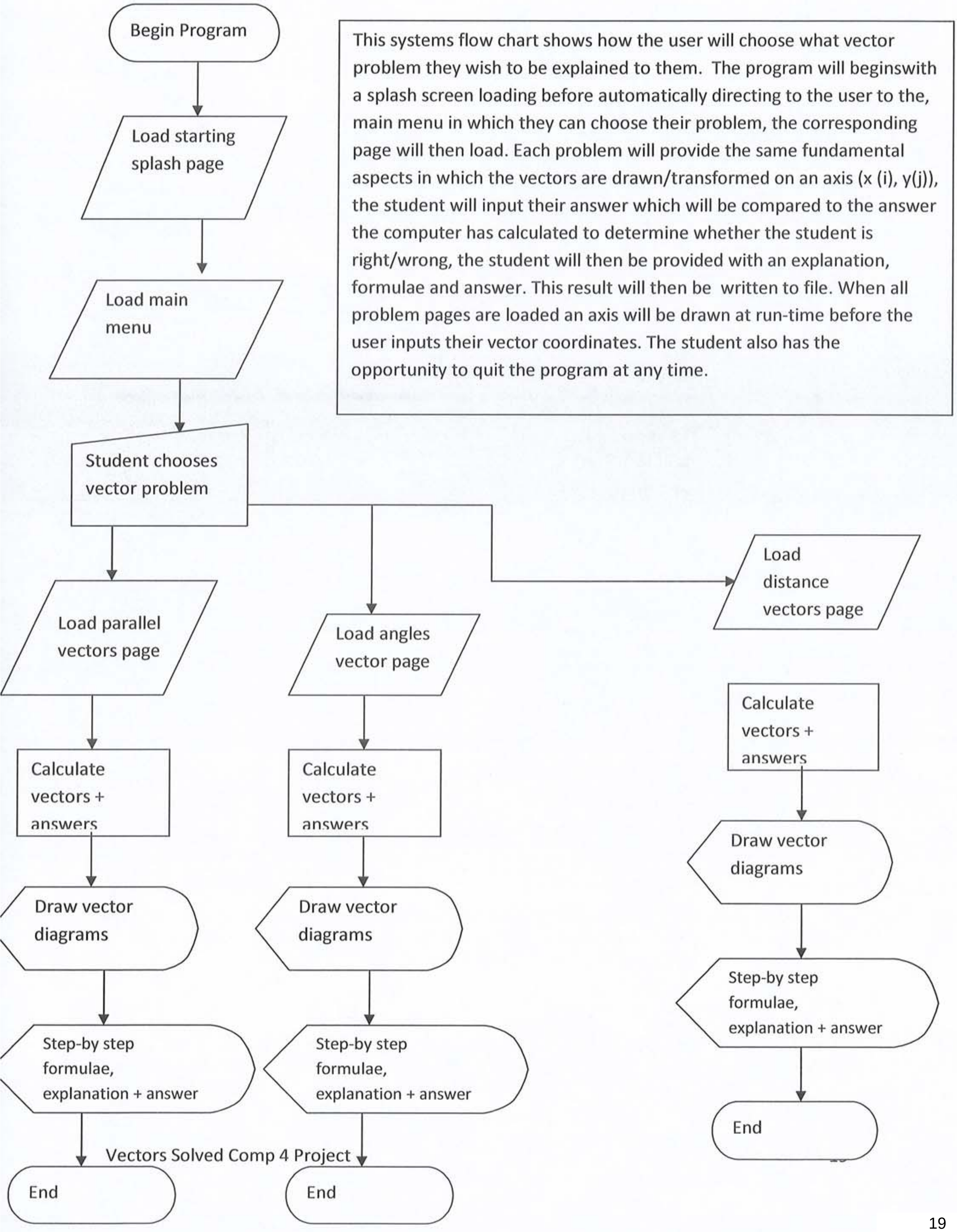
- The computer uses formulae within the system to calculate the correct answer.
- Step by step formulae is shown to the student along with an explanation and answer.

- The pupil can either enter new coordinates to try again, go back to the main menu, print the page or quit.

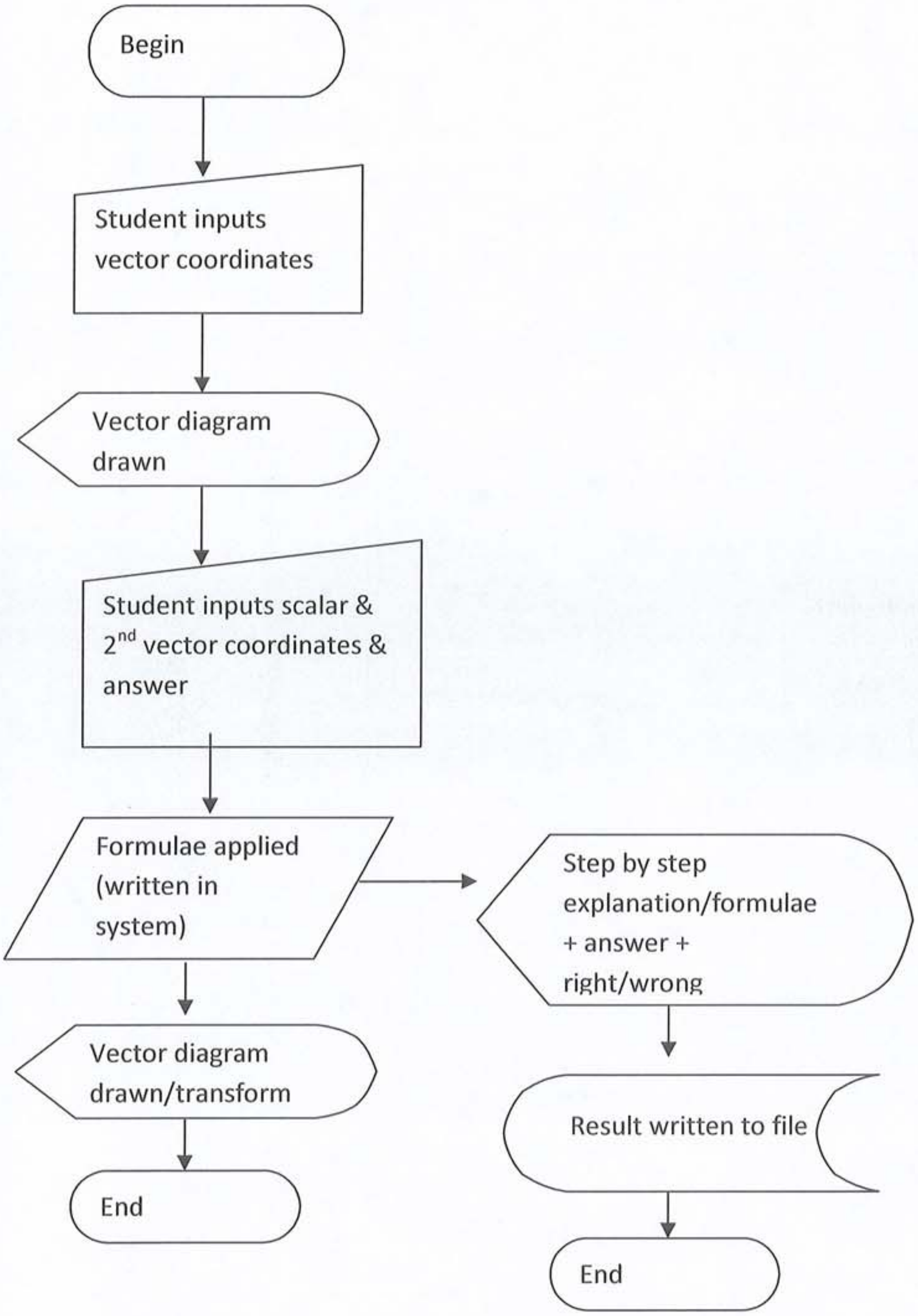
System Flowcharts

Selection of Vector Problem

This systems flow chart shows how the user will choose what vector problem they wish to be explained to them. The program will beginswith a splash screen loading before automatically directing to the user to the, main menu in which they can choose their problem, the corresponding page will then load. Each problem will provide the same fundamental aspects in which the vectors are drawn/transformed on an axis (x (i), y(j)), the student will input their answer which will be compared to the answer the computer has calculated to determine whether the student is right/wrong, the student will then be provided with an explanation, formulae and answer. This result will then be written to file. When all problem pages are loaded an axis will be drawn at run-time before the user inputs their vector coordinates. The student also has the opportunity to quit the program at any time.

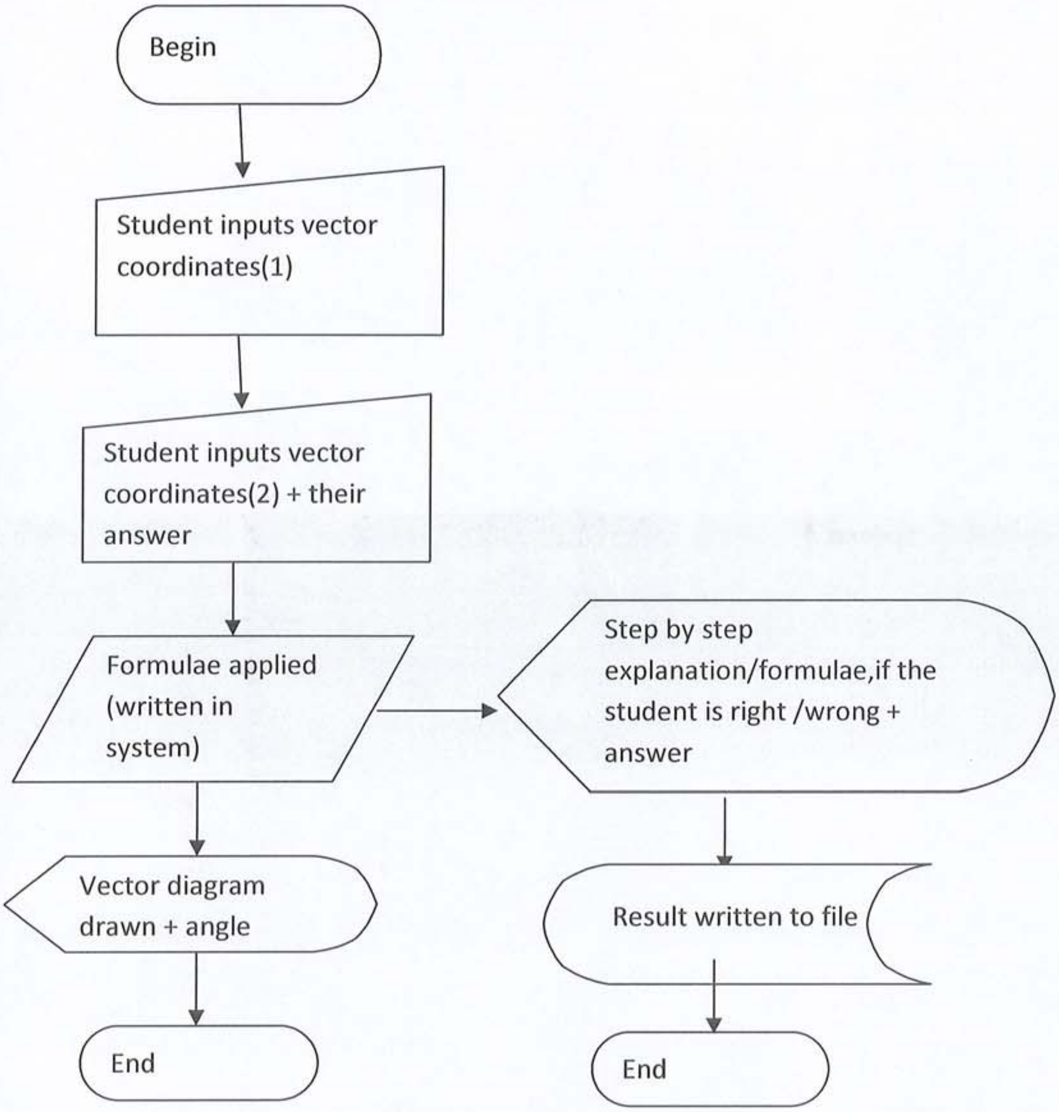

Begin Program
Load starting splash page
Load main menu
Student chooses vector problem
Load parallel vectors page
Load angles vector page
Load distance vectors page
Calculate vectors + answers
Calculate vectors + answers
Calculate vectors + answers
Draw vector diagrams
Draw vector diagrams
Draw vector diagrams
Step-by step formulae, explanation + answer
Step-by step formulae, explanation + answer
Step-by step formulae, explanation + answer
End
End
End

Calculating Parallel Vectors


Begin
Student inputs vector coordinates
Vector diagram drawn
Student inputs scalar & 2nd vector coordinates & answer
Formulae applied (written in system)
Vector diagram drawn/transform
End
Step by step explanation/formulae + answer + right/wrong
Result written to file
End

When the user selects the option of calculating if vectors are parallel the corresponding page will load and the above processes will occur within the system, the user will input the coordinates of their first vector which will then be drawn onto the screen between the axis; the user will then be prompted to enter the coordinates of the vector they wish to compare , its scalar and what they think the answer will be, this 2nd vector will then be drawn in a different colour on the screen and the user will be able to compare if they are parallel. They will then be provided with an explanation and step by step formulae along with whether their answer was correct, their result of how many their score will then be written to file.

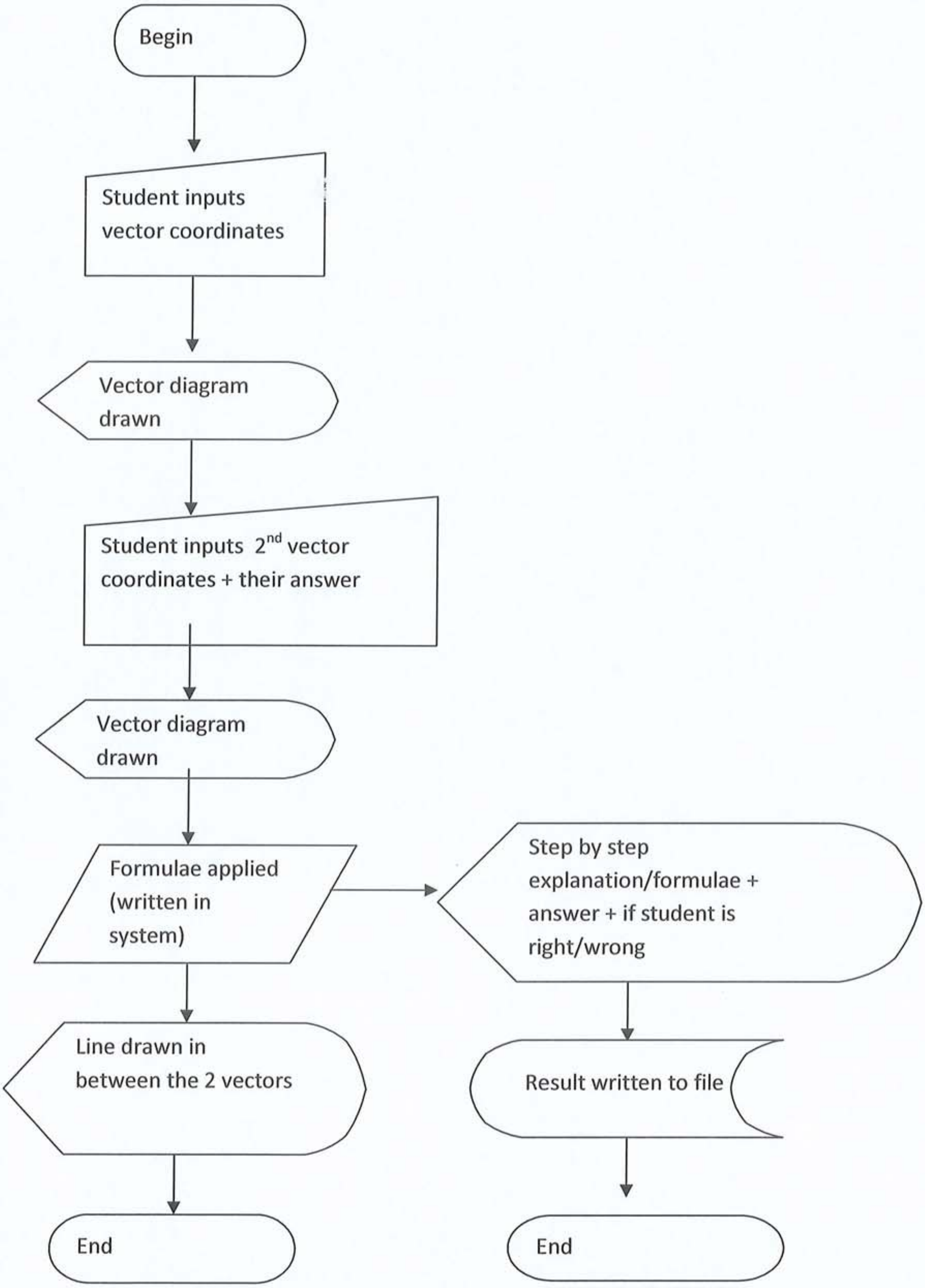
Calculating angles between 2 vectors


Begin
Student inputs vector coordinates(1)
Student inputs vector coordinates(2) + their answer
Formulae applied (written in system)
Step by step explanation/formulae,if the student is right /wrong + answer
Vector diagram drawn + angle
Result written to file
End
End

The above flow chart demonstrates what occurs when the user opens the calculate the angle size between 2 vectors page. The user will be prompted to input 2 sets of vector coordinates, the necessary formulae is then applied to these coordinates and they will be drawn in the precise manner with the correct angle size between them, as this is occurring the step by step explanation and formulae will be displayed on screen along with whether the answer the student has inputted is correct. The answer will be finally displayed to the user. Once again this will all be conducted in a timely manner using timers and making sure that things don't run too quickly so the student has enough time to embrace the concept. Once this is complete the result of how the student has performed will be written to file.

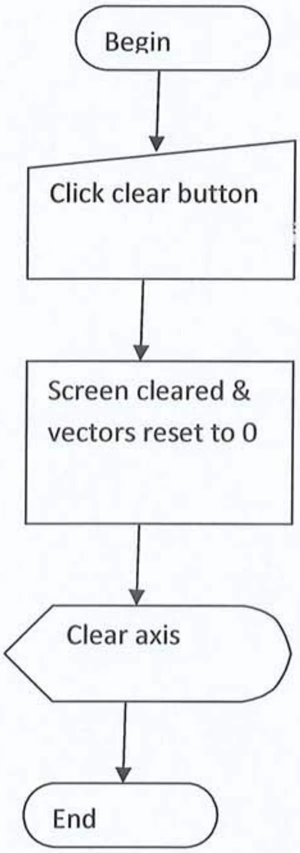
Calculating the distance between 2 vectors

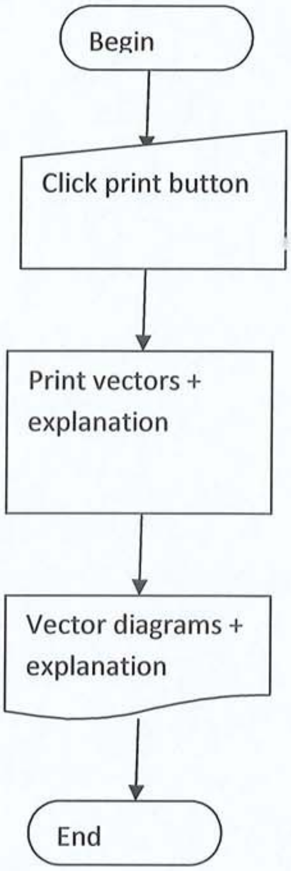
Calculating the distance between 2 vectors


Begin
Student inputs vector coordinates
Vector diagram drawn
Student inputs 2nd vector coordinates + their answer
Vector diagram drawn
Formulae applied (written in system)
Step by step explanation/formulae + answer + if student is right/wrong
Line drawn in between the 2 vectors
Result written to file
End
End

Similar to the above 2 flowcharts this flow chart follows a similar pattern apart from the formulae used is of course different and the system draws another line between the 2 vectors of a different colour to symbolise the distance between the 2 vectors.

Resetting vectors and printing





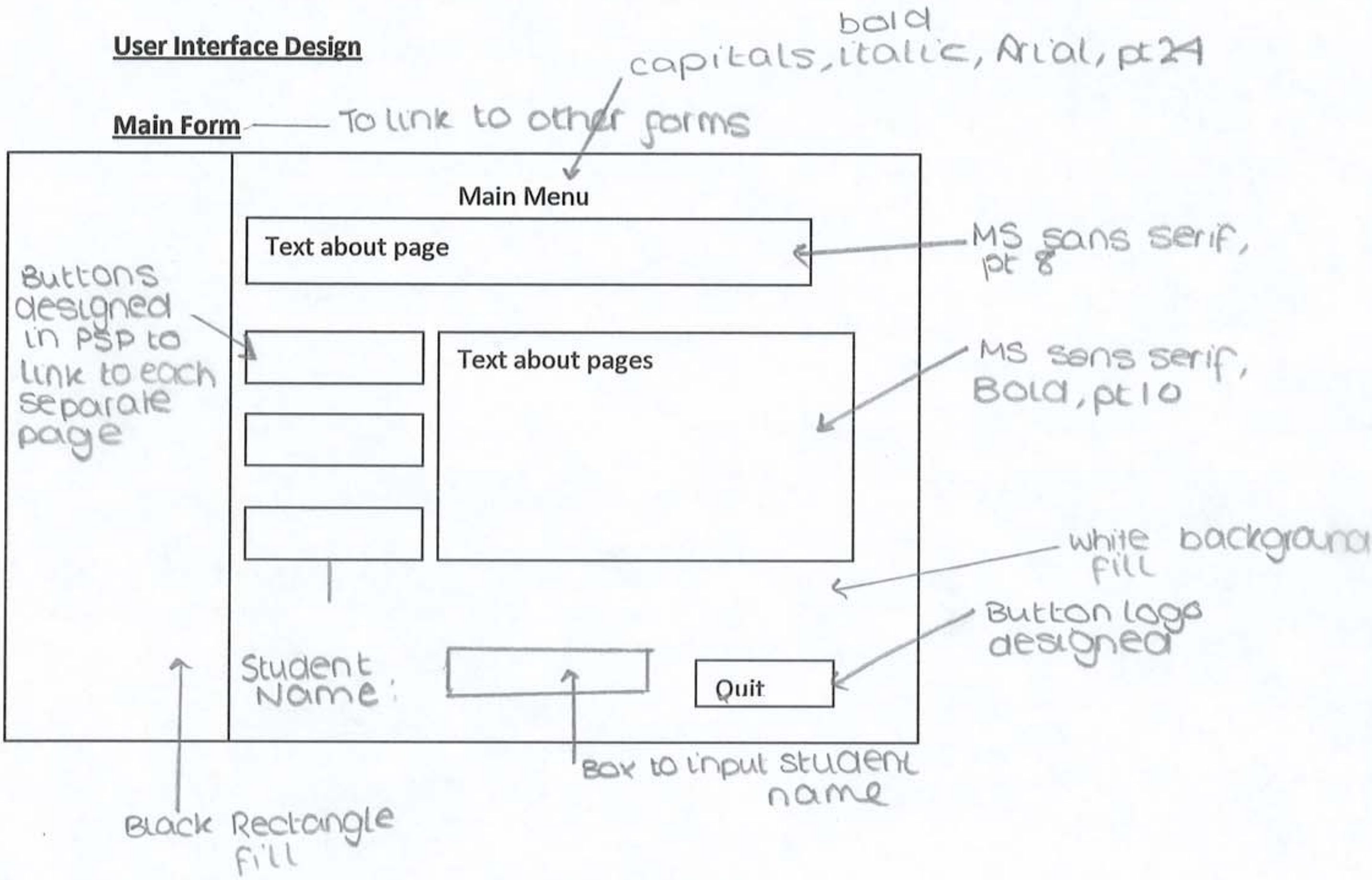
These flowcharts represent the process that is undertaken if the student clicks one of the command buttons to either reset the vector coordinates to 0 so they can enter a different problem; or if they design to print off their vector diagram and explanation so they have a hard copy of their work.

HCI Rationale

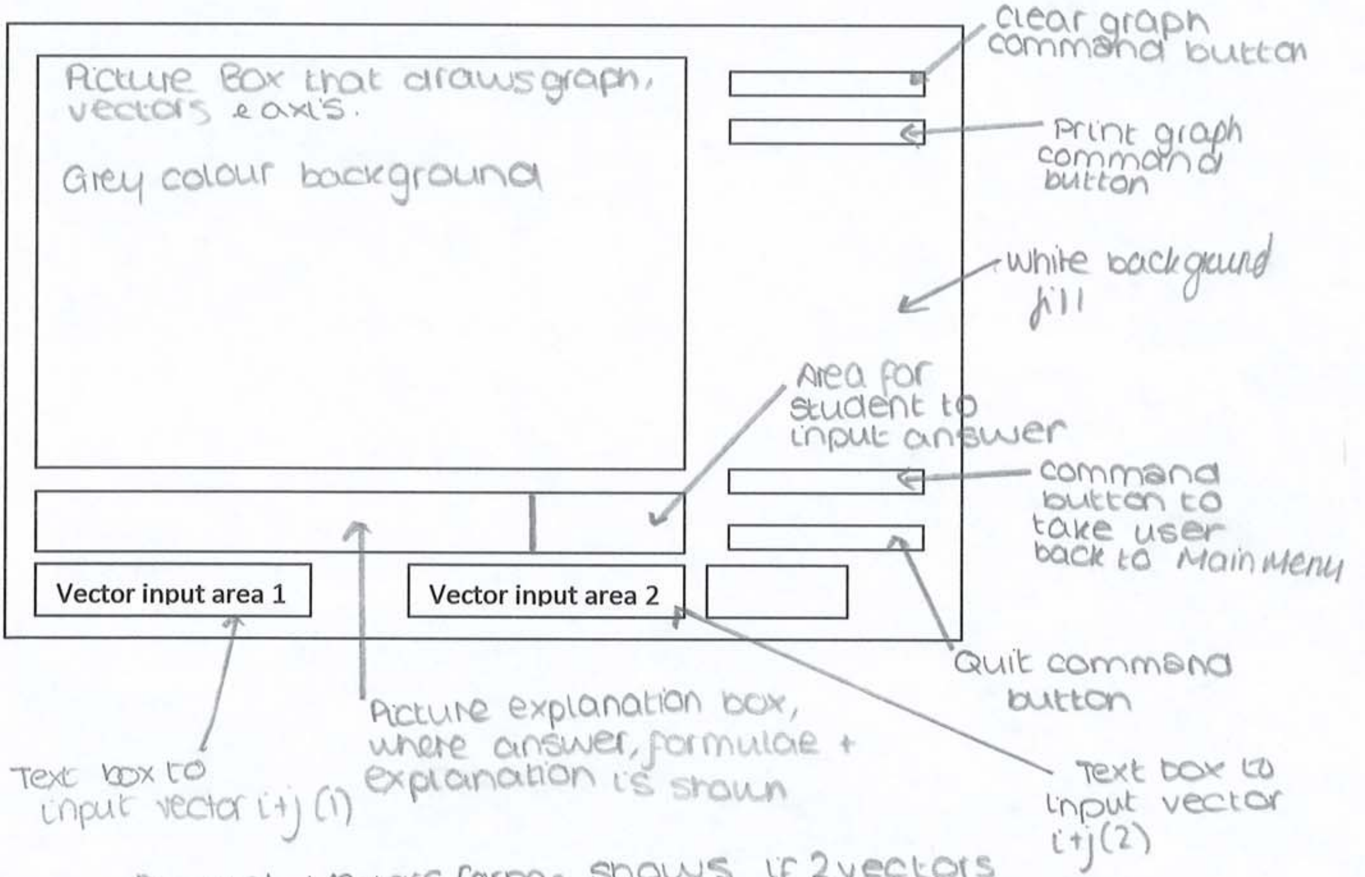
The following style and colour scheme has been chosen for my system for numerous reasons, the classy black feel with a slight hint of colour on the main loading form makes the program presentable but also fun for the student, without confusing them too much. The main forms that run throughout the program are plain grey in colour; however different colours have been used to draw different vectors to help reinforce the students understanding of the concept of vectors.
I have used plain Arial font throughout the program as I believe this makes it easy for the student to read and understand the system; the layout is also simplistic with large buttons to make it clear as to what each buttons function is, and easy to enter into text boxes that make working around the system much more user friendly.

User Interface Design

Main Form



Parallel Vectors, Angle Vectors & Distance Vectors Form

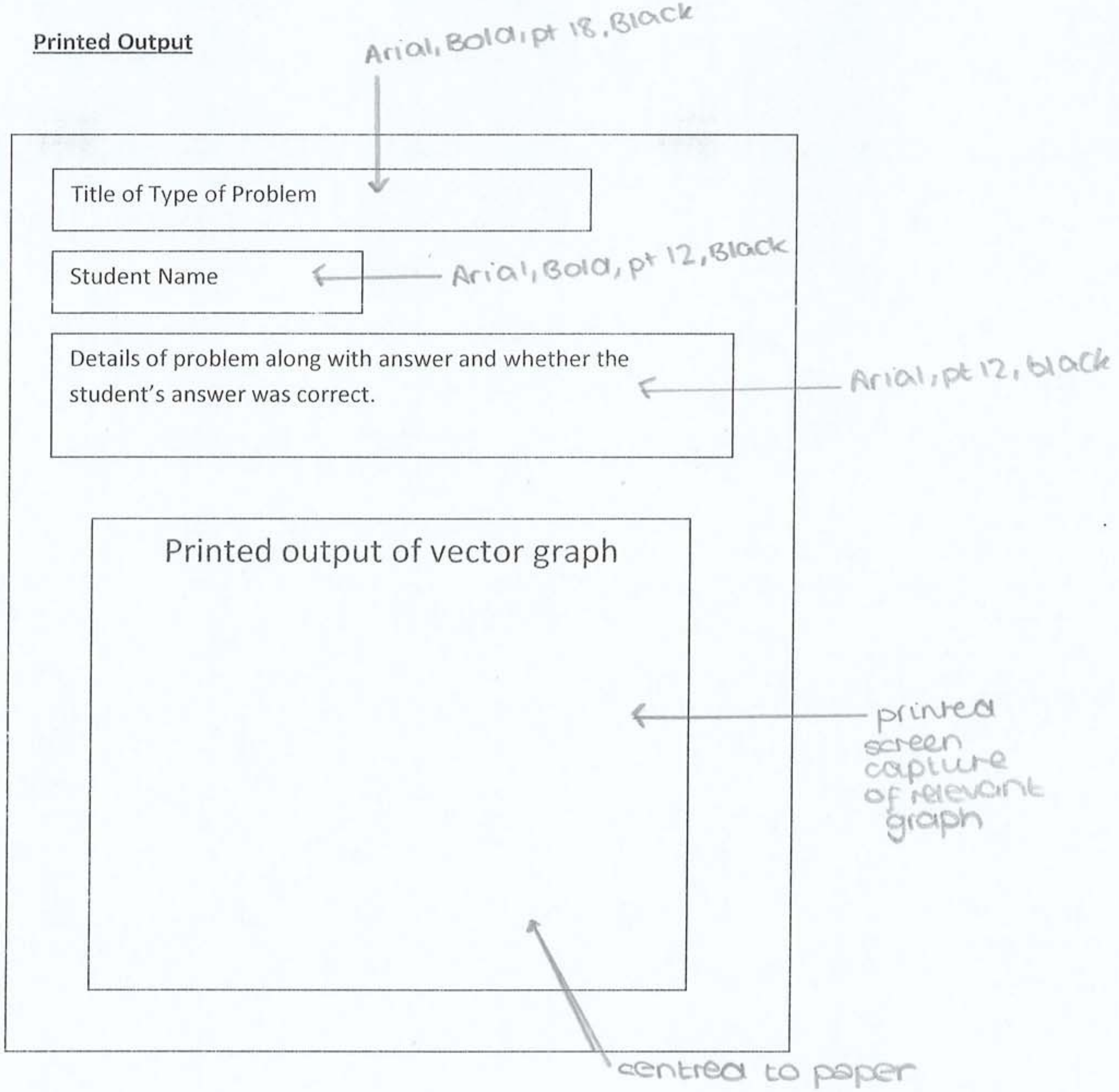

clear graph command button
Picture Box that draws graph, vectors & axis.
Grey colour background
Print graph command button
white background fill
Area for student to input answer
command button to take user back to Main Menu
Vector input area 1
Vector input area 2
Quit command button
Picture explanation box, where answer, formulae + explanation is shown
Text box to input vector i+j (1)
Text box to input vector i+j (2)

Parallel vectors-form - shows if 2 vectors are parallel

Angle vectors form - calculates the angle between 2 vectors

Distance vectors - calculates the distance between 2 vectors

Printed Output



Title of Type of Problem

Student Name

Details of problem along with answer and whether the student's answer was correct.

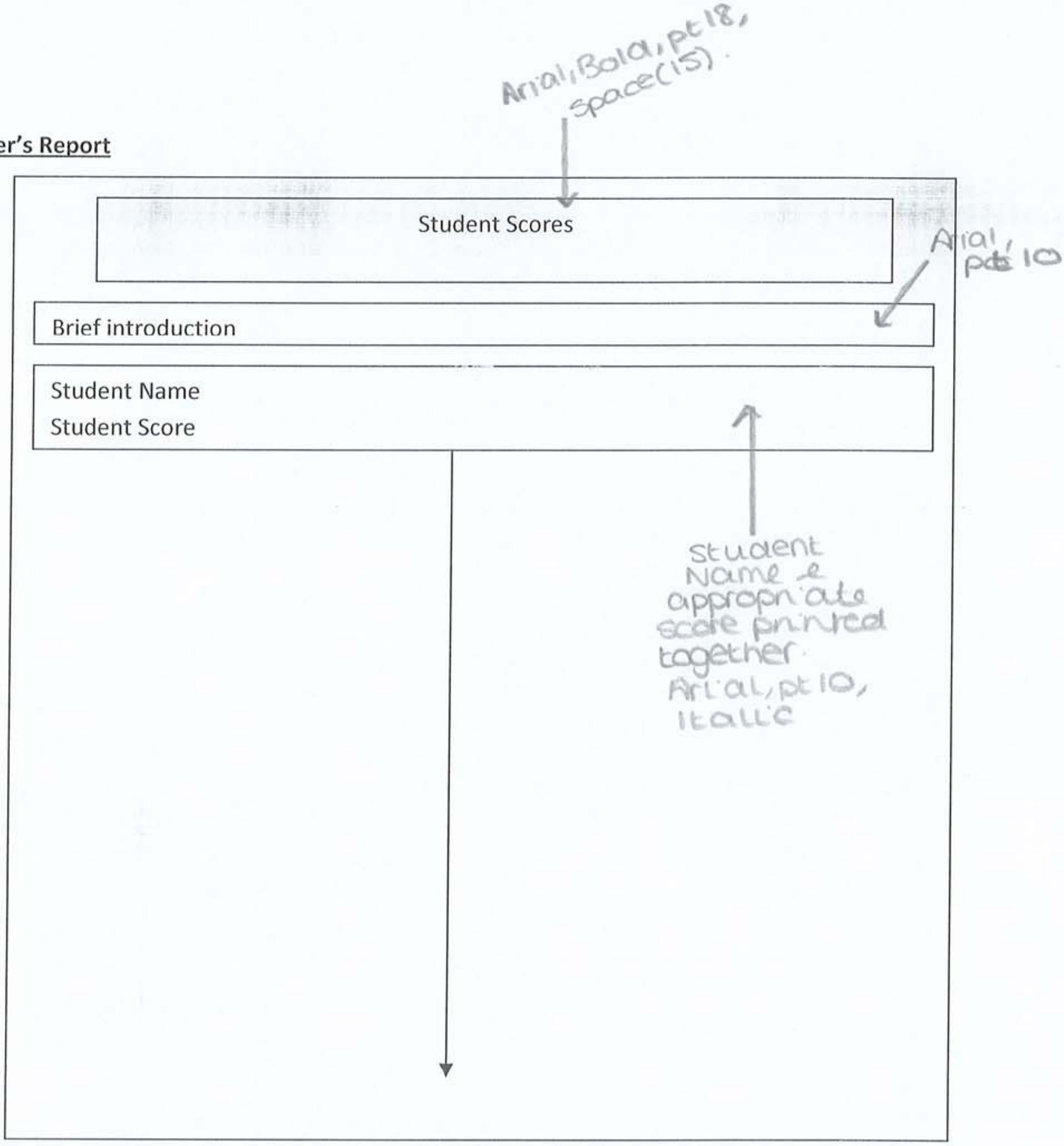
Printed output of vector graph

Arial, Bold, pt 18, Black
Arial, Bold, pt 12, Black
Arial, pt 12, black
printed screen capture of relevant graph
centred to paper

Along with the forms that I have designed for my system I also have needed to design what I want the printed output that will be given to the student will look like; although fairly simplistic in design, so as to not confuse the student it will contain all the necessary information that is needed for them & Mr M[redacted].

It will contain the name of the vector problem they have solved along with details of the vector coordinates and how the answer has been reached. It will then provide them with feedback as to whether they answered the question correctly or not and will be finished with their name and printed output of their vector graph to refer back to.

Teacher's Report



Arial, Bold, pt18, space(15).

Student Scores

Arial, pts 10

Brief introduction

Student Name

Student Score

Student Name & appropriate score printed together. Arial, pt 10, Italic

A teacher's report will also be needed in order for Mr M to have a hard copy of how the students are performing in each of the topics, so he can see which pupils are struggling with which problems. This will printed from the Main Menu form and will contain details of how the students have scored on what vector page and how many questions they have answered correctly.

Data Requirements

Data Dictionary

Variable	Type	Max Length	Validation	Description
Vectori1	Integer	2	Range check between axis values & type check	The i (x) coordinate of the first vector the student inputs.
Vectorj1	Integer	2	Range check between axis values & type check	The j (y) coordinate of the first vector the student inputs.
Vectori2	Integer	2	Range check between axis values & type check	The i (x) coordinate of the second vector.
Vectorj2	Integer	2	Range check between axis values & type check	The j (y) coordinate of the second vector
IMin	Integer	3	Written before run time (none)	The Minimum value that an I vector coordinate can take.
JMin	Integer	3	Written before run time (none)	The Minimum value that a J vector coordinate can take.
IMax	Integer	3	Written before run time(none)	The Maximum value that an I vector coordinate can take.
JMax	Integer	3	Written before run time (none)	The Maximum value that a J vector coordinate can take.
Scalar	Integer	2	Range check between -5 & 5 & type check	The non-zero constant that is multiplied by a vector to see if it is parallel to another.
Angle Size	Integer	3	None is output from a process	The angle size between 2 vectors that is calculated using formulae
Formulae	String Array	30	Written before run time (none)	Will contain 3 elements that will be used for each of the needed formulae.

Explanation	String - Array	100	Written before run time (none)	The step-by-step explanation given to students depending on their chosen problem.
Line	Integer	3	None (output from process)	This will be used when calculating the distance between 2 vectors
StudentAnswer	Integer	4	Presence check	The student will input this to see whether they are able to calculate the answer correctly.
Correct	Boolean	2	None (output from formulae)	Will compare student answer to answer to see if the student is correct or not.
StudentResult	Integer	3	None (output from formulae)	Calculated by computer using correct boolean to determine how many answers the student has entered correctly.
Answer	Integer	5	None (ouput from formulae)	Answer worked out by computer using formulae for distance between vectors.
Parallel	Boolean	3	None (ouput from formulae)	Calculated by computer to determine whether 2 vectors are paralle.
QuestionsAns	Integer	20	None (ouput from formulae)	Calculated by computer to see how many questions the student has answered
StudentName	String	30	Presence Check	Input by the student on the main menu to be written to file along with their results.

Storage Requirements

As you can see from the above data dictionary my program will not be holding large amounts of data as a database would, therefore the amount of memory that my program takes up is very minimal and only requires for the actual program itself and the small text file that will be used to store the student's names and their results unlike a data base. Although I am creating a running model/simulation system I will need a few KB of memory space to store this text file and it will need to be able to be accessed by Mr M so he can see how his students are performing in each of the separate sections.

Type of Storage

As I have stated above my program won't take up vast amounts of memory and therefore it seems to me that the most appropriate method of storage would be that of the Hard Disk of the schools computers. The program will be installed on the networks main server as a stand alone executable file and will therefore be accessible to any pupil throughout the school who wishes to use it. It will make more sense to just install the program on the main hard drive of the network as this will save large amounts of time, as it will mean that it will not have to be installed on every separate computer within the school and the students have the confidence to know that no matter what computer they are on they are able to access the program from the network. Students will also have the option to install the program at home for revision; this will be made possible by the use of the student's memory stick, the program will easily be stored on the memory stick and then taken home for the student to run on their own computer. In accordance with this my program will also be uploaded onto the school's VLE, which will allow the students to access the program anywhere as long as they can log into the school network.
The text file however used to store the student's results will need to be stored in a different area and can only be accessed by Mr M ; this will ensure that the students cannot change their scores or cheat; the hard disk will still be the most appropriate method of storage but it will be stored in a different area of it off limits to students. Mr M has also requested the text file to be backed up onto his memory stick.

Therefore the following hardware will be needed in order to use the system :-

VDU (Visual Display Unit) *– In order for the pupil to see the application and their vectors*
Keyboard *– To type the values needed for their coordinates*
Mouse *– To interact with the GUI*
Memory Stick *– So the pupil can take the program home*
Network Hard Drive *– So anyone in the school can access the program no matter where they are*

Security Requirements

Data Security & Integrity

None of the data that I will be using in my program is of a sensitive nature about a data subject as it would be in a database; therefore security isn't as essential in the application that I am going to produce as Mr M wants everyone to be able to access the program with ease, if there is too much security on it, e.g. passwords etc it may make it too awkward for the students to easily log on and stop them using it. The text file will need some security on it but not large amounts; I will make sure that only Mr M will have access rights to the section of the hard disk where the text file is to be stored to make sure that no other pupils can access it.
To make sure that the integrity of my data is satisfactory I will be using validation checks within my program to make sure that the student can only enter an integer in an integer variable and that it is

within a given range, this will make sure that there is not overflow within the program and that the program can work correctly with the given values. Validation checks will be extremely important in my program as I must make sure that the program is giving the student the correct values for the problems they input as to make sure that the students are being taught correctly. Pupils will only be able to access the compiled executable version of the program and therefore will have no access to the inner workings of the code.

System Security

System Security isn't vital in my program as Mr M wants anyone to be able to access the program with ease meaning that it will be available to any pupil on any computer so they have the freedom to use it whenever they want.

Never the less it is still important that the actual coding of my program is kept secure as any slight alterations in the code could cause major problems when it comes to the running of the program, therefore this will be stored more securely on the network along with the text file of results and only myself and Mr M , who may wish to look at the workings of his program & the results, have the access rights to view it. As the application I am going to produce is small enough to fit onto a memory stick this will be used as backup copies of the application in the event of it being wiped from the network.

Text File

I will be using a text file within my system to score the students' results as they interact with and answer questions on their chosen vector problems. The text file will be serial access and fairly simple to make it easy for Mr M to see the students progress and enable him to give them the relevant feedback and amend his lessons accordingly.

Record Structure of File

Student Scores Text File
Name as **String**
Score as **Integer**
End of File

Suitable Algorithms Required

Of course, as with any complex program my system will require a various number of different algorithms in order to make it work efficiently, I have identified these below :

- **DrawAxis** – This will draw the axis that I will be using for my vectors inside the picture box.
- **DrawVectors** – This will draw the inputted vector units that the user has inputted.
- **VectorAnim** – This algorithm will be a recursive algorithm that will animate the vectors moving across the screen to show the transformation taking place.
- **GetAngle** – This algorithm will calculate the angle between the 2 vector lines that the user has inputted.

- **GetAnswer** – This algorithm calculates the answer that is needed to calculate the angle between 2 lines.
- **GetParallel** – This algorithm works out whether 2 vectors are parallel to one another and returns the value.
- **ClearGraph** – This algorithm clears the picture box of its most recent vector values so the user can draw another graph.
- **GetDistance** – This calculates the distance between 2 vectors.
- **PrintVectors** – This algorithm prints the explanation and the graph.
- **WriteStuName** - This algorithm writes the students name & score to file.
- **ReadStuResults** – This algorithm reads the students results from file and prints them so Mr M has a hard copy of results he can refer to.

DrawAxis

```
Procedure DrawAxis (xmin,xmax,ymin,ymax)
DrawScale = (xmin,xmax) – (ymin,ymax)
   DrawLine  (xmin, 0)-(xmax, 0)        (This draws the xaxis)
For counter = (xmin+1) To (xmax-1)
   DrawLine (counter, -1)-(counter, 1) (This draws the intervals in my x axis)
Next counter
   Draw Line (0, ymin)-(0, ymax)   (This draws the yaxis)
For counter = (ymin+1) To (ymax-1)
   DrawLine  (-1, counter)-(1, counter) (This draws the intervals in my y axis)
Next counter
ForeColour=black
End Procedure
```

To draw the axis within the picture box, I use 2 loops; firstly the x axis is drawn using the minimum and maximum values of the constant xmin, xmax. The interval lines are then drawn along the x axis using the first for loop, the y axis is then drawn using the constant ymin, ymax, this is followed by using the second For Loop to draw the intervals on the y axis.

DrawVectors

```
Procedure DrawVectors (vectori(),vectorj())
Draw Line   (vectori(1), 0)-((vectori(1) + vectori(1)), vectorj(1)),  Black (Draws first vector)
Call  Vector Anim(pass variables) (Calls the animation vector procedure)
End Procedure
```

This algorithm draws the first vector coordinates, starting from 0 on the y axis, to make sure the length etc is correct the second i coordinate is vector i(1)+vectori(1), this procedure then calls the vector animation procedure to draw the second vector line through the process of recursion and animation using timers.

VectorAnim

```
Procedure VectorAnim (vectori(),vectorj())
If vectori(1) < vectori(2) And vectorj(1) < vectorj(2) Then
    vectori(1) = vectori(1) + 1
    vectorj(1) = vectorj(1) + 1
Start Timer
   Call VectorAnim(vectori(),vectorj())  (Procedure calls itself (recursion))
ElseIf vectori(1) = vectori(2) And vectorj(1) < vectorj(2) Then
    vectorj(1) = vectorj(1) + 1
Start Timer
   Call VectorAnim(vectori(),vectorj())  (Procedure calls itself (recursion))
ElseIf vectori(1) < vectori(2) And vectorj(1) = vectorj(2) Then
    vectori(1) = vectori(1) + 1
Start Timer
   Call VectorAnim(vectori(),vectorj())  (Procedure calls itself (recursion))
Stop Timer  (Once vectors are the same timer stops)
End If
End Procedure
```

The algorithm that I will use to make sure that the vectors animate properly is a recursive algorithm in which the procedure VectorAnim calls itself. I will use one large IF statement to portray the various occurrences that can occur between the vector coordinates. To make sure the program doesn't crash the recursive algorithm makes sure that if none of the IF statements are met it does not enter the rest of the code. During this algorithm the timer function is called to make sure the vectors are drawn in a timely manner.

GetAngle

```
Function GetAngle (answer)
GetAngle = InverseTan(-answer/SquareRoot(-answer*answer+1))+2*InverseTan(1))*(180/Pi)
 Format GetAngle "#0.00"
End Function
```

This function takes the value of tan of the answer and uses it along with the answer to produce the value of the inverse of cos, it then formats the angle to 2d.p.

Get Answer

```
Function GetAnswer (vectori(),vectorj())
answer1 = ((vectori(1) * vectori(2)) + (vectorj(1) * vectorj(2)))
answer2 = (SquareRoot(vectori(1) ^ 2 + vectorj(1) ^ 2)) * (SquareRoot(vectori(2) ^ 2 + vectorj(2) ^ 2))
GetAnswer = answer1 / answer2
End Function
```

This function works out the two different answers by using mathematical formulae, it then divides the answers by each other to work out the answer to put into Inverse Cos function.

Get Parallel

```
Function GetParallel (vectori(),vectorj())
If (vectori(1) * vectorj(2)) - (vectorj(1) * vectori(2)) = 0 Then
 GetParallel = True
 Else
 GetParallel = False
End If
End Function
```

This algorithm again uses mathematical formulae to work out if the 2 vectors that have been inputted are parallel or not. It multiplies the 1st i coordinate with the 2nd j coordinate and subtracts it the 1st j coordinate multiplied by the 2nd i coordinate. If this equals 0 then the vectors are parallel.

Clear Graph

```
Command Clear Click
Clear Picture Box
Clear Explanation Box
Call DrawAxis (Redraws the axis ready to be used again)
End
```

This algorithm is used when the clear graph button is clicked, it firstly clears the picture box and then the explanation box where the answer, formulae etc is written. It then calls the drawaxis again to redraw the axis ready to be used once more.

GetDistance

```
Function GetDistance (vectori(),vectorj())
answer = Square Root(((vectori(1) - vectori(2)) ^ 2) + ((vectorj(1) - vectorj(2)) ^ 2))
End Function
```

This algorithm the value of the distance between the 2 vectors it firstly calculates the distance between the i vectors and raises it to the power of 2 and then adds it to the difference between the j vectors to the power of 2.

WriteStuName

```
Procedure WriteName(StuScore,QuestionsAns)
Name as String
Filename = VectorQuestionsScore.text
Name = frmMain.txtName
Open Filename For Append
Write, Name, StuScore, QuestionsAns
Close File
End Sub
```

This algorithm takes the students name, their score and the amount of questions they have answered and writes them to a text file.

PrintGraph

```
Procedure PrintGraph
Initialise Printer
Font=Bold
FontSize = 18
Print Vector Problem
FontSize = 12
Print StudentName
Font = Normal
Print Details of Problem
Print StudentAnswer Correct
Print VectorGraph (using function from API)
End Procedure
```

The Print Graph algorithm allows the student to print out their desired vector graph, along with an explanation of the problem and if they have answered the question correctly, this is all formatted appropriately and printed of with their name.

Read StuResults

```
Private Sub PrintResults
Initialise Printer
Tmp as String
Filename = VectorQuestionScores.txt
Font=Bold
FontSize=18
Print "Student Scores"
Font=Normal
FontSize=10
Print"The results of how the students have scored on the various topics are as follows:"
Font = Italic
Open Filename For Reading
While EOF(1) = 0
Input Line, tmp
   Print  tmp
   End Loop
Close File
```

The ReadStu Results algorithm allows Mr Mason to print a hard copy of all the Student's results by reading their results back from the text file and printing them to a hard copy.

Testing Strategy

Before the program can be used by students it needs to be properly tested to ensure that there are no errors within the system and that there is sufficient validation to make sure that the program runs correctly when values are entered by the students and that it provides them with the correct explanation and answers. This is vital to do so as if the program is not tested properly it could create problems later on for Mr M and his students when errors crop up and they do not know what to do. I will be use a mixture of black box and white box testing; black box in order to test that when values are entered into the system they are either accepted or rejected and white box testing to work through the code and pinpoint any points of the system that aren't working as is specified/required.

To begin with I will carry out functional testing this will test the individual components of the system, for example the procedures and functions I create, this will help to break down my system and show clearly which parts of the system are working and which are not.

I will then move on to Integration testing which will involve me combining units of my program to test using a Top Down Testing method where I will test the top integrated modules and work my way down until I have reached the bottom of the modules, helping me to find any missing branch links within my program.

Penultimately a full systems test will occur, testing the application from beginning to end; this will test the security of the system, the GUI, how it handles errors, its performance, its loading etc and make sure that I have a fully working application to give to the user.

Finally the finished program will be given to the user to test, they will use the program and report feedback to me in how the program works and whether it is carrying out the tasks and processes that is required.

Unit Testing

Main Form

Test Number	Test Data	Purpose	Expected Result
1	Splash screen links to Main Form within 5 seconds	To introduce the user to the program before starting up.	The splash screen shows for 5 seconds, the main form then loads.
2	Click cmdParallel	To take the user to the Parallel Vectors page.	Main menu closes and Parallel Vector page opens.
3	Click cmdAngle	To take the user to the Calculate Angle page.	Main menu closes and Angle page opens.
4	Click cmdDistance	To take the user to the	Main Menu closes and

		Distance Between Vectors page.	Distance Between Vectors page opens.
5	Click cmdQuit	To quit the application.	Application closes.

Parallel Vectors Form

Test Number	Test Data	Purpose	Expected Result
6	X Axis (15 & -15)	Check x axis is set between the range	X axis is set between the range
7	Y Axis (15 & -15)	Check y axis is set between the range	Y axis is set between the range
8	X axis (15 & -15)	Check X axis is plot correctly.	X axis is plot correctly
9	Y Axis (15 & -15)	Check Y axis is plot correctly	Y axis is plot correctly
10	Vectori1 (2)	Check x coordinate of vector is correctly drawn	X Coordinate is correctly drawn
11	Vectorj1 (3)	Check y coordinate of vector is correctly drawn	Y Coordinate is correctly drawn
12	Vectori2 (4)	Check x coordinate of 2nd vector is correctly drawn	X coordinate is correctly drawn
13	Vectorj2 (5)	Check y coordinate of 2nd vector is correctly drawn	Y coordinate is correctly drawn
14	Scalar (2)	Check x coordinate of vector 2 is correctly drawn when multiplied by scalar	X Coordinate is correctly drawn
15	Scalar (2)	Check y coordinate of vector 2 is correctly drawn when multiplied by scalar	Y Coordinate is correctly drawn
16	Explanation	Check Explanation shows up correctly step by step	Explanation shows up correctly step by step
17	Answer	Check answer is correct and shows on screen	Answer is correct and is shown on screen
18	Click cmdMainMenu	Takes the user back to the main menu	Parallel vector page closes, main menu page

			opens
19	Click cmdClear	Clears the vector from the screen and allows the user to have another go	Vector, explanation and answer is cleared from the screen and gives the user a chance for another go
20	Click cmdInputVectors	Prompts the user to input their coordinates.	User is prompted to input their coordinates.
21	Click cmdPrint	Allows the user to print off their form	Print Dialog appears for the user to print their form off
22	Click cmdQuit	Allows the user to exit the program	Program closes

Angle Vectors Form

Test Number	Test Data	Purpose	Expected Result
23	X Axis (15 & -15)	Check x axis is set between the range .	X axis is set between the range.
24	Y Axis (15 & -15)	Check y axis is set between the range	Y axis is set between the range
25	X axis (15 & -15)	Check X axis is plot correctly.	X axis is plot correctly
26	Y Axis (15 & -15)	Check Y axis is plot correctly	Y axis is plot correctly
27	Vectori1 (1)	Check x coordinate of vector is correctly drawn	X Coordinate is correctly drawn
28	Vectorj1 (2)	Check y coordinate of vector is correctly drawn	Y Coordinate is correctly drawn
29	Vectori2 (4)	Check x coordinate of 2nd vector is correctly drawn	X coordinate is correctly drawn
30	Vectorj2 (1)	Check y coordinate of 2nd	Y coordinate is correctly

		vector is correctly drawn	drawn
31	Angle Size	Check angle size is drawn correctly	Angle size is drawn correctly
32	Formulae	Check correct formulae is being used	Correct formulae is being used
33	Explanation	Check Explanation shows up correctly step by step	Explanation shows up correctly step by step
34	Answer	Check answer is correct and shows on screen	Answer is correct and is shown on screen
35	Click cmdMainMenu	Takes the user back to the main menu	Parallel vector page closes, main menu page opens
36	Click cmdClear	Clears the vector from the screen and allows the user to have another go	Vector, explanation and answer is cleared from the screen and gives the user a chance for another go
37	Click cmdInputVectors	Prompts the user to input their coordinates.	User is prompted to input their coordinates.
38	Click cmdPrint	Allows the user to print off their form	Print Dialog appears for the user to print their form off
39	Click cmdQuit	Allows the user to exit the program	Program closes

<u>Distance Between 2 Vectors Form</u>

Test Number	Test Data	Purpose	Expected Result
40	X Axis (15 & -15)	Check x axis is set between the range -15 & 15.	X axis is set between the range -15 & 15
41	Y Axis (15 & -15)	Check y axis is set between the range -15 & 15	Y axis is set between the range -15 & 15
42	X axis (15 & -15)	Check X axis is plot correctly.	X axis is plot correctly
43	Y Axis (15 & -15)	Check Y axis is plot correctly	Y axis is plot correctly

44	Vectori1 (2)	Check x coordinate of vector is correctly drawn	X Coordinate is correctly drawn
45	Vectorj1 (3)	Check y coordinate of vector is correctly drawn	Y Coordinate is correctly drawn
46	Vectori2 (4)	Check x coordinate of 2nd vector is correctly drawn	X coordinate is correctly drawn
47	Vectorj2 (5)	Check y coordinate of 2nd vector is correctly drawn	Y coordinate is correctly drawn
48	Line	Check line is drawn correctly between the 2 vectors	Line is drawn correctly
49	LineSize	Check the vector of the line is displayed	Vector of line is displayed
50	Explanation	Check Explanation shows up correctly step by step	Explanation shows up correctly step by step
51	Answer	Check answer is correct and shows on screen	Answer is correct and is shown on screen
52	Click cmdMainMenu	Takes the user back to the main menu	Parallel vector page closes, main menu page opens
53	Click cmdClear	Clears the vector from the screen and allows the user to have another go	Vector, explanation and answer is cleared from the screen and gives the user a chance for another go
54	Click cmdInputVectors	Prompts the user to input their coordinates.	User is prompted to input their coordinates.
55	Click cmdPrint	Allows the user to print off their form	Print Dialog appears for the user to print their form off
56	Click cmdQuit	Allows the user to exit the program	Program closes
57	Line Colour	Check line colour is set to Red	Line Colour is set to red

Integrated Testing

Parallel Vectors Form

Test Number	Test Data	Purpose	Expected Result
58	X & Y Axis (15 & -15)	Check X & Y axis are plotted correctly	X & Y axis are plotted correctly
59	Vectori1 & Vectorj1 (2 & 3)	Check that the vector is correctly drawn using both the coordinates	Vector is correctly drawn
60	Vectori2 & Vectorj2 & Scalar (4 & 5 & 1)	Check that the 2nd vector is correctly drawn	2nd vector is drawn correctly applying the scalar
61	Drawing of vectors & Explanation	Check that as the vector is 2nd vector is drawn the explanation begins to show	Explanation begins to show step-by-step
62	Answer & Formulae	Check that answer & formulae display at the same time	Answer & formulae display together

Angle Vectors Form

Test Number	Test Data	Purpose	Expected Result
63	X & Y Axis (15 & - 15)	Check X & Y axis are plotted correctly	X & Y axis are plotted correctly
64	Vectori1, Vectorj1, Vectori2, Vectorj2,Angle Size, Formulae (1 & 2 & 4 & 1)	Check that the vectors are drawn together and meet at the correct angle size	Vector is correctly drawn with correct angle size
66	Drawing of vectors & Explanation	Check that once the vectors have finished drawing the	Explanation begins to show step-by-step

		explanation shows	
67	Answer & Formulae	Check that answer & formulae display at the same time	Answer & formulae display together

Distance Between 2 Vectors Form

Test Number	Test Data	Purpose	Expected Result
68	X & Y Axis (15 & - 15)	Check X & Y axis are plotted correctly	X & Y axis are plotted correctly
69	Vectori1 & Vectorj1 (2 & 3)	Check that the vector is correctly drawn using both the coordinates	Vector is correctly drawn
70	Vectori2 & Vectorj2 (4 & 5)	Check that the 2nd vector is correctly drawn	2nd vector is drawn correctly
71	Line & Line Colour	Check that the line is drawn in the correct colour	Line is drawn in the correct colour
72	Drawing of line & Explanation	Check that as the line between the 2 vectors is being drawn the explanation shows	Explanation begins to show step-by-step
73	Answer & Formulae	Check that answer & formulae display at the same time	Answer & formulae display together

Validation – Angle Vectors Form, Parallel Vectors Form, Distance Vectors Form

74	Making sure vectori(1),(2) and vectorj(1),(2) and scalar are not empty – **Presence check**	To make sure that all text boxes are filled	If text boxes are not filled a message box will pop up prompting the student to enter something

75	Making sure student answer text box is not empty – **Presence Check**	To make sure the student has inputted answer	If the student has not inputted an answer they will be prompted to.
76	Making sure vectori(1),(2) and vectorj(1),(2) and scalar are the correct format (integer) –**Format Check**	To make sure the input is of the correct format	If not the correct format an error message will appear asking the student to alter their values.
77	Making sure vectori(1),(2) and vectorj(1),(2) and scalar are within the correct range – **Range Check**	To make sure the input is within the correct limits.	If the vector coordinates are below -15 or above 15 an error message will appear.

Validation is of course extremely important within the design of a system in order to make sure that data entered into the system is reasonable, below I have outlined some pseudo code that will need to be incorporated into the validation of my system.

75

```
If vectori(1) ="" or vectori(2)="" or vectorj(1)="" or vectorj(2)="" or scalar="" Then
MessageBox "Please enter something into the text boxes"
End If
```

76

```
If vectori(1) is not numeric or vectori(2) is not numeric or vectorj(1) is not numeric
or vectorj(2) is not numeric or scalar is not numeric Then
MessageBox "Please enter a valid number into the text boxes"
End If
```

Reading and Printing Results from File

77	Clicking the Print Student Results command button	Making sure that when the Print Student Results button is	The students' results should be printed out in hard copy form.

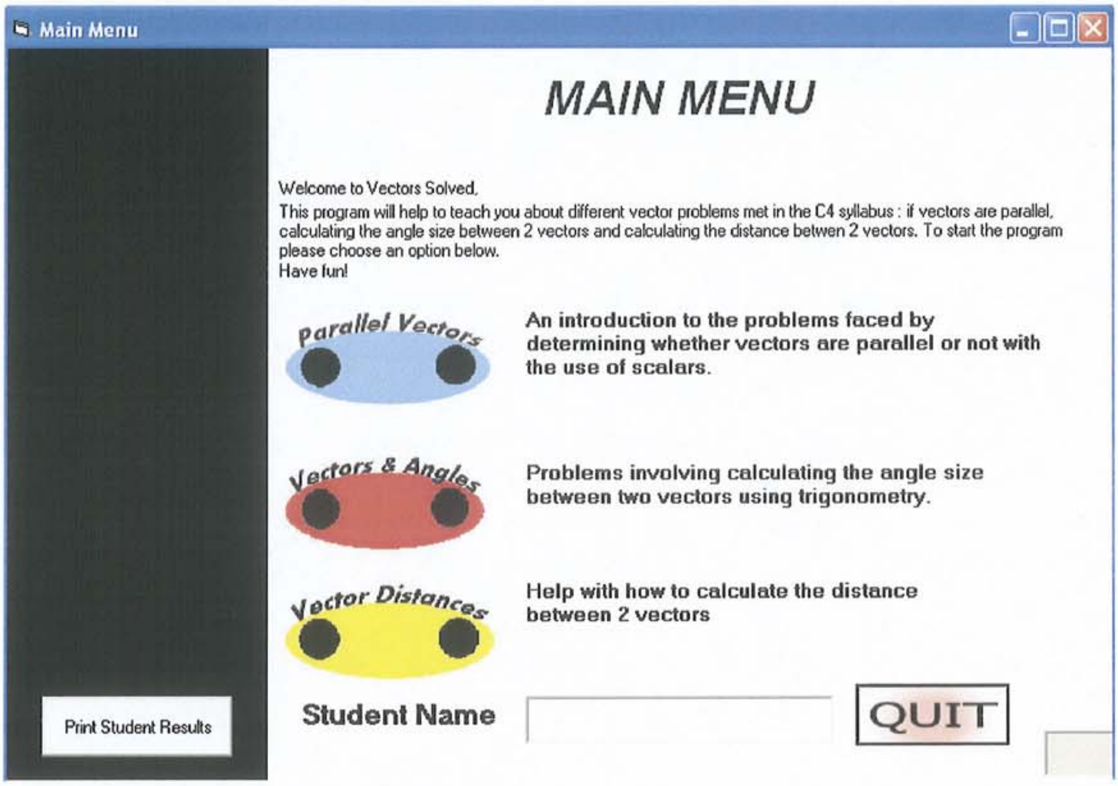
		results are printed out correctly	

Technical

Solution

Technical Solution

Main Menu



Main Menu

MAIN MENU

Welcome to Vectors Solved.
This program will help to teach you about different vector problems met in the C4 syllabus : if vectors are parallel, calculating the angle size between 2 vectors and calculating the distance betwen 2 vectors. To start the program please choose an option below.
Have fun!

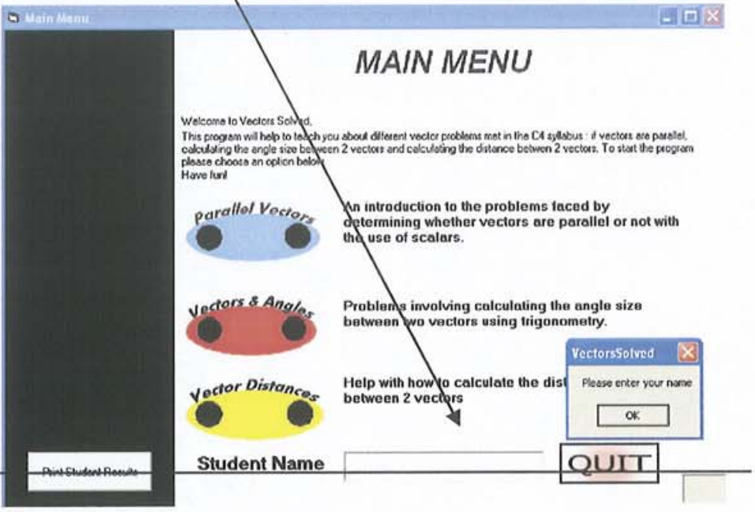
An introduction to the problems faced by determining whether vectors are parallel or not with the use of scalars.

Problems involving calculating the angle size between two vectors using trigonometry.

Help with how to calculate the distance between 2 vectors

```
Dim StuName As String
```

```
Public Sub cmdAngles_Click() 'Makes sure the student enters their name
If txtName.Text = "" Then
MsgBox ("Please enter your name")
Else
StuName = txtName.Text
frmAngleVectors.Show
Call Module1.WriteName
Unload Me
End If
End Sub
```



Main Menu

MAIN MENU

Welcome to Vectors Solved.
This program will help to teach you about different vector problems met in the C4 syllabus : if vectors are parallel, calculating the angle size between 2 vectors and calculating the distance betwen 2 vectors. To start the program please choose an option below.
Have fun!

An introduction to the problems faced by determining whether vectors are parallel or not with the use of scalars.

Problems involving calculating the angle size between two vectors using trigonometry.

Help with how to calculate the dist between 2 vectors

VectorsSolved

Please enter your name

```
Public Sub cmdDistances_Click()
'Makes sure the student enters their name
If txtName.Text = "" Then
MsgBox ("Please enter your name")
Else
```

```
StuName = txtName.Text
frmDistanceVectors.Show
Call Module1.WriteName
Unload Me
End If
End Sub
```

```
Private Sub cmdEnter_Click()
End Sub
```

```
Public Sub cmdParallel_Click() 'Makes sure the student enters their name
If txtName.Text = "" Then
MsgBox ("Please enter your name")
Else
StuName = txtName.Text
frmParallelVectors.Show
Call Module1.WriteName
Unload Me
End If
End Sub
```

```
Private Sub cmdPrintResults_Click()
'Reads results from file and prints them out to a hardcopy
 iniprinter
Dim tmp As String
Filename = App.Path & "\VectorQuestionScores.txt"
Printer.FontBold = True
 Printer.FontSize = 18
 Printer.Print Spc(15); "Student Scores" 'Writes heading
 Printer.Print
 Printer.Print
  Printer.FontBold = False
 Printer.FontSize = 10
 Printer.Print ; Spc(15); "The results of how the students have scored on the various topics are as
follows:"
 Printer.Print
 Printer.Print
 Printer.FontItalic = True
Open Filename For Input As #1
'Opens file and reads results until end of file is reached
While EOF(1) = 0
   Line Input #1, tmp
   Printer.Print Spc(5); tmp
   Printer.Print
Wend
Close #1
   End Sub
```

```
Public Sub iniprinter()
Printer.Print
End Sub
Private Sub cmdQuit_Click()
Unload Me
End Sub
```

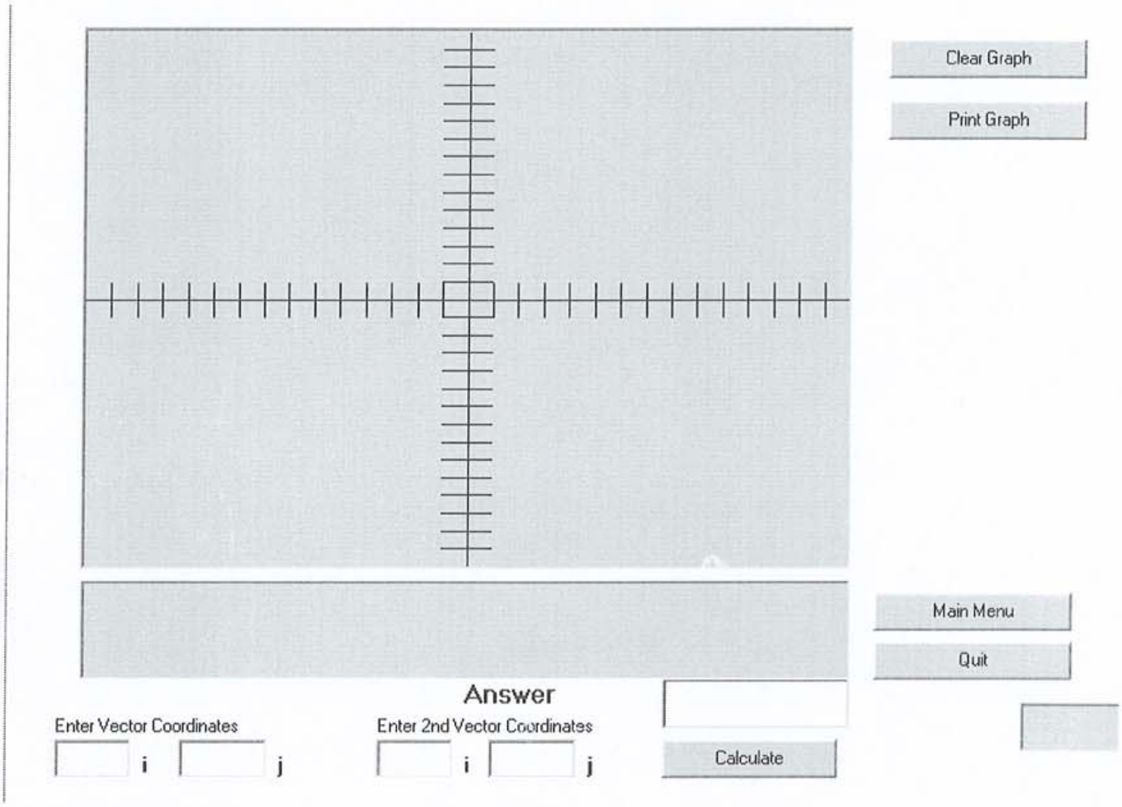
```
Public Sub PassName(StudentName)
StudentName = StuName
```

```
End Sub
```

```
Private Sub Form_Load()
End Sub
```

```
Private Sub Label6_Click()
End Sub
```

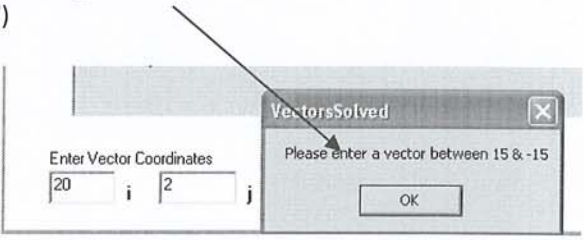
Angle Vectors Form



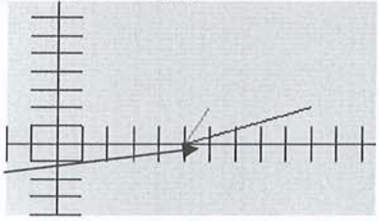
```
Dim vectori(1 To 3) As Integer
Dim vectorj(1 To 3) As Integer
Dim Angle As Double
Dim answer1 As Double
Dim answer2 As Double
Dim validated As Boolean
Dim StudentName As String
Const Pi As Double = 3.14159265358979
Const xmax As Integer = 15
Const xmin As Integer = -15
Const ymax As Integer = 15
Const ymin As Integer = -15
Dim Answer As Double
Dim StuAns As Double
Dim correct As Boolean
```

```
Private Declare Function SendMessage Lib "user32.dll" Alias _
  "SendMessageA" (ByVal hwnd As Long, ByVal wMsg As Long, _
  ByVal wParam As Long, ByVal lParam As Long) As Long
Private Const WM_PAINT = &HF
Private Const WM_PRINT = &H317
Private Const PRF_CLIENT = &H4&    ' Draw the window's client area
Private Const PRF_CHILDREN = &H10& ' Draw all visible child
Private Const PRF_OWNED = &H20&    ' Draw all owned windows
Dim StuScore As Integer
Dim QuestionsAns As Integer
Dim Explanation As String

Private Sub cmdCalculate_Click()
StuScore = 0
If txtVectorI1.Text = "" Or txtVectorJ1.Text = "" Or txtVectorI2.Text = "" Or txtVectorJ2.Text = "" Then
     MsgBox ("Please enter a number into the text boxes")
   validated = False
   ElseIf txtStuAns.Text = "" Then
   MsgBox ("Please enter your answer") 'Makes sure all text boxes are filled
    validated = False
    ElseIf Not IsNumeric(txtVectorI1.Text) Or Not IsNumeric(txtVectorJ1.Text) Or Not
IsNumeric(txtVectorI2.Text) Or Not IsNumeric(txtVectorJ2.Text) Or Not IsNumeric(txtStuAns.Text)
Then
   validated = False
   MsgBox ("Please enter a number into the text boxes") 'Makes sure all values are numeric
   ElseIf txtVectorI1.Text > 15 Or txtVectorJ1.Text > 15 Or txtVectorI2.Text > 15 Or txtVectorJ2.Text >
15 Then
   validated = False
   MsgBox ("Please enter a vector between 15 & -15")
    ElseIf txtVectorI1.Text < -15 Or txtVectorJ1.Text < -15 Or txtVectorI2.Text < -15 Or txtVectorJ2.Text
< -15 Then
   validated = False 'Makes sure all numbers are within the given range
   MsgBox ("Please enter a vector between 15 & -15")
   Else
vectori(1) = txtVectorI1.Text
vectorj(1) = txtVectorJ1.Text
vectori(2) = txtVectorI2.Text
vectorj(2) = txtVectorJ2.Text
vectori(3) = vectori(2)
vectorj(3) = vectorj(2)
StuAns = txtStuAns.Text
validated = True
End If
If validated = True Then
Answer = GetAnswer(vectori(), vectorj()) 'Calls getanswer function
Angle = GetAngle(Answer, Pi) 'Calls get angle function
pctExplanation.Print "The answer is " & Angle & " degrees because  (" & vectori(1) & " x " & vectori(2)
& ") + (" & vectorj(1) & " x " & vectorj(2); ")"; vbCrLf; Tab(35); "_____________"; vbCrLf; Tab(35);
vectori(1); "² + " & vectorj(1); " ² " & vectori(2); " ² + "; vectorj(2); "² "
```



Please enter a vector between 15 & -15



```
Call DrawVectors(pctDraw1) 'Draws vectors
End If
```

```
QuestionsAns = QuestionsAns + 1 'increments questions answered by one
If StuAns = Angle Then
correct = True
StuScore = StuScore + 1
 'if student answer is correct increments student answer by one
pctExplanation.Print "Well done, your answer is correct"
Else
pctExplanation.Print "Sorry that's wrong, try again"
End If
End Sub
```

```
Private Sub cmdClear_Click()
pctDraw1.Cls
pctExplanation.Cls
Call frmDistanceVectors.DrawAxis(pctDraw1, xmax, xmin, ymax, ymin)
End Sub
```

```
Private Sub cmdMainMenu_Click()
frmMain.Show
Filename = App.Path & "\VectorQuestionScores.txt" 'opens file
Open Filename For Append As #1
Print #1, "Angle vectors score ="; StuScore; "out of"; QuestionsAns
'Writes student results to file
Close #1
Unload Me
End Sub
```

```
Private Sub cmdPrint_Click() 'Prints a hard copy of the graph along with an explanation
 iniprinter
  Printer.FontBold = True
 Printer.FontSize = 18
 Printer.Print Spc(5); "Angle Vector Problem"
 Printer.Print
 Printer.FontSize = 12
 Printer.Print "Student Name:"; Spc(3); StudentName
 Printer.Print
 Printer.FontBold = False
 Printer.Print "The vector graph of "; vectori(1); "i + "; vectorj(1); "j and "; vectori(3); "i + "; vectorj(3);
"j"
 Printer.Print
 Printer.Print
 Printer.Print "The answer is " & Angle & " degrees because "; ""
 Printer.Print Tab(43); "(" & vectori(1) & " x " & vectori(3) & ") + (" & vectorj(1) & " x " & vectorj(3); ")";
vbCrLf; Tab(43); "_____________"; vbCrLf; Tab(43); vectori(1); "² + " & vectorj(1); " ² " & vectori(2); " ²
+ "; vectorj(2); "² "
 Printer.Print

 If correct = True Then
 Printer.Print "Well done you correctly answered this question"
 Else
 Printer.Print "Unfortunately your answer was wrong, either try again or make sure to contact your
teacher "
 Printer.Print "with the relevant questions."
End If
AddPictureBox pctDraw1, Printer.ScaleWidth / 2 - pctDraw1.Width / 2, Printer.ScaleHeight / 2 -
pctDraw1.Height / 2
Printer.EndDoc 'screenshots picture box and prints it to the page centrally
 Call frmDistanceVectors.DrawAxis(pctDraw1, xmax, xmin, ymax, ymin)
```

```
End Sub
Public Sub iniprinter()
Printer.Print
End Sub
```

```
'Small sections of this code were from an API
Public Sub AddPictureBox(Box As PictureBox, _
              Optional x As Single = 0, _
              Optional Y As Single = 0)
Dim rv As Long
Dim ar As Boolean
   On Error GoTo Exit_Sub
   With Box
       ar = .AutoRedraw
         .AutoRedraw = True
                 rv = SendMessage(.hwnd, WM_PAINT, .hDC, 0)
    rv = SendMessage(.hwnd, WM_PRINT, .hDC, _
      PRF_CHILDREN Or PRF_CLIENT Or PRF_OWNED)
          .Picture = .Image
          Printer.PaintPicture .Picture, x, Y
               Box.Line (0, 0)-(.ScaleWidth, .ScaleHeight), .BackColor, BF
          .AutoRedraw = ar
  End With
 Exit_Sub:
  If Err.Number Then MsgBox Err.Description, vbOKOnly, "Printer Error!"
 End Sub
```

```
Private Sub cmdQuit_Click()
Unload Me
End Sub
```

```
Private Sub Form_Load()
Call frmDistanceVectors.DrawAxis(pctDraw1, xmax, xmin, ymax, ymin) 'Draws axis
QuestionsAns = 0
Call frmMain.PassName(StudentName)
End Sub
```

```
Public Sub DrawVectors(pctDraw1)
pctDraw1.Line (vectori(1), 0)-(vectori(1) + vectori(1), vectorj(1)) 'Draws the first vector
Call VectorAnim(vectori(), vectorj())
End Sub
```

```
Public Function GetAngle(ByRef Answer As Double, ByRef Pi As Double) As Double
On Error Resume Next 'Error catch to stop division by 0
If Answer = 1 Then
GetAngle = 0
Exit Function
End If
GetAngle = (Atn(-Answer / Sqr(-Answer * Answer + 1)) + 2 * Atn(1)) * (180 / Pi) 'Takes ArcTan to
calculate angle
On Error GoTo 0
GetAngle = Format(GetAngle, "#0.00") 'Formats it to 2 decimal places
End Function
```

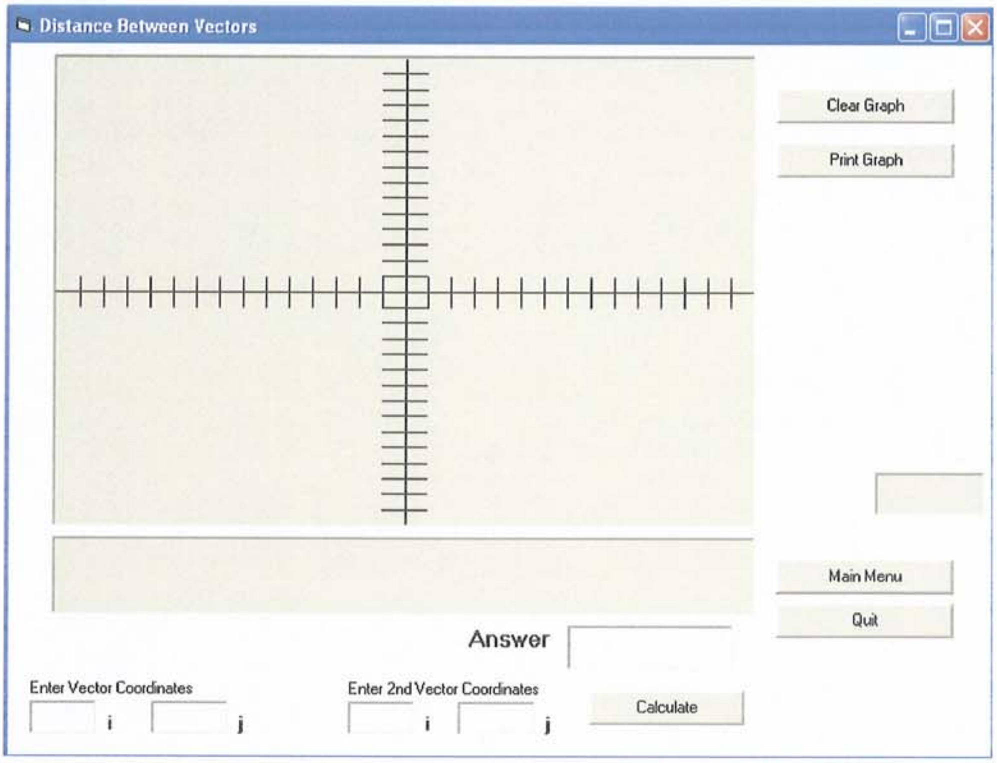
```
Private Sub VectorAnim(ByRef vectori() As Integer, ByRef vectorj() As Integer)
pctDraw1.Line (vectori(1), 0)-(vectori(1) + vectori(1), vectorj(1))
pctDraw1.Line (vectori(1), 0)-(vectori(1) + vectori(2), vectorj(2)), vbRed
```

```
End Sub
```

```
Public Function GetAnswer(ByRef vectori() As Integer, ByRef vectorj() As Integer) As Double
Dim answer1 As Double
Dim answer2 As Double
answer1 = ((vectori(1) * vectori(2)) + (vectorj(1) * vectorj(2))) 'Using c4 angle formulae uses coordinates to calculate the answer to pass to GetAngle formulae
answer2 = (Sqr(vectori(1) ^ 2 + vectorj(1) ^ 2)) * (Sqr(vectori(2) ^ 2 + vectorj(2) ^ 2))
GetAnswer = (answer1 / answer2)
End Function
```

```
Private Sub Timer2_Timer()
pctDraw1.Line (vectori(1), 0)-(vectori(1) + vectori(1), vectorj(1)), vbBlack
Call VectorAnim(vectori(), vectorj())
Timer2.Enabled = False
End Sub
```

<u>Distance Vectors Form</u>



```
Dim vectori(1 To 3) As Integer
Dim vectorj(1 To 3) As Integer
Dim Explanation As String
Dim StudentName As String
Dim validated As Boolean
Dim QuestionsAns As Integer
Const xmax As Integer = 15
Const xmin As Integer = -15
Const ymax As Integer = 15
Const ymin As Integer = -15
Dim Answer As Double
```

```
 Dim StuAns As Double
 Dim correct As Boolean
 Dim LineI As Integer

 Private Declare Function SendMessage Lib "user32.dll" Alias _
   "SendMessageA" (ByVal hwnd As Long, ByVal wMsg As Long, _
   ByVal wParam As Long, ByVal lParam As Long) As Long
 Private Const WM_PAINT = &HF
 Private Const WM_PRINT = &H317
 Private Const PRF_CLIENT = &H4&    ' Draw the window's client area
 Private Const PRF_CHILDREN = &H10& ' Draw all visible child
 Private Const PRF_OWNED = &H20&    ' Draw all owned windows
```

```
 Dim LineJ As Integer
 Dim StuScore As Integer
```

```
 Private Sub cmdCalculate2_Click()
  If txtVectorI1.Text = "" Or txtVectorJ1.Text = "" Or txtVectorI2.Text = "" Or txtVectorJ2.Text = "" Then
      MsgBox ("Please enter a number into the text boxes")
   validated = False
    ElseIf txtStuAns.Text = "" Then
   MsgBox ("Please enter your answer") 'Makes sure all text boxes are filled
    validated = False
    ElseIf Not IsNumeric(txtVectorI1.Text) Or Not IsNumeric(txtVectorJ1.Text) Or Not
IsNumeric(txtVectorI2.Text) Or Not IsNumeric(txtVectorJ2.Text) Or Not IsNumeric(txtStuAns.Text)
Then
   validated = False
   MsgBox ("Please enter a number into the text boxes") 'Makes sure all text boxes are filled with a
number
   ElseIf txtVectorI1.Text > 15 Or txtVectorJ1.Text > 15 Or txtVectorI2.Text > 15 Or txtVectorJ2.Text >
15 Then
   validated = False
   MsgBox ("Please enter a vector between 15 & -15")
    ElseIf txtVectorI1.Text < -15 Or txtVectorJ1.Text < -15 Or txtVectorI2.Text < -15 Or txtVectorJ2.Text
< -15 Then
   validated = False ' makes sure that numbers are between the given range
   MsgBox ("Please enter a vector between 15 & -15")
   Else
vectori(1) = txtVectorI1.Text
vectorj(1) = txtVectorJ1.Text
vectori(2) = txtVectorI2.Text
vectorj(2) = txtVectorJ2.Text
vectorj(3) = vectorj(1)
vectori(3) = vectori(1)
StuAns = txtStuAns.Text
validated = True
End If
If validated = True Then
pctDraw1.Line (vectori(1) + vectori(1), vectorj(1))-(vectori(2) + vectori(2), vectorj(2)), vbBlue
Answer = GetDistance(vectori(), vectorj()) 'Calls GetDistance function
Answer = Format(Answer, "#0.00") 'Formats the answer
LineI = (vectori(2) - vectori(1))
LineJ = (vectorj(2) - vectorj(1))
pctExplanation.Print "The answer is " & Answer; " as Sqr (" & vectori(1); " - " & vectori(2) & ")² + ( " &
vectorj(1); " - " & vectorj(2); ")² = " & Answer; " to 2.d.p"
pctExplanation.Print "The vector of the distance is " & LineI; "i + " & LineJ; "j"
Call DrawVectors(pctDraw1)
```

```
End If
QuestionsAns = QuestionsAns + 1
If StuAns = Answer Then
correct = True
StuScore = StuScore + 1 'If student answer is correct score is incremented by one
pctExplanation.Print "Well done, your answer is correct"
Else
pctExplanation.Print "Sorry that's wrong, try again"
End If
End Sub
```

```
Private Sub cmdClear_Click()
pctDraw1.Cls 'Clears the picture box
pctExplanation.Cls
Call frmDistanceVectors.DrawAxis(pctDraw1, xmax, xmin, ymax, ymin) 'Redraws the axis
Timer1.Enabled = False
End Sub
```

```
Private Sub cmdMainMenu_Click()
frmMain.Show
Filename = App.Path & "\VectorQuestionScores.txt"
Open Filename For Append As #1 'Loads the file and writes the students results to it
Print #1, "Distance Vector Score ="; StuScore; "out of"; QuestionsAns
Close #1
Unload Me
End Sub
```

```
Private Sub cmdPrint_Click() 'Prints a hard copy of the students graph along with an explanation
 iniprinter
  Printer.FontBold = True
 Printer.FontSize = 18
 Printer.Print Spc(5); "Distance Vector Problem"
  Printer.Print
 Printer.FontSize = 12
 Printer.Print "Student Name:"; Spc(3); StudentName
 Printer.Print
 Printer.FontBold = False
 Printer.Print "The vector graph of "; vectori(3); "i + "; vectorj(3); "j and "; vectori(2); "i + "; vectorj(2);
"j"
 Printer.Print
 Printer.Print "The answer is "; Answer;
 Printer.Print "as "; "Sqr (" & vectori(3); " - " & vectori(2) & ")² + ( " & vectorj(3); " - " & vectorj(2); ")² =
"; Answer; " to 2.d.p"
 Printer.Print
 Printer.Print "The vector of the distance is " & LineI; "i + " & LineJ; "j"
 Printer.Print
 If correct = True Then
 Printer.Print "Well done you correctly answered this question"
 Else
 Printer.Print "Unfortunately your answer was wrong, either try again or make sure to contact your
teacher"
 Printer.Print "with the relevant questions."
End If
AddPictureBox pctDraw1, Printer.ScaleWidth / 2 - pctDraw1.Width / 2, Printer.ScaleHeight / 2 -
pctDraw1.Height / 2 'Screenshots the picture box and prints it centrally in the form
Call frmDistanceVectors.DrawAxis(pctDraw1, xmax, xmin, ymax, ymin)
 Printer.EndDoc
```

```
End Sub
```

```
Public Sub iniprinter()
Printer.Print
End Sub
```

```
'Small sections of this code were from an API
Public Sub AddPictureBox(Box As PictureBox, _
                Optional x As Single = 0, _
                Optional Y As Single = 0)
Dim rv As Long
Dim ar As Boolean
    On Error GoTo Exit_Sub
    With Box
        ar = .AutoRedraw
          .AutoRedraw = True
                rv = SendMessage(.hwnd, WM_PAINT, .hDC, 0)
      rv = SendMessage(.hwnd, WM_PRINT, .hDC, _
         PRF_CHILDREN Or PRF_CLIENT Or PRF_OWNED)
          .Picture = .Image
            Printer.PaintPicture .Picture, x, Y
          Box.Line (0, 0)-(.ScaleWidth, .ScaleHeight), .BackColor, BF
      .AutoRedraw = ar
   End With
 Exit_Sub:
   If Err.Number Then MsgBox Err.Description, vbOKOnly, "Printer Error!"
  End Sub
```

— reference to API code

this is a section of API from www.vbmonster.com I have adjusted it slightly to fit my program

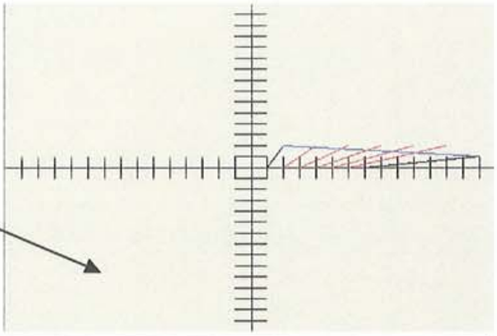
```
Private Sub cmdQuit_Click()
Unload Me
End Sub
```

```
Private Sub Form_Load()
QuestionsAns = 0
StuScore = 0
Call frmMain.PassName(StudentName)
Call DrawAxis(pctDraw1, xmax, xmin, ymax, ymin)
End Sub
```

```
Public Sub DrawAxis(pctDraw1, xmax, xmin, ymax, ymin) 'Draws axis using a counter
Dim counter As Integer
Dim x As Integer
Dim Y As Integer
pctDraw1.Scale (xmin, xmax)-(ymax, ymin)
pctDraw1.Line (xmin, 0)-(xmax, 0)
For counter = (xmin + 1) To (xmax - 1)
pctDraw1.Line (counter, -1)-(counter, 1)
Next counter
pctDraw1.Line (0, ymin)-(0, ymax)
For counter = (ymin + 1) To (ymax - 1)
pctDraw1.Line (-1, counter)-(1, counter)
Next counter
pctDraw1.ForeColor = vbBlack
End Sub
```

```
Public Sub DrawVectors(pctDraw1)
pctDraw1.Line (vectori(1), 0)-(vectori(1) + vectori(1), vectorj(1)), vbBlack 'Draws first vector
```

```
Call VectorAnim(vectori(), vectorj())
End Sub
Public Sub VectorAnim(ByRef vectori() As Integer, ByRef vectorj() As Integer) 'Recursive algorithm to map vector1 to vector2
If vectori(1) < vectori(2) And vectorj(1) < vectorj(2) Then
vectori(1) = vectori(1) + 1
vectorj(1) = vectorj(1) + 1
Timer1.Enabled = True
pctDraw1.Line (vectori(1), 0)-(vectori(1) + vectori(1), vectorj(1)), vbRed
Call VectorAnim(vectori(), vectorj()) 'Recursion, calls itself again
ElseIf vectori(1) = vectori(2) And vectorj(1) < vectorj(2) Then
vectorj(1) = vectorj(1) + 1
Timer1.Enabled = True
pctDraw1.Line (vectori(1), 0)-(vectori(1) + vectori(1), vectorj(1)), vbRed
Call VectorAnim(vectori(), vectorj())
ElseIf vectori(1) < vectori(2) And vectorj(1) = vectorj(2) Then
vectori(1) = vectori(1) + 1
Timer1.Enabled = True
pctDraw1.Line (vectori(1), 0)-(vectori(1) + vectori(1), vectorj(1)), vbRed
Call VectorAnim(vectori(), vectorj())
ElseIf vectori(1) > vectori(2) And vectorj(1) > vectorj(2) Then
vectori(1) = vectori(1) - 1
vectorj(1) = vectorj(1) - 1
Timer1.Enabled = True
pctDraw1.Line (vectori(1), 0)-(vectori(1) + vectori(1), vectorj(1)), vbRed
Call VectorAnim(vectori(), vectorj())
ElseIf vectori(1) = vectori(2) And vectorj(1) > vectorj(2) Then
vectorj(1) = vectorj(1) - 1
Call VectorAnim(vectori(), vectorj())
Timer1.Enabled = True
pctDraw1.Line (vectori(1), 0)-(vectori(1) + vectori(1), vectorj(1)), vbRed
ElseIf vectori(1) > vectori(2) And vectorj(1) = vectorj(2) Then
vectori(1) = vectori(1) - 1
Timer1.Enabled = True
pctDraw1.Line (vectori(1), 0)-(vectori(1) + vectori(1), vectorj(1)), vbRed
Call VectorAnim(vectori(), vectorj())
ElseIf vectori(1) > vectori(2) And vectorj(1) < vectorj(2) Then
vectori(1) = vectori(1) - 1
vectorj(1) = vectorj(1) + 1
Timer1.Enabled = True
pctDraw1.Line (vectori(1), 0)-(vectori(1) + vectori(1), vectorj(1)), vbRed
Call VectorAnim(vectori(), vectorj())
ElseIf vectori(1) < vectori(2) And vectorj(1) > vectorj(2) Then
vectori(1) = vectori(1) + 1
vectorj(1) = vectorj(1) + 1
Timer1.Enabled = True
pctDraw1.Line (vectori(1), 0)-(vectori(1) + vectori(1), vectorj(1)), vbRed
Call VectorAnim(vectori(), vectorj())
End If
End Sub
```

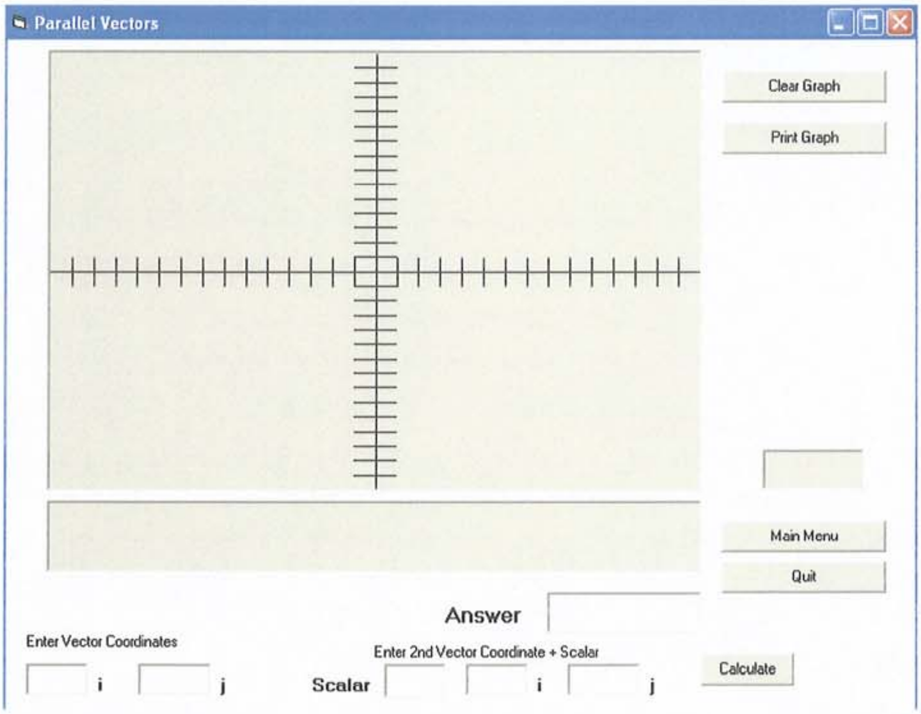


```
Public Function GetDistance(ByRef vectori() As Integer, ByRef vectorj() As Integer)
GetDistance = Sqr(((vectori(1) - vectori(2)) ^ 2) + ((vectorj(1) - vectorj(2)) ^ 2)) 'C4 maths formulae to calculate the distance between 2 vectors
```

```
End Function
```

```
Private Sub Timer1_Timer()
pctDraw1.Line (vectori(1), 0)-((vectori(1) + vectori(1)), vectorj(1)), vbBlack
Timer1.Enabled = False
End Sub
```

Parallel Vectors Form

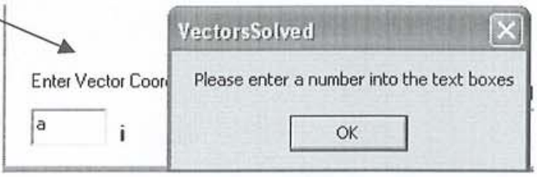


```
Dim vectori(1 To 3) As Integer
Dim vectorj(1 To 3) As Integer
Dim validated As Boolean
Dim StudentName As String
Const xmax As Integer = 15
Const xmin As Integer = -15
Const ymax As Integer = 15
Const ymin As Integer = -15
Dim StuAns As String
```

```
Private Declare Function SendMessage Lib "user32.dll" Alias _
  "SendMessageA" (ByVal hwnd As Long, ByVal wMsg As Long, _
  ByVal wParam As Long, ByVal lParam As Long) As Long
Private Const WM_PAINT = &HF
Private Const WM_PRINT = &H317
Private Const PRF_CLIENT = &H4&
Private Const PRF_CHILDREN = &H10&
Private Const PRF_OWNED = &H20&
```

```
Dim QuestionsAns As Integer
Dim correct As Boolean
Dim StuScore As Integer
Dim Explanation As String
Dim Scalar As Single
```

```
Private Sub cmdCalculate_Click()
   If txtVectorI1.Text = "" Or txtVectorJ1.Text = "" Or txtVectorI2.Text = "" Or txtVectorJ2.Text = "" Or
txtScalar.Text = "" Then
      MsgBox ("Please enter a number into the text boxes")
   validated = False
    ElseIf txtStuAns.Text = "" Then 'Makes sure that something is input into all of the textboxes
   MsgBox ("Please enter your answer")
   validated = False
   ElseIf Not IsNumeric(txtVectorI1.Text) Or Not IsNumeric(txtVectorJ1.Text) Or Not
IsNumeric(txtVectorI2.Text) Or Not IsNumeric(txtScalar.Text) Or Not IsNumeric(txtVectorJ2.Text)
Then
   validated = False
   MsgBox ("Please enter a number into the text boxes") 'Makes sure that only numbers can be
entered into the text boxes
```



VectorsSolved

Please enter a number into the text boxes

```
   ElseIf txtVectorI1.Text > 15 Or txtScalar.Text > 15 Or
txtVectorJ1.Text > 15 Or txtVectorI2.Text > 15 Or
txtVectorJ2.Text > 15 Then
   MsgBox ("Please enter a number into the text
boxes")
   validated = False
   MsgBox ("Please enter a vector between 15 & -15") 'Makes sure that the numbers are within the
given range
    ElseIf txtVectorI1.Text < -15 Or txtScalar.Text < -15 Or txtVectorJ1.Text < -15 Or txtVectorI2.Text < -
15 Or txtVectorJ2.Text < -15 Then
   validated = False
   MsgBox ("Please enter a vector between 15 & -15")
   Else
   StuAns = txtStuAns.Text
vectori(1) = txtVectorI1.Text
vectorj(1) = txtVectorJ1.Text
vectori(2) = txtVectorI2.Text * txtScalar.Text
vectorj(2) = txtVectorJ2.Text * txtScalar.Text
vectori(3) = vectori(1)
vectorj(3) = vectorj(1)
validated = True
End If
   If validated = True Then
Parallel = GetParallel(vectori(), vectorj()) 'calls GetParallel function
If Parallel = True Then
pctExplanation.Print "The two vectors are parallel, as you can see in the diagram this is because " &
vectori(1) & " X " & vectorj(2); " - " & vectorj(1); "X " & vectori(2); " = 0"; vbCrLf; "This shows that the
2nd vector coordinates are a scalar multiplication of the 1st vector coordinates"
Else
pctExplanation.Print "The two vectors are not parallel, the 2nd vector coordinates aren't a correct
scalar multiplication of the first."
End If
Call DrawVectors(pctDraw1) 'Calls the DrawVectors Function

If StuAns = Parallel Then
correct = True
StuScore = StuScore + 1 'If the student has got the answer correct their score is incremented by one
pctExplanation.Print "Well done, your answer is correct"
Else
pctExplanation.Print "Sorry that's wrong, try again"
End If
```

```
QuestionsAns = QuestionsAns + 1 '
'Questions answered is incremented by one each time a student enters an answer
 End If

Explanation = pctExplanation
End Sub

Private Sub cmdClear_Click()
pctDraw1.Cls
pctExplanation.Cls 'Clears the graph
Call frmDistanceVectors.DrawAxis(pctDraw1, xmax, xmin, ymax, ymin) 'Redraws the graph
End Sub

Private Sub cmdMainMenu_Click()
frmMain.Show
Filename = App.Path & "\VectorQuestionScores.txt"
Open Filename For Append As #1 'Writes the results of how the student has scored to file
Print #1, "Parallel vectors score ="; StuScore; "out of"; QuestionsAns
Close #1
Unload Me
End Sub
```

```
Private Sub cmdPrint_Click() 'prints the graph along with an explanation/answer for the student
 iniprinter
 Printer.FontBold = True
 Printer.FontSize = 18
 Printer.Print Spc(5); "Parallel Vector Problem" 'Writes heading
 Printer.Print
 Printer.FontSize = 12
 Printer.Print "Student Name:  "; Spc(3); StudentName
 Printer.Print
 Printer.FontBold = False
 Printer.Print "The vector graph of "; vectori(3); "i + "; vectorj(3); "j and "; vectori(2); "i + "; vectorj(2);
"j"
 Printer.Print
 If Parallel = True Then
 Printer.Print "The two vectors are parallel, as you can see in the diagram this is because "
 Printer.Print vectori(3) & " X " & vectorj(2); " - " & vectorj(3); "X " & vectori(2); " = 0"; vbCrLf;
 Printer.Print "This shows that the 2nd vector coordinates are a scalar multiplication of the 1st vector
coordinates"
 Else
 Printer.Print "The two vectors are not parallel, the 2nd vector coordinates aren't a correct scalar
multiplication of the first."
 Printer.Print
 Printer.Print
 End If
 If correct = True Then
 Printer.Print "Well done you correctly answered this question"
 Else
 Printer.Print "Unfortunately your answer was wrong, either try again or make sure to contact your
teacher "
 Printer.Print "with the relevant questions."
End If

AddPictureBox pctDraw1, Printer.ScaleWidth / 2 - pctDraw1.Width / 2, Printer.ScaleHeight / 2 -
pctDraw1.Height / 2
 Printer.EndDoc 'Code that screenshots the graph and prints it out into central form
```

```
End Sub
Public Sub iniprinter()
Printer.Print
End Sub
```

```
'Small sections of this code were from an API
Public Sub AddPictureBox(Box As PictureBox, _
            Optional x As Single = 0, _
            Optional Y As Single = 0)
Dim rv As Long
Dim ar As Boolean
   On Error GoTo Exit_Sub
   With Box
      ar = .AutoRedraw
       .AutoRedraw = True
                              rv = SendMessage(.hwnd, WM_PAINT, .hDC, 0)
    rv = SendMessage(.hwnd, WM_PRINT, .hDC, _
      PRF_CHILDREN Or PRF_CLIENT Or PRF_OWNED)
         .Picture = .Image
          Printer.PaintPicture .Picture, x, Y
             Box.Line (-15, 15)-(.ScaleWidth, .ScaleHeight), .BackColor, BF
          .AutoRedraw = ar
  End With
 Exit_Sub:
  If Err.Number Then MsgBox Err.Description, vbOKOnly, "Printer Error!"
 End Sub
```

```
Private Sub cmdQuit_Click()
Unload frmParallelVectors
End Sub
```

```
Private Sub Form_Load()
Call frmDistanceVectors.DrawAxis(pctDraw1, xmax, xmin, ymax, ymin)
QuestionsAns = 0
StuScore = 0
Call frmMain.PassName(StudentName)
End Sub
```

```
Public Sub DrawVectors(pctDraw1) 'Draws the first vector
pctDraw1.Line (vectori(1), 0)-((vectori(1) + vectori(1)), vectorj(1)), vbBlack
Call VectorAnim(vectori(), vectorj())
End Sub
```

```
Private Sub Timer1_Timer()
pctDraw1.Line (vectori(1), 0)-((vectori(1) + vectori(1)), vectorj(1)), vbBlack
Call VectorAnim(vectori(), vectorj())
Timer1.Enabled = False
End Sub
```

```
Public Sub VectorAnim(ByRef vectori() As Integer, ByRef vectorj() As Integer) 'Recursive algorithm
used to map vector1 to vector 2
If vectori(1) < vectori(2) And vectorj(1) < vectorj(2) Then
vectori(1) = vectori(1) + 1
vectorj(1) = vectorj(1) + 1
Timer1.Enabled = True
pctDraw1.Line (vectori(1), 0)-(vectori(1) + vectori(1), vectorj(1)), vbRed
```

```
Call VectorAnim(vectori(), vectorj()) 'Uses recursion to call upon itself to draw the animated line
ElseIf vectori(1) = vectori(2) And vectorj(1) < vectorj(2) Then
vectorj(1) = vectorj(1) + 1
Timer1.Enabled = True
pctDraw1.Line (vectori(1), 0)-(vectori(1) + vectori(1), vectorj(1)), vbRed
Call VectorAnim(vectori(), vectorj())
ElseIf vectori(1) < vectori(2) And vectorj(1) = vectorj(2) Then
vectori(1) = vectori(1) + 1
Timer1.Enabled = True
pctDraw1.Line (vectori(1), 0)-(vectori(1) + vectori(1), vectorj(1)), vbRed
Call VectorAnim(vectori(), vectorj())
ElseIf vectori(1) > vectori(2) And vectorj(1) > vectorj(2) Then
vectori(1) = vectori(1) - 1
vectorj(1) = vectorj(1) - 1
Timer1.Enabled = True
pctDraw1.Line (vectori(1), 0)-(vectori(1) + vectori(1), vectorj(1)), vbRed
Call VectorAnim(vectori(), vectorj())
ElseIf vectori(1) = vectori(2) And vectorj(1) > vectorj(2) Then
vectorj(1) = vectorj(1) - 1
Call VectorAnim(vectori(), vectorj())
Timer1.Enabled = True
pctDraw1.Line (vectori(1), 0)-(vectori(1) + vectori(1), vectorj(1)), vbRed
ElseIf vectori(1) > vectori(2) And vectorj(1) = vectorj(2) Then
vectori(1) = vectori(1) - 1
Timer1.Enabled = True
pctDraw1.Line (vectori(1), 0)-(vectori(1) + vectori(1), vectorj(1)), vbRed
Call VectorAnim(vectori(), vectorj())
ElseIf vectori(1) > vectori(2) And vectorj(1) < vectorj(2) Then
vectori(1) = vectori(1) - 1
vectorj(1) = vectorj(1) + 1
Timer1.Enabled = True
pctDraw1.Line (vectori(1), 0)-(vectori(1) + vectori(1), vectorj(1)), vbRed
Call VectorAnim(vectori(), vectorj())
ElseIf vectori(1) < vectori(2) And vectorj(1) > vectorj(2) Then
vectori(1) = vectori(1) + 1
vectorj(1) = vectorj(1) + 1
Timer1.Enabled = True
pctDraw1.Line (vectori(1), 0)-(vectori(1) + vectori(1), vectorj(1)), vbRed
Call VectorAnim(vectori(), vectorj())
End If
End Sub
```

```
Public Function GetParallel(ByRef vectori() As Integer, ByRef vectorj() As Integer) As Boolean
If (vectori(1) * vectorj(2)) - (vectorj(1) * vectori(2)) = 0 Then 'Formula used from C4 maths to calculate
if 2 vectors are parallel
 GetParallel = True
 Else
 GetParallel = False
End If
End Function
```

Module

```
Dim Name As String
Dim Filename As String
```

```
Public Sub WriteName()
Filename = App.Path & "\VectorQuestionScores.txt"
```

```
Name = frmMain.txtName
Open Filename For Append As #1 'Opens the file
Print #1, Name 'Writes student name to file
Close #1
End Sub
```

```
Public Function PassName(StuName)
StuName = Name
End Function
```

Testing

Testing

As stated in the Design part of my project I will use a combination of testing strategies in order to make sure my program is fully functional; a mixture of both black and white box testing and unit, integrated and a full systems test to make sure that every aspect of the system runs correctly before Mr Mason uses the system with his students.

Main Form

Test Number	Test Data	Purpose	Expected Result	Actual Result	Reference
1	Splash screen links to Main Form within 5 seconds	To introduce the user to the program before starting up.	The splash screen shows for 5 seconds, the main form then loads.	Splash screen shows for 5 seconds before main menu appearing	Proof – Teacher Signature
2	Click cmdParallel	To take the user to the Parallel Vectors page.	Main menu closes and Parallel Vector page opens.	Parallel Vector page opens, Main Menu disappears.	Proof - Teacher Signature
3	Click cmdAngle	To take the user to the Calculate Angle page.	Main menu closes and Angle page opens.	Angle Vectors page opens, Main Menu page disappears.	Proof - Teacher Signature
4	Click cmdDistance	To take the user to the Distance Between Vectors page.	Main Menu closes and Distance Between Vectors page opens.	Distance Vector page opens, Main Menu page disappears.	Proof - Teacher Signature
5	Click cmdQuit	To quit the application.	Application closes.	Application ends.	Proof - Teacher Signature

Parallel Vectors Form

Test Number	Test Data	Purpose	Expected Result	Actual Result	Reference
6	X Axis	Check x axis is set between the range -15 & 15	X axis is set between the range	X axis is correctly declared as a constant between	Screen shot.

				15 & -15.	
7	Y Axis	Check y axis is set between the range -15 & 15	Y axis is set between the range	Y axis is correctly declares as a constant between 15 & -15.	Screen shot
8	X axis	Check X axis is plot correctly.	X axis is plot correctly	X axis is plotted correctly	Screen shot
9	Y Axis	Check Y axis is plot correctly	Y axis is plot correctly	Y Axis is plotted correctly	Screenshot
10	Vectori1 = 2	Check x coordinate of vector is correctly drawn	X Coordinate is correctly drawn	X Coordinate is correctly drawn	Screenshot
11	Vectorj1 = 3	Check y coordinate of vector is correctly drawn	Y Coordinate is correctly drawn	Y Coordinate is correctly drawn	Screenshot
12	Vectori2 = 4	Check x coordinate of 2nd vector is correctly drawn	X coordinate is correctly drawn	2nd X Coordinate is correctly drawn	Screenshot
13	Vectorj2 = 5	Check y coordinate of 2nd vector is correctly drawn	Y coordinate is correctly drawn	2nd Y Coordinate is correctly drawn	Screenshot
14	Scalar = 2	Check x coordinate of vector 2 is correctly drawn when multiplied by scalar	X Coordinate is correctly drawn	X Coordinate is correctly drawn using scalar.	Screenshot
15	Scalar = 2	Check y coordinate of vector 2 is correctly drawn	Y Coordinate is correctly drawn	Y Coordinate is correctly drawn	Screenshot

		when multiplied by scalar		using scalar.	
16	Explanation	Check Explanation shows up correctly step by step	Explanation shows up correctly step by step	Explanation shows up correctly.	Screenshot
17	Answer	Check answer is correct and shows on screen	Answer is correct and is shown on screen	Answer is correct and shows up on screen.	Screenshot
18	Click cmdMainMenu	Takes the user back to the main menu	Parallel vector page closes, main menu page opens	Parallel vector page closes as expected and main menu page opens.	Proof - Teacher Signature
19	Click cmdClear	Clears the vector from the screen and allows the user to have another go	Vector, explanation and answer is cleared from the screen and gives the user a chance for another go	As expected vector explanation & answer disappear and allow the user to start again.	Proof – Teacher Signature
20	Click cmdCalculate	Prompts the user to input their coordinates if " "	User is prompted to input their coordinates.	Message box appears prompting the input of vector coordinates	Screen shot
21	Click cmdPrint	Allows the user to print off their form	Printout prints.	Print out prints incorrectly, code edited.	Hard copies.
22	Click cmdQuit	Allows the user to	Program closes	Program	Proof - Teacher

		exit the program		ends	Signature

Angle Vectors Form

Test Number	Test Data	Purpose	Expected Result	Actual Result	Reference
23	X Axis	Check x axis is set between the range -15 & 15.	X axis is set between the range.	X axis is correctly set	Screenshot
24	Y Axis	Check y axis is set between the range -15 & 15	Y axis is set between the range	Y Axis is correctly set	Screenshot
25	X axis	Check X axis is plot correctly.	X axis is plot correctly	X axis is plot correctly	Screenshot
26	Y Axis	Check Y axis is plot correctly	Y axis is plot correctly	Y axis is plot correctly	Screenshot
27	Vectori1 = 1	Check x coordinate of vector is correctly drawn	X Coordinate is correctly drawn	X coordinate is correctly drawn	Screenshot
28	Vectorj1 = 2	Check y coordinate of vector is correctly drawn	Y Coordinate is correctly drawn	Y Coordinate is correctly drawn	Screenshot
29	Vectori2 = 4	Check x coordinate of 2nd vector is correctly drawn	X coordinate is correctly drawn	2nd x coordinate is correctly drawn	Screenshot
30	Vectorj2 = 1	Check y coordinate of 2nd	Y coordinate is	2nd y coordinate	Screenshot

		vector is correctly drawn	correctly drawn	is correctly drawn	
31	Angle Size	Check angle size is drawn correctly	Angle size is drawn correctly	Angle size is drawn correctly	Screenshot
32	Formulae	Check correct formulae is being used	Correct formulae is being used	Mistake in typing of formulae, needs a catch clause – added.	Screenshot
33	Explanation	Check Explanation shows up correctly step by step	Explanation shows up correctly step by step	Explanation shows up as expected	Screenshot
34	Answer	Check answer is correct and shows on screen	Answer is correct and is shown on screen	Answer does not show up correctly due to mistake in formula – changed so that the answer is no longer being divided by 0.	Screenshot
35	Click cmdMainMenu	Takes the user back to the main menu	Parallel vector page closes, main menu page opens	Parallel Vector Page Closes and Main Menu page opens	Proof – Teacher Signature
36	Click cmdClear	Clears the vector from the screen and allows the user to have	Vector, explanation and answer is cleared from the screen and gives the	Picture Boxes containing - vector, explanation	Proof - Teacher Signature

		another go	user a chance for another go	and answer are correctly cleared.	
37	Click cmdCalculate	Prompts the user to input their coordinates when " "	User is prompted to input their coordinates.	Message Box appears asking the user to input their coordinates	Screenshot
38	Click cmdPrint	Allows the user to print off their form	Printer prints out correct output.	Print out doesn't print out correctly, code edited so that the print out is centred in the middle of the page with the correct headings.	Hard copy.
39	Click cmdQuit	Allows the user to exit the program	Program closes	Program ends.	Proof – Teacher Signature

<u>Distance Between 2 Vectors Form</u>

Test Number	**Test Data**	**Purpose**	**Expected Result**	**Actual Result**	**Reference**
40	X Axis will appear correctly	Check x axis is set between the range -15 & 15.	X axis is set between the range -15 & 15	X axis is correctly set between range.	Screenshot
41	Y Axis will appear correctly	Check y axis is set between the	Y axis is set between the	Y axis is correctly set between	Screenshot

44	Vectori1 = 2	Check x coordinate of vector is correctly drawn	X Coordinate is correctly drawn	X Coordinate is correctly drawn	Screenshot
45	Vectorj1 = 3	Check y coordinate of vector is correctly drawn	Y Coordinate is correctly drawn	Y Coordinate is correctly drawn	Screenshot
46	Vectori2 = 4	Check x coordinate of 2nd vector is correctly drawn	X coordinate is correctly drawn	2nd X Coordinate is correctly drawn	Screenshot
47	Vectorj2 = 5	Check y coordinate of 2nd vector is correctly drawn	Y coordinate is correctly drawn	2nd Y Coordinate is correctly drawn	Screenshot
48	Line	Check line is drawn correctly between the 2 vectors	Line is drawn correctly	Line is correctly drawn in different colour.	Screenshot
49	LineSize	Check the vector of the line is displayed	Vector of line is displayed	Vector of line isn't displayed, modification made in coding to calculate distance of line & print to explanation picture box.	Screenshot
50	Explanation	Check Explanation shows up correctly step by step	Explanation shows up correctly step by step	Explanation is correctly shown	Screenshot
51	Answer	Check answer is correct and	Answer is correct and is	Answer is correctly	Screenshot

		shows on screen	shown on screen	shown	
52	Click cmdMainMenu	Takes the user back to the main menu	Distance vector page closes, main menu page opens	Main menu page opens, distance vector page closes	Proof – Teacher Signature
53	Click cmdClear	Clears the vector from the screen and allows the user to have another go	Vector, explanation and answer is cleared from the screen and gives the user a chance for another go	Picture boxes containing explanation & vector graph are cleared	Proof – Teacher Signature
54	Click cmdInputVectors	Prompts the user to input their coordinates.	User is prompted to input their coordinates.	User is prompted to input their coordinates	Screenshots
55	Click cmdPrint	Allows the user to print off their form	Printer prints off correct output.	Print out prints incorrectly.	Hard copies
56	Click cmdQuit	Allows the user to exit the program	Program closes	Program ends	Proof – Teacher Signature
57	Line Colour	Check line colour of distance is set to Blue	Line Colour is set to Blue	Line Colour is set to blue	Screenshot

Integrated Testing

Parallel Vectors Form

Test Number	Test Data	Purpose	Expected Result	Actual Result	Reference
58	X & Y Axis will appear correctly	Check X & Y axis are plotted correctly	X & Y axis are plotted correctly	X & Y axis correctly plotted	Screenshot
59	Vectori1 = 2 & Vectorj1 = 3	Check that the vector is correctly drawn using both the coordinates	Vector is correctly drawn	Vector 1 correctly drawn	Screenshot
60	Vectori2 = 4 & Vectorj2 = 5 & Scalar = 1	Check that the 2nd vector is correctly drawn	2nd vector is drawn correctly applying the scalar	Vector 2 correctly drawn	Screenshot
61	Drawing of vectors & Explanation	Check that as the vector is 2nd vector is drawn the explanation begins to show	Explanation begins to show step-by-step	Explanation is shown correctly with drawing of graph	Screenshot
62	Answer & Formulae	Check that answer & formulae display at the same time	Answer & formulae display together	Answer & formulate correctly displayed	Screenshot

Angle Vectors Form

Test Number	**Test Data**	**Purpose**	**Expected Result**	**Actual Result**	**Reference**
63	X & Y Axis will appear correctly	Check X & Y axis are plotted correctly	X & Y axis are plotted correctly	X & Y axis are plotted correctly	Screenshot

64	Vectori1, Vectorj1, Vectori2, Vectorj2,Angle Size, Formulae	Check that the vectors are drawn together and meet at the correct angle size	Vector is correctly drawn with correct angle size	Angle size is correctly drawn along with correct vectors	Screenshot
65	Drawing of vectors & Explanation	Check that once the vectors have finished drawing the explanation shows	Explanation begins to show step-by-step	Explanation and drawing of the vectors are displayed correctly	Screenshot
66	Answer & Formulae	Check that answer & formulae display at the same time	Answer & formulae display together	Answer & Formulae are correctly shown	Screenshot

Distance Between 2 Vectors Form

Test Number	Test Data	Purpose	Expected Result	Actual Result	Reference
67	X & Y Axis	Check X & Y axis are plotted correctly	X & Y axis are plotted correctly	X & Y axis are plotted correctly	Screenshot
68	Vectori1 = 2 & Vectorj1 = 3	Check that the vector is correctly drawn using both the coordinates	Vector is correctly drawn	Vector 1 is correctly drawn	Screenshot
69	Vectori2 = 4 & Vectorj2 = 5	Check that the 2nd vector is correctly drawn	2nd vector is drawn correctly	Vector 2 is correctly drawn	Screenshot
70	Line & Line Colour	Check that the line is drawn in the correct colour	Line is drawn in the correct colour	Line colour is correctly drawn in blue	Screenshot

71	Drawing of line & Explanation	Check that as the line between the 2 vectors is being drawn the explanation shows	Explanation begins to show step-by-step	Explanation is shown along with vector graphs	Screenshot
72	Answer & Formulae	Check that answer & formulae display at the same time	Answer & formulae display together	Answer & Formulae are displayed correctly	Screenshot

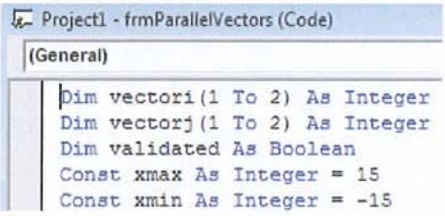
Checking answers & writing /reading file

Test Number	**Test Data**	**Purpose**	**Expected Result**	**Actual Result**	**Reference**
73	Parallel Vectors Form StuAns = True Answer= True	To see whether the system acknowledges the student has got the correct answer	System tells the student they have the answer right and student score incremented by 1	System informs student they have the answer right and result is written to file	Screenshot
74	Parallel Vectors Form StuAns = False Answer= True	To see whether the system can acknowledge the student has the answer wrong.	System informs the student their answer is wrong and no mark is given	System informs the student the answer is wrong and no mark is incremented	Screenshot
75	Angle Vectors Form Stu Ans = 0 Angle = 0	To see whether the system acknowledges the student has got the correct answer	System tells the student they have the answer right and student score incremented by 1	System informs the student that the result is wrong, modification made to code (StuAns=Angle) rather than	Screenshot

				(StuAns=Answer)	
76	Angle Vectors Form StuAns=5.43 Angle=4.97	To make sure that the system recognises when the student has the answer wrong	System tells student their answer is wrong and score is not incremented.	As expected the student is told their answer is wrong and score is not incremented	Screenshot
77	Distance Vectors Form Stu Ans = 2 Answer = 2.00	To make sure that the system recognises the answer is still correct even if not to the decimal place	System informs the student their answer is correct and score is incremented by 1	As expected the student is told their answer is correct and score is incremented by 1 and result is written to file	Screenshot
78	Distance Vectors Form StuAns=4 Answer = 5.1	To make sure the system recognises the answer is wrong	System informs the student their answer is wrong and the score is not incremented	System shows the student that their answer is incorrect and result is written to file	Screenshot
79	Main Menu Form	Check Print Student Results Button works	Student Results are correctly printed	Student results are printed, but with quotation marks, changes are made to the writing process of the file to ensure quotations don't show.	Hard copies.

Evidence of Testing

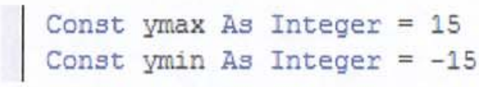
Test 6



```
Dim vectori(1 To 2) As Integer
Dim vectorj(1 To 2) As Integer
Dim validated As Boolean
Const xmax As Integer = 15
Const xmin As Integer = -15
```

Xmax and Xmin are set to the constant values 15 & -15.

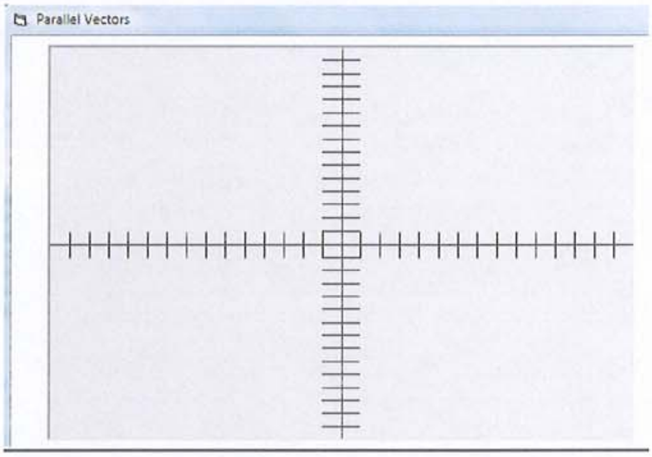
Test 7



```
Const ymax As Integer = 15
Const ymin As Integer = -15
```

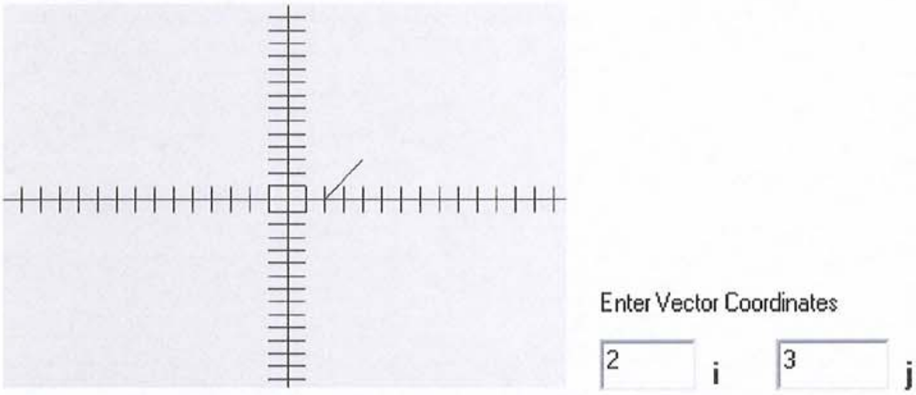
Ymax and Ymin are set to the constant values 15 & -15.

Test 8/9/58



X and Y axis are correctly drawn with 15 separations on each side of the axis.

Test 10/11/59

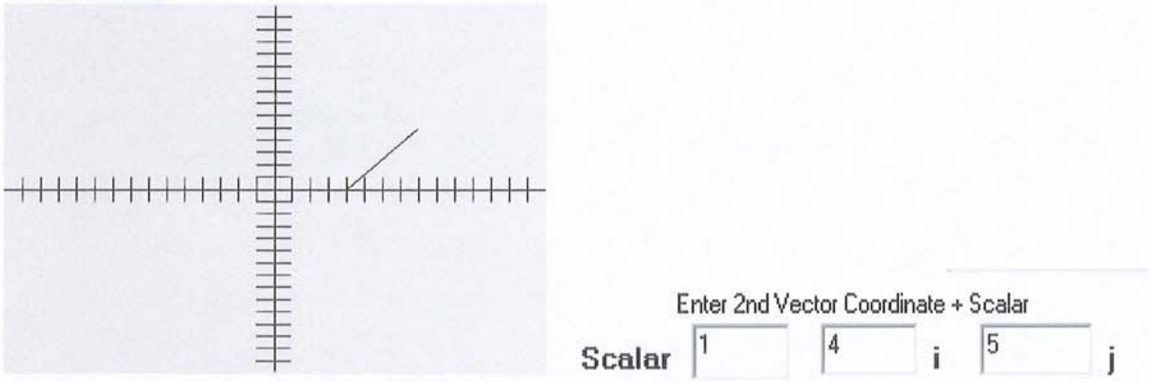


Enter Vector Coordinates

2 i 3 j

Vector is correctly drawn with 2 on the x axis and 3 on the y axis.

Test 12/13/60



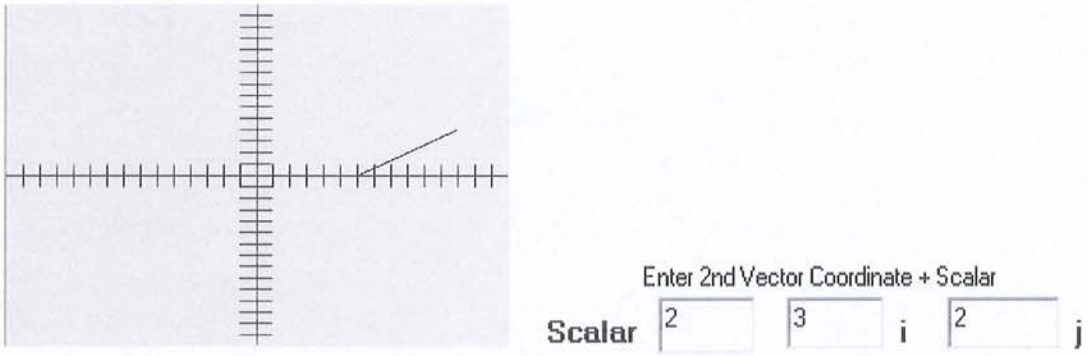
Enter 2nd Vector Coordinate + Scalar

Scalar 1 4 i 5 j

2

__2nd Vector is drawn correctly starting at 4 on the x axis and going up to 5 on the y axis.__

Test 14 & 15

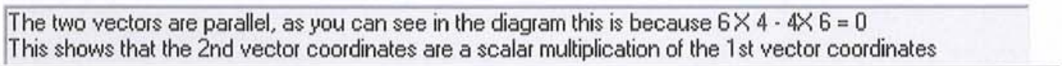


Enter 2nd Vector Coordinate + Scalar

Scalar 2 3 i 2 j

Scalar is correctly applied and vector is drawn from 6 on the x axis to 4 on the y axis.

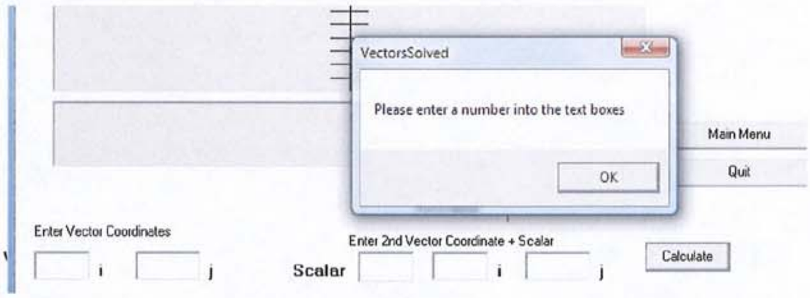
Test 16/17/61/62



The two vectors are parallel, as you can see in the diagram this is because 6 X 4 - 4X 6 = 0
This shows that the 2nd vector coordinates are a scalar multiplication of the 1st vector coordinates

Explanation and answer correctly show up in the picture box.

Test 20

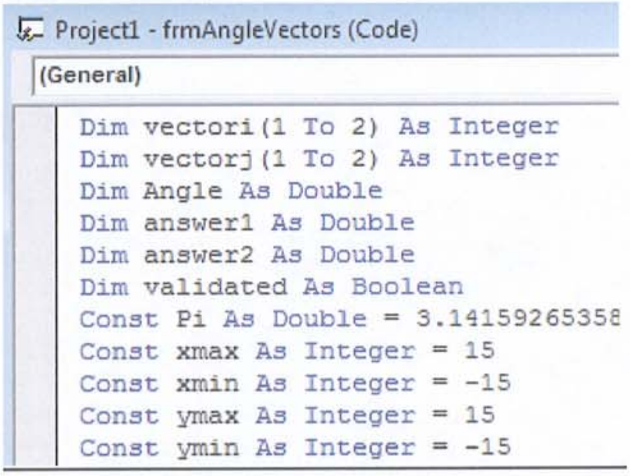


VectorsSolved

Please enter a number into the text boxes

User is correctly prompted.

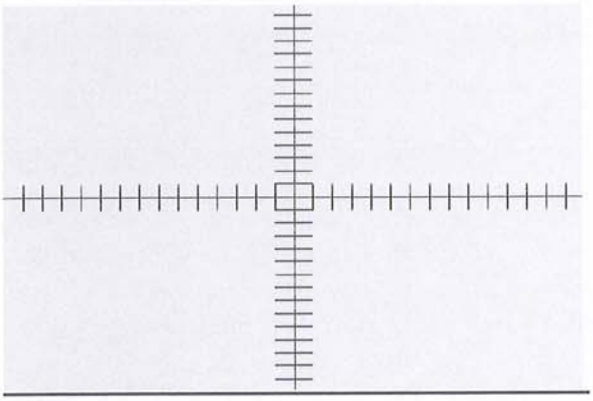
Test 23 & Test 24



```
Dim vectori(1 To 2) As Integer
Dim vectorj(1 To 2) As Integer
Dim Angle As Double
Dim answer1 As Double
Dim answer2 As Double
Dim validated As Boolean
Const Pi As Double = 3.14159265358
Const xmax As Integer = 15
Const xmin As Integer = -15
Const ymax As Integer = 15
Const ymin As Integer = -15
```

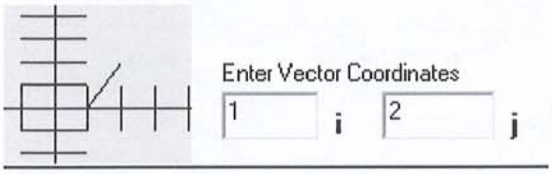
Xmax, xmin, ymax and y min all declared as constants correctly.

Test 25/ 26/63



X and Y axis drawn correctly between the ranges of -15 & 15.

Test 27 & 28

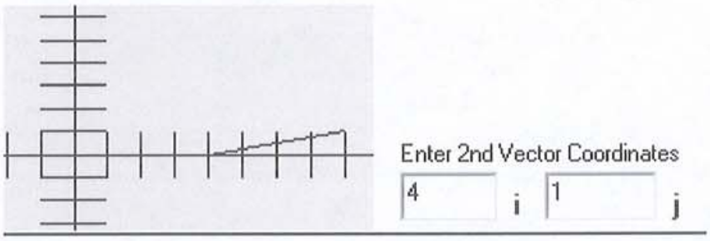


Enter Vector Coordinates

1 i 2 j

X Coordinate is correctly drawn at 1 with Y coordinate correctly drawn at 2

Test 29 & 30

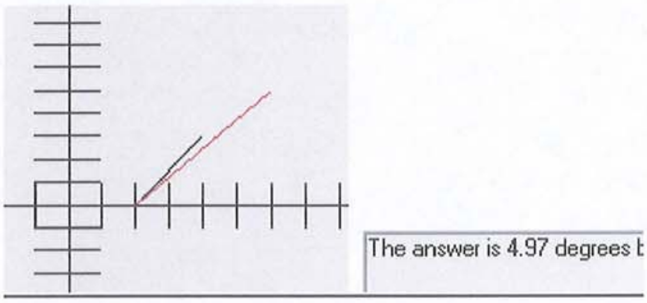


Enter 2nd Vector Coordinates

4 i 1 j

2nd vector coordinates are correctly drawn with x drawn at 4 and y drawn at 1.

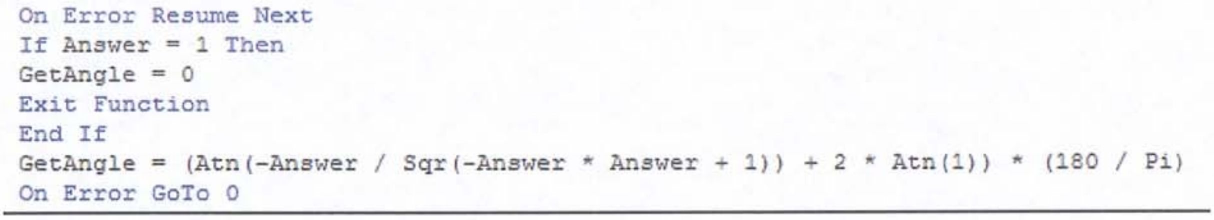
Test 31



The answer is 4.97 degrees b

Angle size is correctly drawn

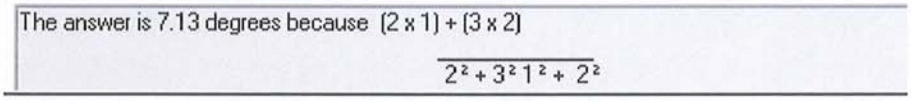
Test 32



```
On Error Resume Next
If Answer = 1 Then
GetAngle = 0
Exit Function
End If
GetAngle = (Atn(-Answer / Sqr(-Answer * Answer + 1)) + 2 * Atn(1)) * (180 / Pi)
On Error GoTo 0
```

Error Clause added to stop division by 0

Test 33/34/64

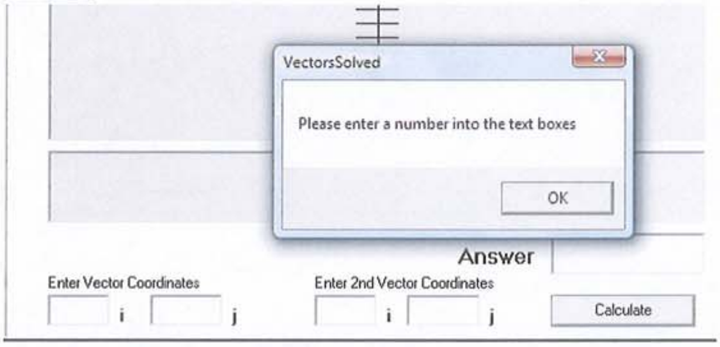


The answer is 7.13 degrees because (2 x 1) + (3 x 2)

$\overline{2^2 + 3^2 1^2 + 2^2}$

Explanation and answer are shown correctly in picture box.

Test 37

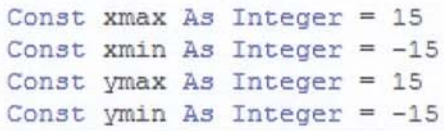


VectorsSolved

Please enter a number into the text boxes

User is prompted to enter their coordinates

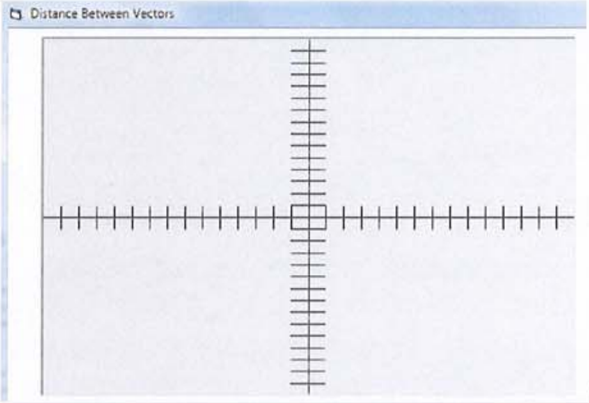
Test 40 & 41



```
Const xmax As Integer = 15
Const xmin As Integer = -15
Const ymax As Integer = 15
Const ymin As Integer = -15
```

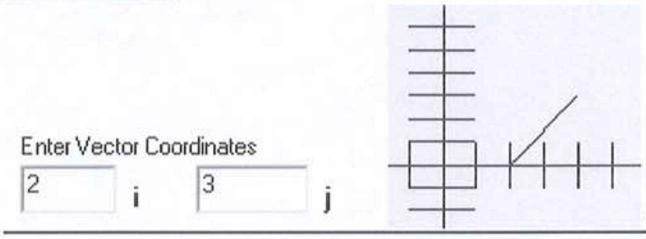
X & Y range are correctly defined as constants between 15 & -15

Test 42/43/67



Axis are plotted correctly

Test 44/45/68

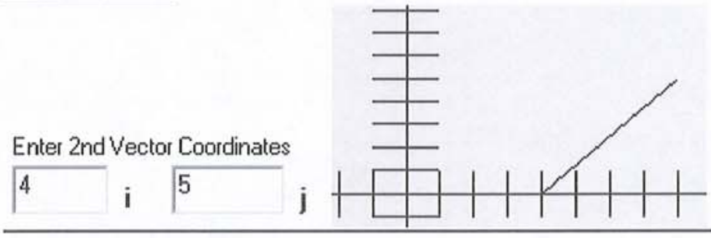


Enter Vector Coordinates

2 i 3 j

Vector is correctly drawn

Test 46/47/69

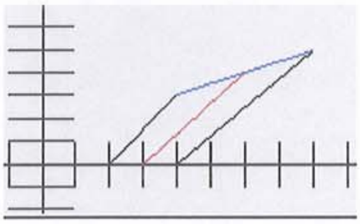


Enter 2nd Vector Coordinates

4 i 5 j

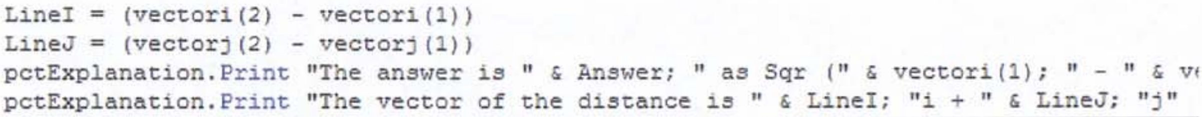
Vector is correctly drawn

Test 48/70



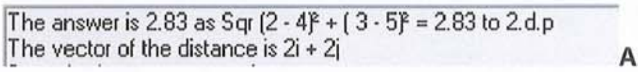
Line between vectors is correctly drawn in blue to show distance.

Test 49/50/51/72



```
LineI = (vectori(2) - vectori(1))
LineJ = (vectorj(2) - vectorj(1))
pctExplanation.Print "The answer is " & Answer; " as Sqr (" & vectori(1); " - " & v
pctExplanation.Print "The vector of the distance is " & LineI; "i + " & LineJ; "j"
```

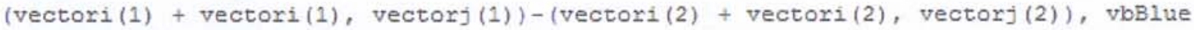
Modification made in coding to calculate vector of line



The answer is 2.83 as Sqr $(2 - 4)^2$ + $(3 - 5)^2$ = 2.83 to 2.d.p
The vector of the distance is 2i + 2j

Answer & Explanation & Vector of the distance is shown correctly

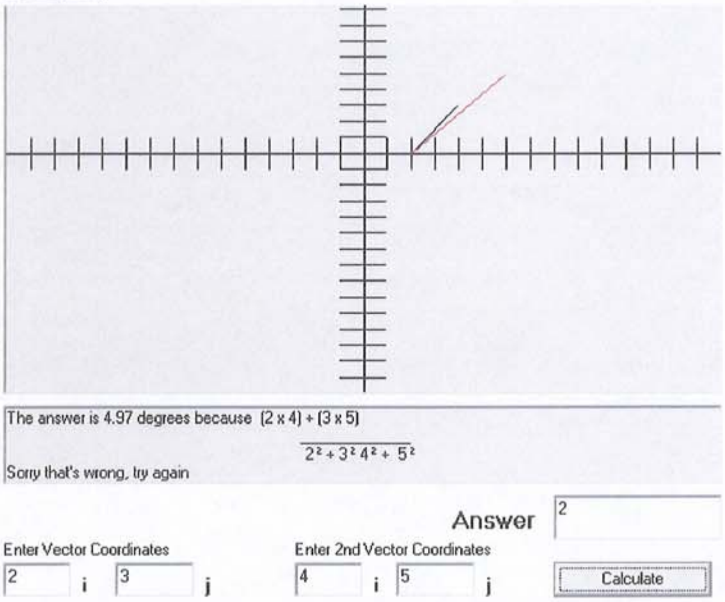
Test 57



```
(vectori(1) + vectori(1), vectorj(1))-(vectori(2) + vectori(2), vectorj(2)), vbBlue
```

Line Colour correctly set to blue

Test 64/65



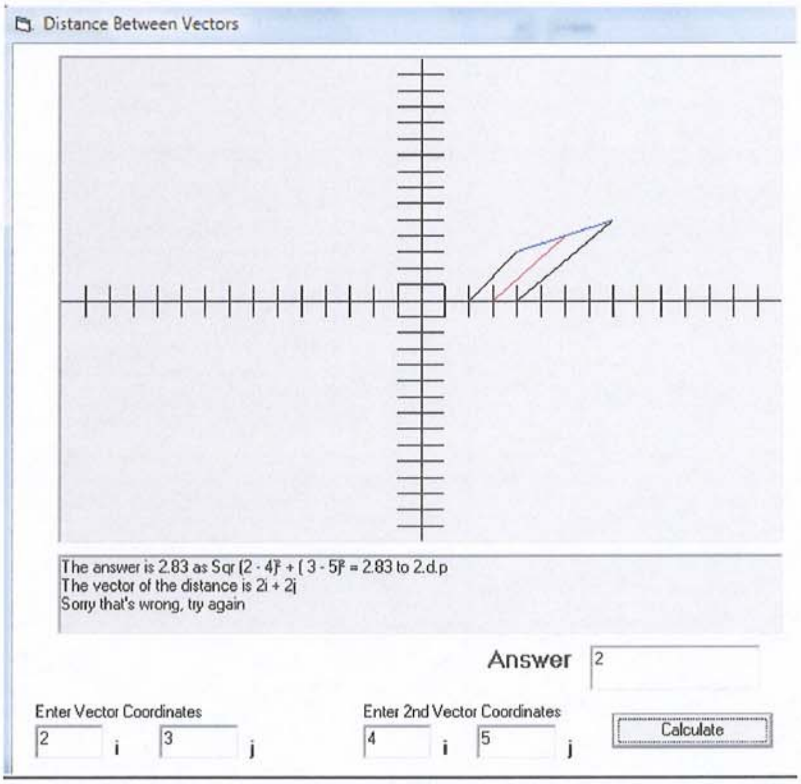
The answer is 4.97 degrees because (2 x 4) + (3 x 5)

$\overline{2^2 + 3^2 4^2 + 5^2}$

Sorry that's wrong, try again

Vectors & angle size is drawn correctly

Test 71



The answer is 2.83 as Sqr (2 - 4)^2 + (3 - 5)^2 = 2.83 to 2.d.p
The vector of the distance is 2i + 2j
Sorry that's wrong, try again

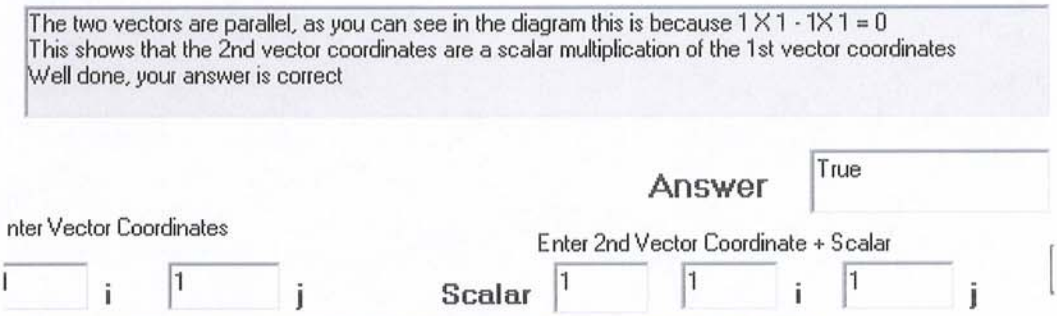
Answer 2

Enter Vector Coordinates 2 i 3 j

Enter 2nd Vector Coordinates 4 i 5 j

Explanation, answer and vector drawings are all shown together

Test 73



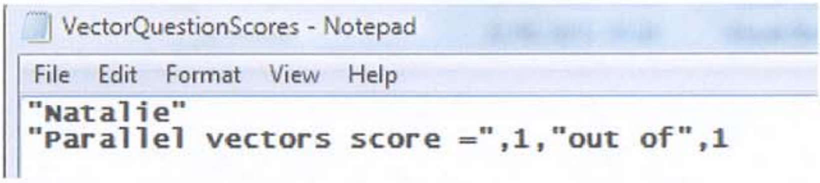
The two vectors are parallel, as you can see in the diagram this is because 1 X 1 - 1X 1 = 0
This shows that the 2nd vector coordinates are a scalar multiplication of the 1st vector coordinates
Well done, your answer is correct

Answer True

nter Vector Coordinates 1 i 1 j

Enter 2nd Vector Coordinate + Scalar

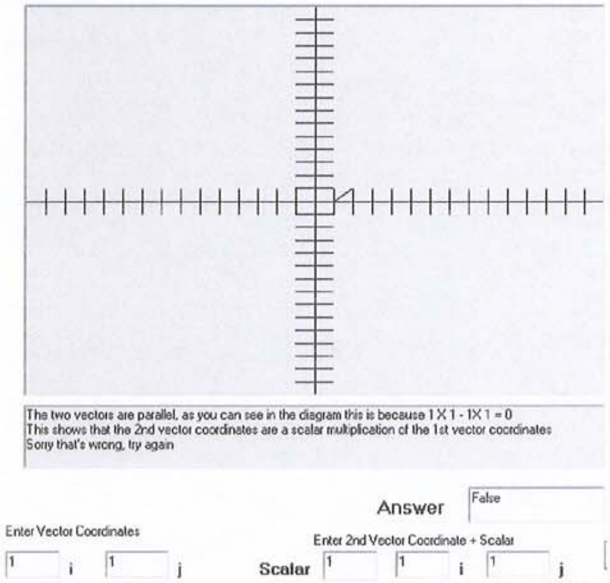
Scalar 1 1 i 1 j



```
"Natalie"
"Parallel vectors score =",1,"out of",1
```

Correctly marked and result written to file

Test 74



The two vectors are parallel, as you can see in the diagram this is because 1 X 1 - 1X 1 = 0
This shows that the 2nd vector coordinates are a scalar multiplication of the 1st vector coordinates
Sorry that's wrong, try again

Answer: False

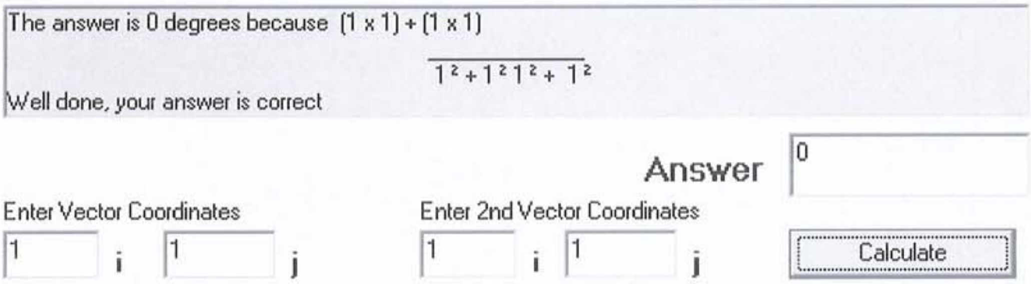
Enter Vector Coordinates: 1 i 1 j

Enter 2nd Vector Coordinate + Scalar: Scalar 1 | 1 i 1 j

```
"Bob"
"Parallel vectors score =",0,"out of",1
```

Correctly informed that the answer is wrong and score is not incremented.

Test 75



The answer is 0 degrees because (1 x 1) + (1 x 1)
1² + 1² 1² + 1²
Well done, your answer is correct

Answer: 0

Enter Vector Coordinates: 1 i 1 j

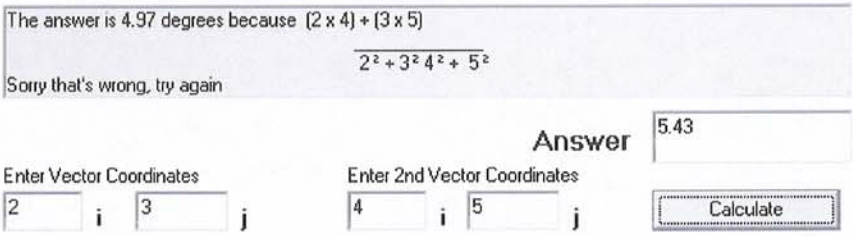
Enter 2nd Vector Coordinates: 1 i 1 j

```
If StuAns = Angle Then
correct = True
StuScore = StuScore + 1
```

```
"Fred"
"Angle vectors score =",1,"out of",2
```

Correction made to code to make sure that the StuAns is being compared to Angle

Test 76



The answer is 4.97 degrees because (2 x 4) + (3 x 5)
2² + 3² 4² + 5²
Sorry that's wrong, try again

Answer: 5.43

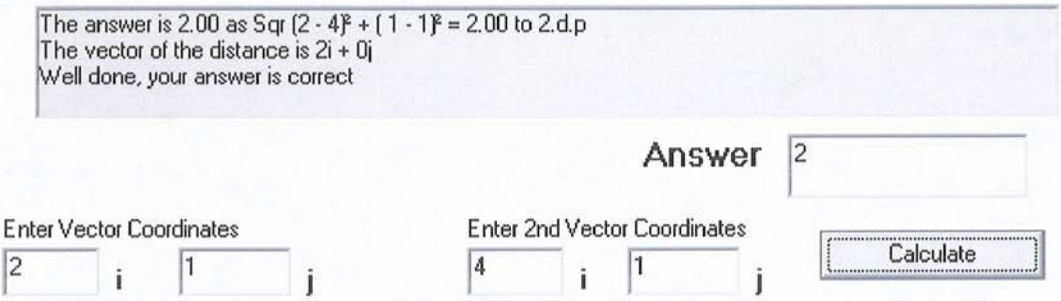
Enter Vector Coordinates: 2 i 3 j

Enter 2nd Vector Coordinates: 4 i 5 j

```
"Mark"
"Angle vectors score =",0,"out of",1
```

System recognises answer is wrong and score is not incremented in file.

Test 77

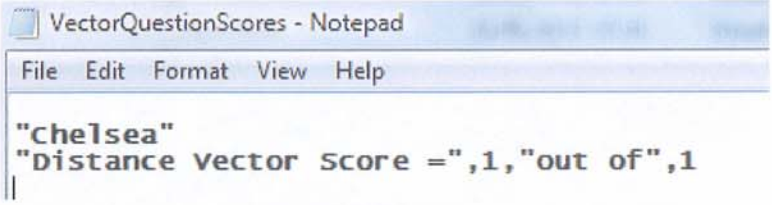


The answer is 2.00 as Sqr $(2 - 4)^2 + (1 - 1)^2$ = 2.00 to 2.d.p
The vector of the distance is 2i + 0j
Well done, your answer is correct

Answer 2

Enter Vector Coordinates 2 i 1 j

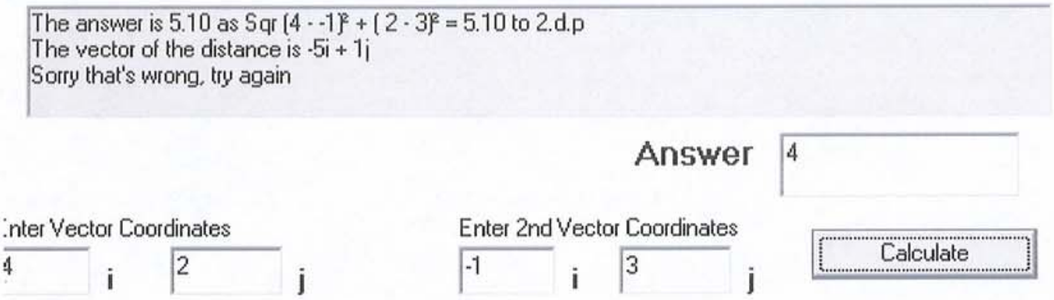
Enter 2nd Vector Coordinates 4 i 1 j



```
"Chelsea"
"Distance Vector Score =",1,"out of",1
```

System reacts correctly and as expected

Test 78

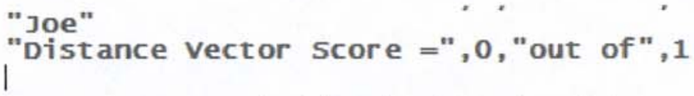


The answer is 5.10 as Sqr $(4 - -1)^2 + (2 - 3)^2$ = 5.10 to 2.d.p
The vector of the distance is -5i + 1j
Sorry that's wrong, try again

Answer 4

:nter Vector Coordinates 4 i 2 j

Enter 2nd Vector Coordinates -1 i 3 j



```
"Joe"
"Distance Vector Score =",0,"out of",1
```

System reacts correctly, informing the student their answer is wrong and

Validation

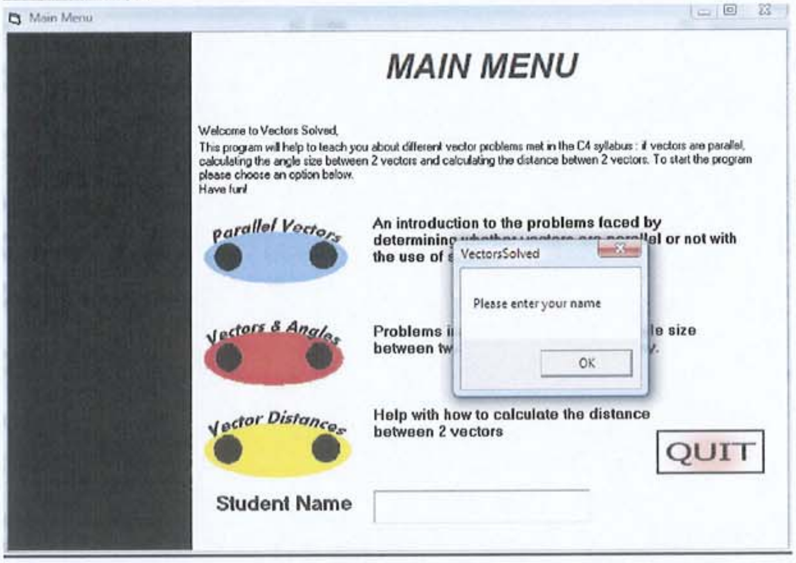
Validation testing must also be carried out to ensure that the data going into the program is both sensible and reasonable and that data must be entered into the system to avoid the system crashing. I will do this by testing the system with erroneous, boundary and valid data to see how if my system reacts correctly to each situation.

Validation Number	**Validation Data and Type of Check**	**Expected Result**	**Actual Result**	**Reference**
1	Student Name - " " (Presence Check)	A Message Box will show asking the user to input their name	The user is not allowed to move on until they have entered their	Screenshot

			name, prompted with message box.	
2	Student Name – “N ” (Format Check)	The Name will be accepted and written to file.	The name is accepted and written to file.	Screenshot
3	Vector Coordinate 1 (i) – “h” (Erroneous data) (Format Check)	Data is rejected.	Data is rejected and error message produced	Screenshot
4	Vector Coordinate 1 (i) – “15” (Boundary Data) (Range Check)	Data is accepted	Data is accepted and graph is drawn	Screenshot
5	Vector Coordinate 1 (i) – “-3” (Valid Data) (Range check)	Data is accepted	Data is accepted and graph is drawn	Screenshot
6	Vector Coordinate 1(j) – “20” (Erroneous data) (Range Check)	Data is rejected	Data is rejected and error message produced, asking user to input vector between the range	Screenshot
7	Vector Coordinate 1(j) – “-15” (Boundary Data) Range Check	Data is accepted	Data is accepted and graph is drawn	Screenshot
8	Vector Coordinate 1(j) – “ “ (Erroneous Data) Presence Check	Message Box appears asking the user to input a number	Message Box Appears	Screenshot
9	Vector Coordinate 2(i) –	Data is accepted	Data accepted	Screenshot

	“10” (Valid Data) Range Check			
10	Vector Coordinate 2(i) – “hjki” (Erroneous data) (Format Check)	Data is rejected and message box appears	Data rejected and according message/error box is shown	Screenshot
11	Vector Coordinate 2 (j) – “15” (Boundary Data) Range Check	Data is accepted	Data is accepted	Screenshot
12	Vector Coordinate 2 (j) – “002” (Valid Data) Range Check/Format Check	Data is accepted	Data is accepted	Screenshot
13	Scalar – “mk” (Erroneous data) Format Check	Data rejected and error box appears.	Data rejected and error box appears	Screenshot
14	Scalar – “-17” (Erroneous data) Range Check	Data rejected and error box appears	Data rejected and error box appears	Screenshot
15	Scalar – “3” (Valid Data) Range Check	Data accepted	Data accepted	Screenshot
16	Student Answer – “ “ (Presence Check)	Rejected and error message box appears asking the user to input an answer	As expected error box appears asking the student to input a value	Screenshot

Validation 1



MAIN MENU

Welcome to Vectors Solved,
This program will help to teach you about different vector problems met in the C4 syllabus : if vectors are parallel, calculating the angle size between 2 vectors and calculating the distance betwen 2 vectors. To start the program please choose an option below.
Have fun!

An introduction to the problems faced by determining [illegible] or not with the use of [illegible]

VectorsSolved

Please enter your name

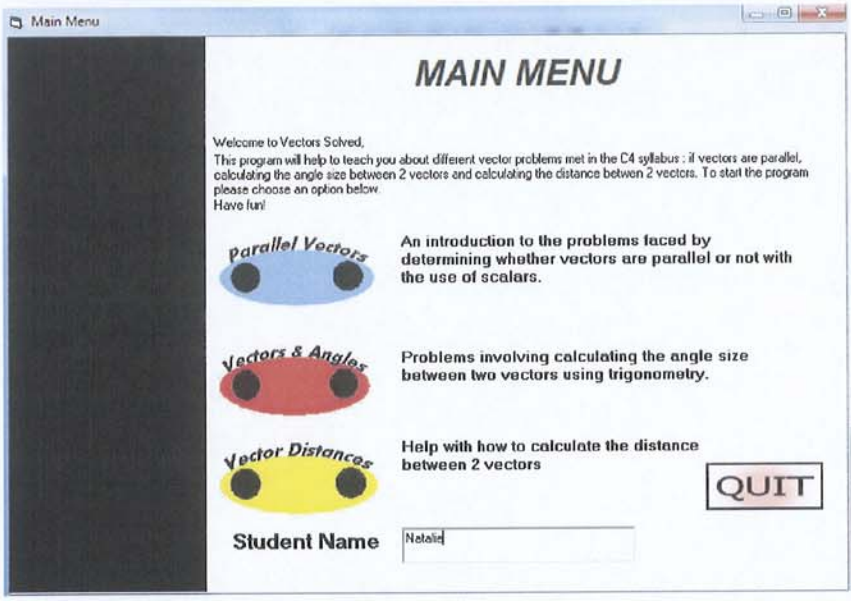
Problems i [illegible] le size between tw [illegible]

Help with how to calculate the distance between 2 vectors

Student Name

Student cannot move on until they have entered their name.

Validation 2



MAIN MENU

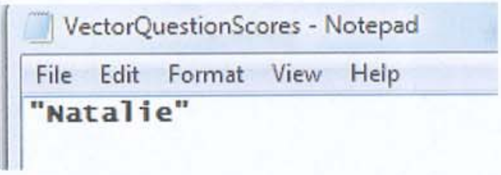
Welcome to Vectors Solved,
This program will help to teach you about different vector problems met in the C4 syllabus : if vectors are parallel, calculating the angle size between 2 vectors and calculating the distance betwen 2 vectors. To start the program please choose an option below.
Have fun!

An introduction to the problems faced by determining whether vectors are parallel or not with the use of scalars.

Problems involving calculating the angle size between two vectors using trigonometry.

Help with how to calculate the distance between 2 vectors

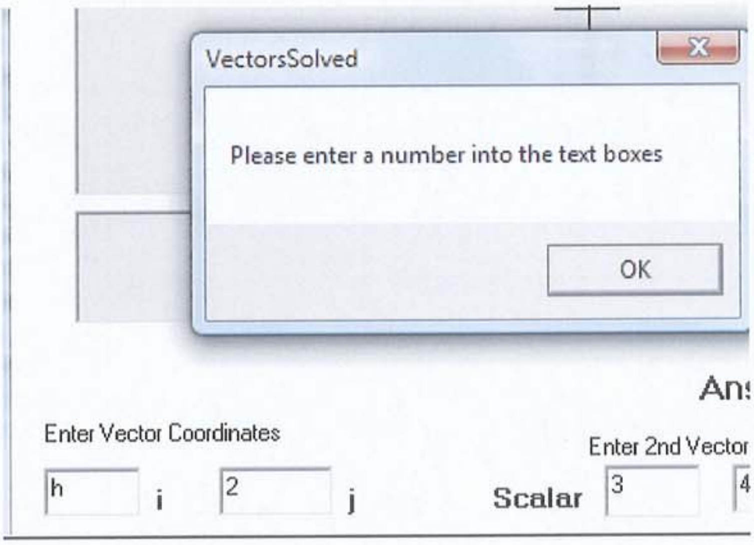
Student Name: Natalie



```
"Natalie"
```

Name is accepted and written to file.

Validation 3


VectorsSolved

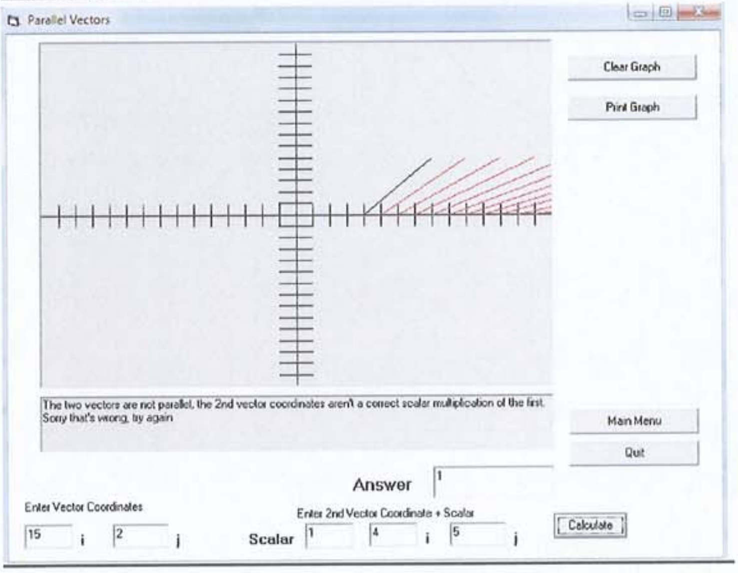
Please enter a number into the text boxes

Enter Vector Coordinates: h i 2 j

Enter 2nd Vector: 3 4

Letter 'h' is rejected

Validation 4


Parallel Vectors

The two vectors are not parallel, the 2nd vector coordinates aren't a correct scalar multiplication of the first.
Sorry that's wrong, try again

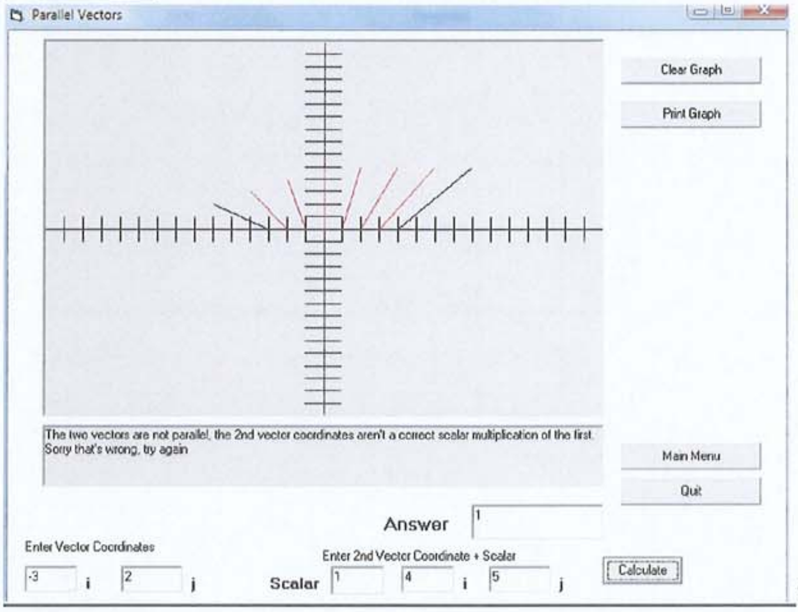
Answer: 1

Enter Vector Coordinates: 15 i 2 j

Enter 2nd Vector Coordinate + Scalar: Scalar 1, 4 i 5 j

'15' is accepted and graph is drawn

Validation 5



The two vectors are not parallel, the 2nd vector coordinates aren't a correct scalar multiplication of the first.
Sorry that's wrong, try again

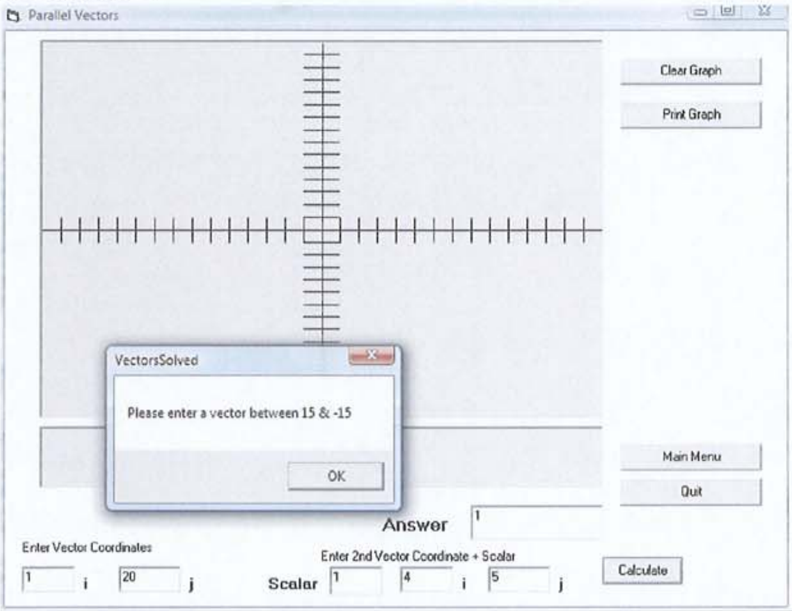
Answer 1

Enter Vector Coordinates
-3 i 2 j

Enter 2nd Vector Coordinate + Scalar
Scalar 1 4 i 5 j

"-3" is accepted and graph is drawn

Validation 6



Please enter a vector between 15 & -15

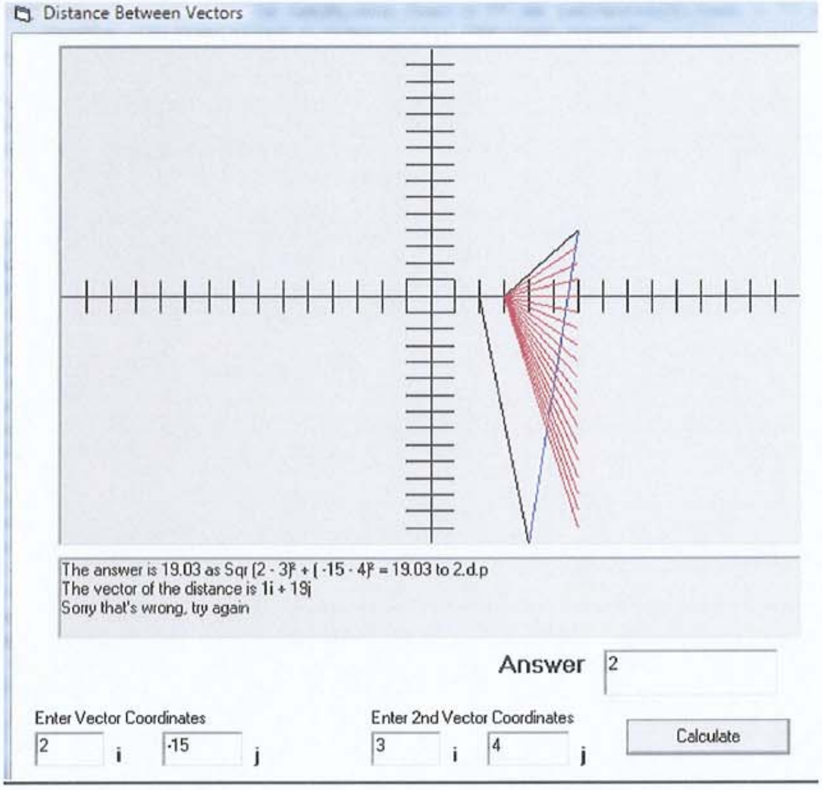
Answer 1

Enter Vector Coordinates
1 i 20 j

Enter 2nd Vector Coordinate + Scalar
Scalar 1 4 i 5 j

"20" is rejected and error message box appears

Validation 7



```
The answer is 19.03 as Sqr (2 - 3)² + ( -15 - 4)² = 19.03 to 2.d.p
The vector of the distance is 1i + 19j
Sorry that's wrong, try again
```

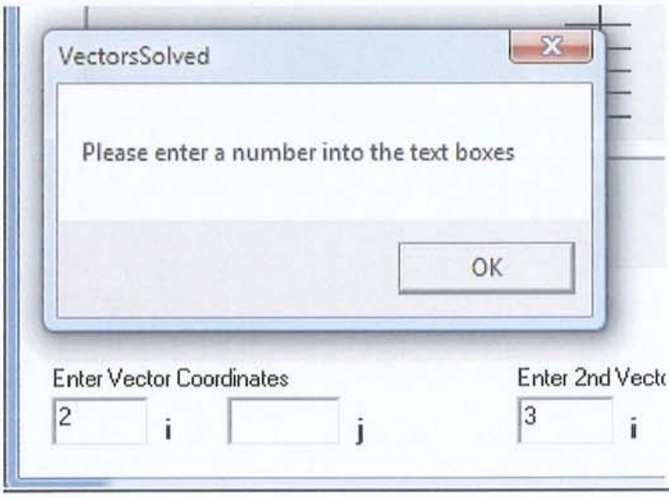
Answer: 2

Enter Vector Coordinates: 2 i -15 j

Enter 2nd Vector Coordinates: 3 i 4 j

"-15" is accepted and graph drawn

Validation 8



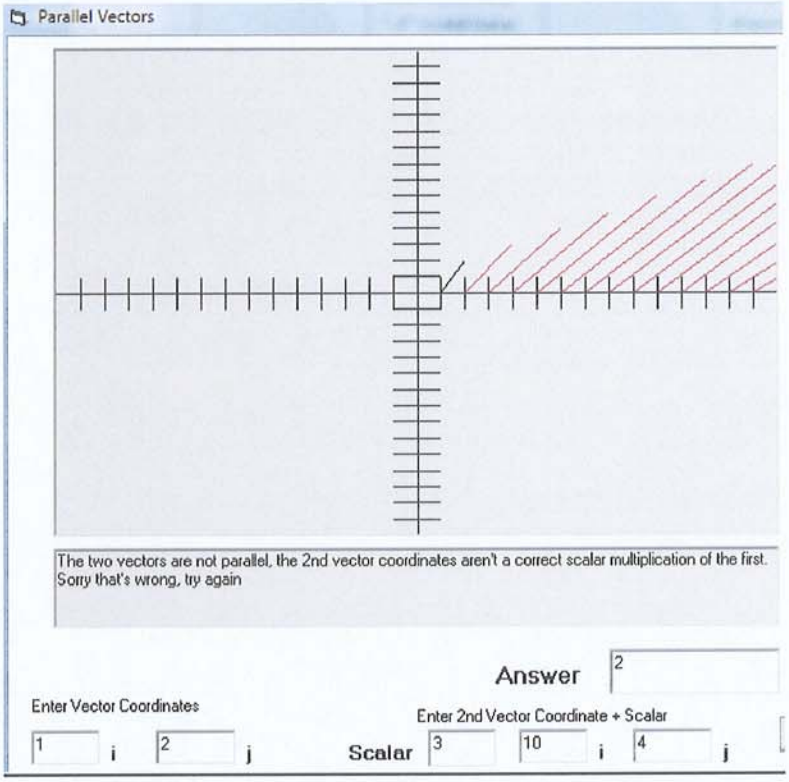
Please enter a number into the text boxes

Enter Vector Coordinates: 2 i j

Enter 2nd Vect: 3 i

Message box correctly shows up

Validation 9



The two vectors are not parallel, the 2nd vector coordinates aren't a correct scalar multiplication of the first.
Sorry that's wrong, try again

Answer 2

Enter Vector Coordinates

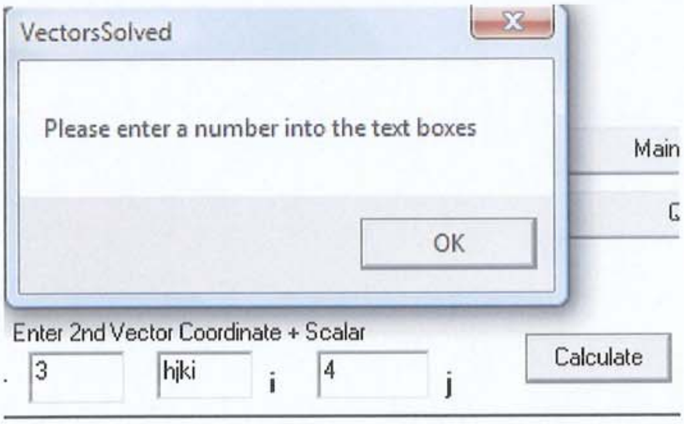
1 i 2 j

Enter 2nd Vector Coordinate + Scalar

Scalar 3 10 i 4 j

'10' is accepted and graph is drawn

Validation 10


VectorsSolved

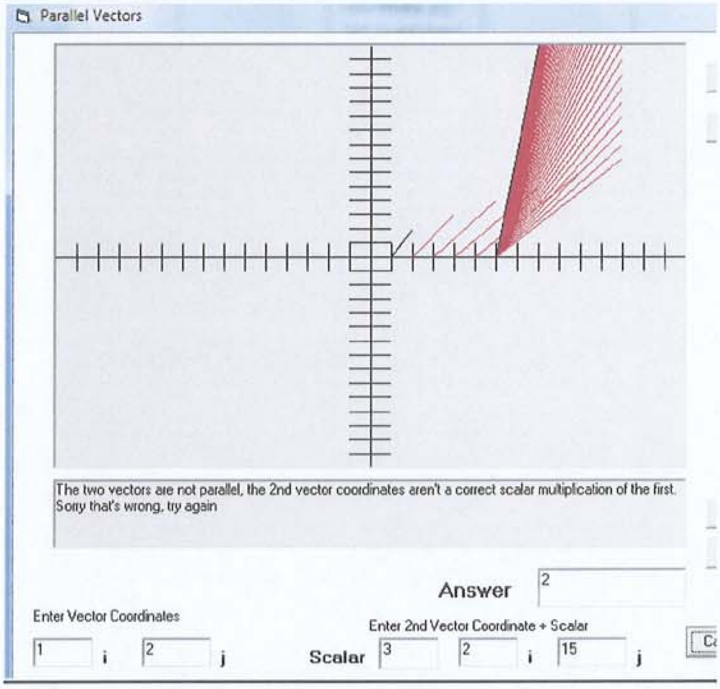
Please enter a number into the text boxes

Enter 2nd Vector Coordinate + Scalar

3 hjki i 4 j

Data is rejected and correct message/error box appears

Validation 11



The two vectors are not parallel, the 2nd vector coordinates aren't a correct scalar multiplication of the first.
Sorry that's wrong, try again

Answer 2

Enter Vector Coordinates

1 i 2 j

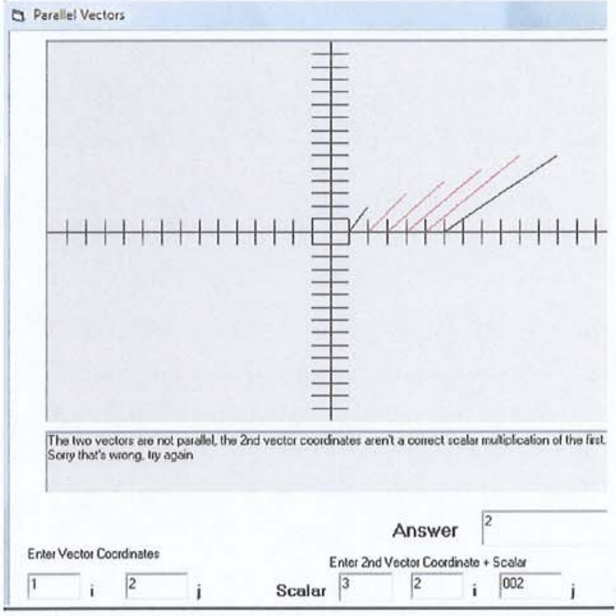
Scalar 3

Enter 2nd Vector Coordinate + Scalar

2 i 15 j

'15' is accepted and graph is drawn

Validation 12



The two vectors are not parallel, the 2nd vector coordinates aren't a correct scalar multiplication of the first.
Sorry that's wrong, try again

Answer 2

Enter Vector Coordinates

1 i 2 j

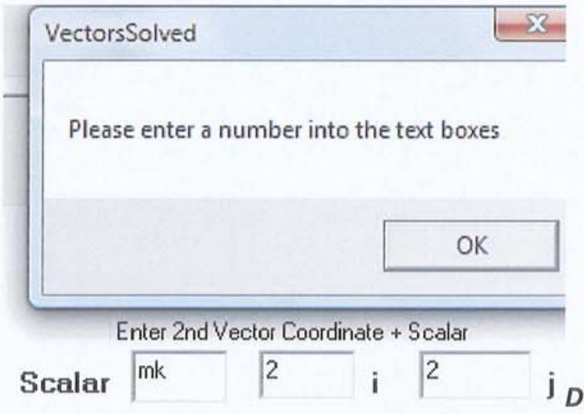
Scalar 3

Enter 2nd Vector Coordinate + Scalar

2 i 002 j

'002' is accepted and graph drawn

Validation 13



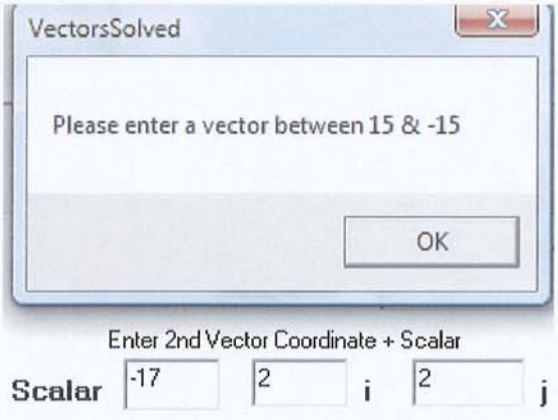
Please enter a number into the text boxes

Enter 2nd Vector Coordinate + Scalar

Scalar mk 2 i 2 j

Data rejected and error box appears

Validation 14



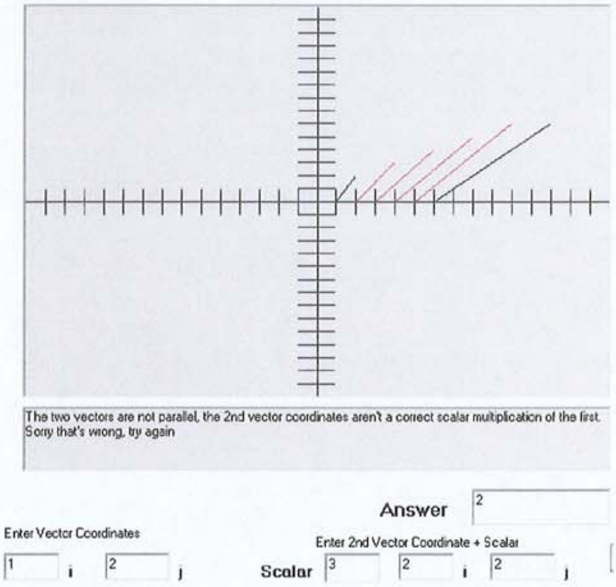
Please enter a vector between 15 & -15

Enter 2nd Vector Coordinate + Scalar

Scalar -17 2 i 2 j

Scalar correctly rejected

Validation 15



The two vectors are not parallel, the 2nd vector coordinates aren't a correct scalar multiplication of the first.
Sorry that's wrong, try again

Answer 2

Enter Vector Coordinates

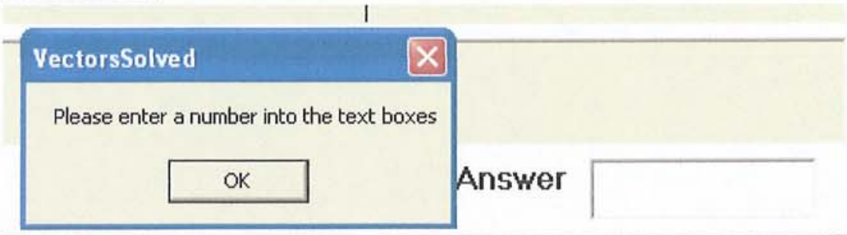
1 i 2 j

Enter 2nd Vector Coordinate + Scalar

Scalar 3 2 i 2 j

Scalar '3' accepted

Validation 16



VectorsSolved

Please enter a number into the text boxes

User is correctly prompted to enter an answer

Above I have described the Alpha Testing I have undergone; however to fully test my program and make sure that it meets the user's expectations and requirements I must also undergo Beta Testing, this involves releasing the program to the user and allowing them to use it and comment on it. Below I have included a sample questionnaire that the user has filled in about the system I have created.

1) How easily did you manage to navigate round the system?

Very easy to navigate round. Lots of buttons which make it easy

2) Did you encounter any errors or difficulties whilst using the program?

No.

3) In terms of the colour schemes/layout of the system was it visually appealing?

Yes, layout is very clear. Main menu colour scheme is nice.

4) Do you feel that the system would help to teach students about vector problems efficiently?

Yes.

5) Were all the major sections covered within the system that you required?

Yes, I find these sections the most difficult.

6) Are you happy with the printed output of the system that can be used as a hard copy for students? Yes, very helpful and in a clear layout

7) Is there anything else you would have like to have seen featured within the program?

No.

8) Any additional comments

NO.

Whilst the students were using the program it was important to use this as testing to see whether they had any problems with the system and whether all their data etc was handled in the correct way, I asked them to carry out the following tasks and for them to write the results in the below space :

Task

Print a copy of your vector graph using the print command button located on any of the problem pages.

Outcome

Printed in a clear layout.

Task

Enter a vector coordinate above the given range (15).

Outcome

Error message displayed.

Task

Don't enter your name into the Student Name text box located on the Main Menu.

Outcome

Error message displayed telling me to enter my name

Task

Enter a letter into one of the vector coordinate text boxes you fill in.

Outcome

Error message displayed telling me to enter a number.

Task

Enter a valid vector coordinate into one of the vector coordinate text boxes you fill in.

Outcome

Vector accepted and graph was drawn

4.4.11 CE

In addition to the regular systems testing that I have undergone, Mr M also wanted me to produce information on how his students are performing on the system I have created; there are 5 pupils in his maths class and each has had time to use & interact with the program.

Each student has produced print outs on some of the problems they have attempted and their results have been managed in the text file that has been written whilst they have been interacting with and using the system. The hard copy of this file is then printed out and has been given for Mr M to refer to in order to see how his students are performing. The results are shown on the following pages where the students have written the outcome of the tasks they have performed.

Student Scores

Test 79 - Formatted

The results of how the students have scored on the various topics are as follows:

Fred

Angle vectors score = 0 out of 2

Victor

Distance Vector Score = 0 out of 2

Test 79 - Before

Student Scores

The results of how the students have scored on the various topics are as follows:

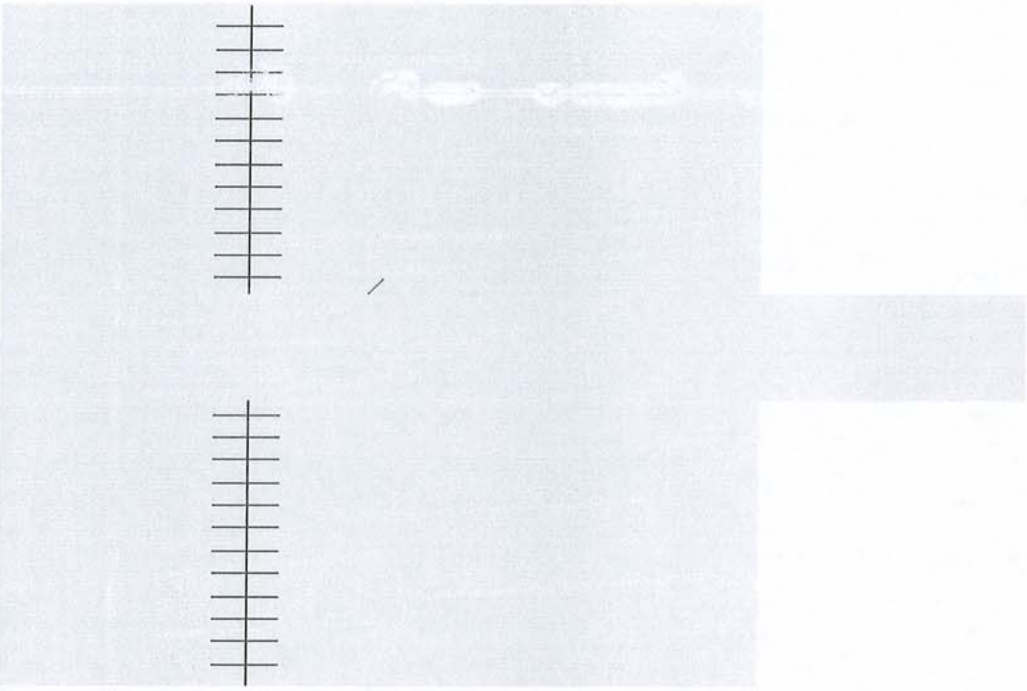
Fred

Angle vectors score = 0 out of 2

Victor

Distance Vector Score = 0 out of 2

Test 38



Test 38

Angle Vector Problem

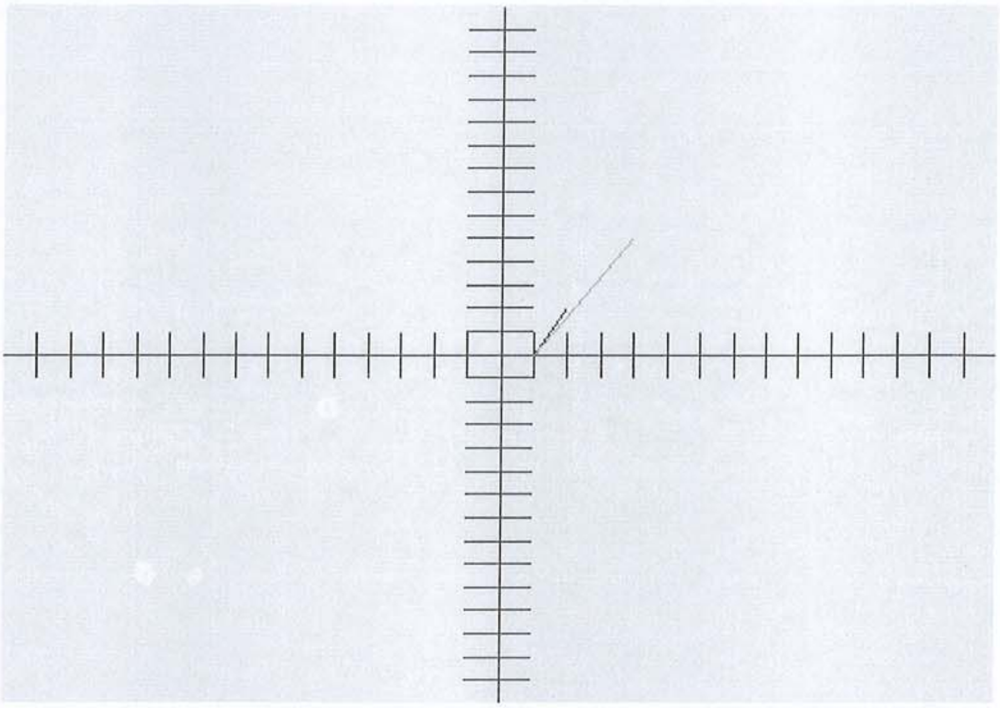
Student Name: **M**

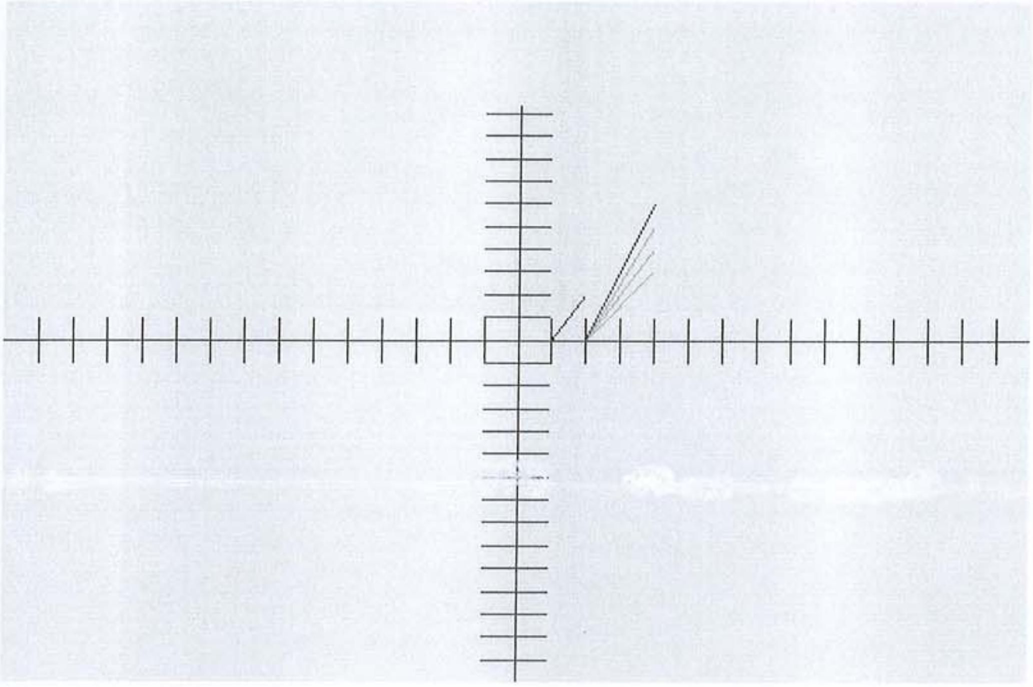
The vector graph of 1 i + 2 j and 3 i + 5 j

The answer is 4.4 degrees because

$$\frac{(1 \times 3) + (2 \times 5)}{1^2 + 2^2 3^2 + 5^2}$$

Unfortunately your answer was wrong, either try again or make sure to contact your teacher with the relevant questions.





Test 55

Maintenance

System Maintenance

System Overview

The system I have created will run as a compiled executable file on almost any platform complete with a keyboard and a mouse for interaction with the system. It is opened by simply double clicking on the executable file and waiting for the system to load.
The system I have created contains 4 main pages, not including the splash screen that loads as an introduction to the program; these can be seen in the modular system diagrams I have created, located in the Design section. Each separate vector problem page teaches the students about different vector questions by using graphs that are drawn when the students enter the vector coordinates they wish to compare. The student also has the opportunity to test themselves by entering what they think the answer will be; these results and their name are then written to file for Mr M to assess his students understanding.

To see the full system design and the structure of the system see page14-18 in Design.

The first main page that the user will encounter is the Main Menu page that provides the links to each separate vector problem page and allows the user to enter their name into the system which is written to file along with their results. Three separate pages are available to be navigated from the Main Menu; the Parallel Vectors page, the Angle Vectors page and the Distance Vectors page, all of these pages incorporate the same general idea of helping to aid the student's learning of C4 vectors whilst testing them; however each one provides slightly different problems.

Difficult Aspects of the System

These are parts of the system that can pose a slight difficult due to the complexity of the system, they have been explained in detail below as a useful walkthrough.

To begin the user enters their vector 1 (i & j) coordinates and vector 2 (i & j) coordinates and the scalar (on the Distance Vectors page), and what they think the correct answer will be, they will then press the Calculate button, this starts the drawing of the vectors onto the graph for the student to see; the first & last vectors are drawn in a black line and animation from Vector 1 to Vector 2 using red lines is used to show the step by step result. This process is slightly different on each separate form; on the Parallel Vectors page simply only the animation from one vector to the other is shown; on the Angle Vectors page just the 2 vectors are drawn but to the exact angle that has been calculated and finally on the Distance Vectors page the animation between the 2 vectors is drawn but also a blue line between the starting and ending vector to show the distance between them.

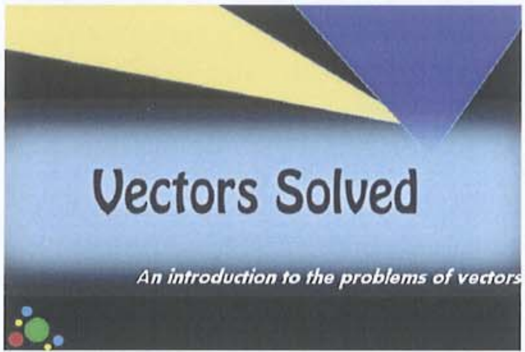
Whilst this is under way below the drawn graph an explanation appears to explain to the student why the answer is what it is; providing not only the correct formulae but also whether the student has the answer right/wrong. The result of how many answers the student has answered correctly and the amount of questions they have answered is then written to a text file stored in the same location as the program along with the name they entered at the beginning of the program.

Integer values can only be entered into the vector coordinates, scalar and answer box (with the exception of the parallel vectors page where the answer must be of Boolean value, True/False) and coordinates and scalar text boxes by the user; this makes sure that the vector can be drawn appropriately on the scale of the graph; which is set within the program at run time.

If the student wishes to do another problem this is easily manageable using the Clear Graph command button, this button erases the vectors that have been drawn along with the explanation, formulae etc and provides the student the opportunity to either re draw the same vector or enter new coordinates.

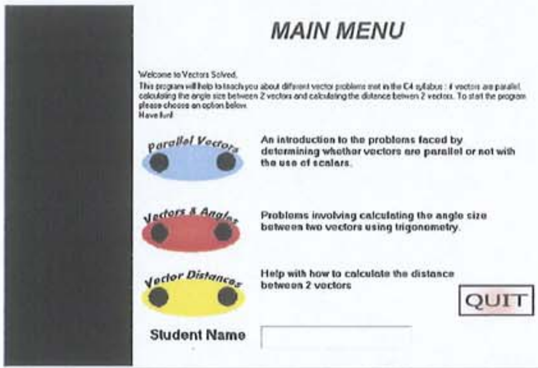
In addition to this there is also a command button on each problem form that allows the user to print off the problem they have completed, this includes the graph, the explanation and their name; this provides the student with a hard copy if they wish to refer back to it.
Of course on each form there is also a navigational button that links back to the Main Menu to access the other problem pages and a Quit button to end the program at any time.

frmSplash.frm



Screen that loads as soon as the user begins the program, no user interaction with the form. Loads to Main Menu .

frmMain.frm


MAIN MENU

Welcome to Vectors Solved.
This program will help to teach you about different vector problems met in the C4 syllabus : if vectors are parallel, calculating the angle size between 2 vectors and calculating the distance between 2 vectors. To start the program please choose an option below.
Have fun!

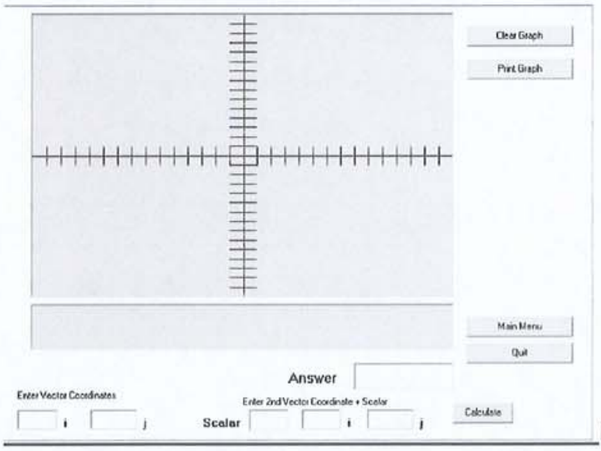
Parallel Vectors — An introduction to the problems faced by determining whether vectors are parallel or not with the use of scalars.

Vectors & Angles — Problems involving calculating the angle size between two vectors using trigonometry.

Vector Distances — Help with how to calculate the distance between 2 vectors

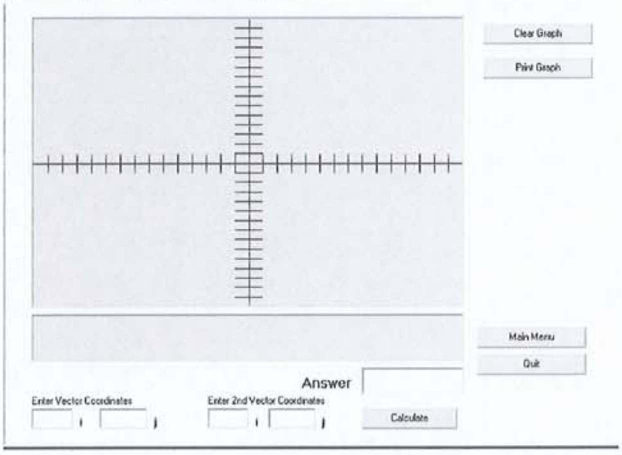
Loaded 5 seconds after splash screen, provides introduction to the program and links to each of the separate problem page. Where the user enters his/her name into the system and chooses which vector problems they want to learn about. Also allows the teacher to print off the students' results.

frmParallelVectors.frm



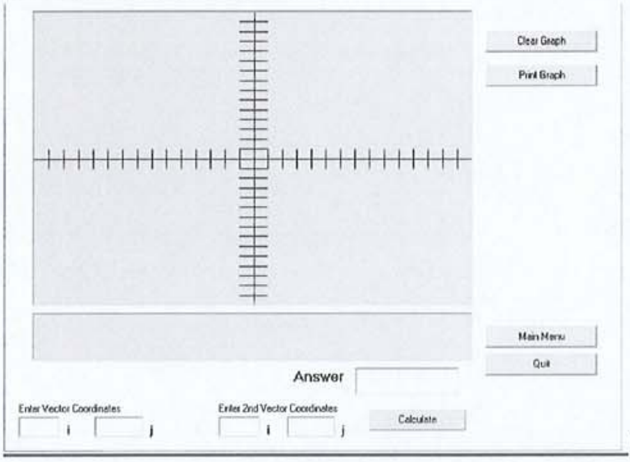
Linked to from frmMain.frm, teaches the user about whether vectors are parallel or not by entering different vector coordinates, scalar and an answer. Allows the user to print their graph, clear the graph, navigate back to the Main Menu or quit the program.

frmAngleVectors.frm



Linked to from frmMain.frm, virtually identical in interface to the frmParallelVectors.frm; however the coding/function of the vectors is different. Allows the user to print their graph, clear the graph, navigate back to the Main Menu or quit the program.

frmDistanceVectors.frm



Linked to from frmMain.frm, again the interface is almost identical to frmAngleVectors.frm and frmParallelVectors.frm but once again the coding/function of the vectors is different. Allows the user to print their graph, clear the graph, navigate back to the Main Menu or quit the program.

Procedure List

Item	Name	Description	Parameters
Procedure	Public Sub WriteName()	Writes the user's name to file.	No parameters passsed
Procedure	Public Sub DrawVectors(pctDraw1)	Draws the first vector the user has input	Passes the picture box into the procedure
Procedure	Public Sub VectorAnim(vectori(),vectorj())	Animates the vector 1 to vector 2	Passes the vector i coordinates (1&2) and the vector j coordinates (1 & 2)
Function	Public Function GetParallel(vectori() , vectorj()) As Boolean	Calculates if the 2 vectors are parallel and passes the result back.	Passes the vector coordinates into the function and passes the result as a Boolean back.
Function	Public Function GetAngle(Answer ,Pi) As Double	Calculates the angle between 2 vectors by using the answer obtained and Pi.	Passes the answer and the constant pi into the function and passes the angle back as a double.
Function	Public Function GetAnswer(vectori() , vectorj()) As Double	Calculates the answer to be used in the GetAngle Function.	Passes the vector coordinates into the function and passes the answer back as a double.
Procedure	Public Sub DrawAxis(pctDraw1, xmax, xmin, ymax, ymin)	Draws the axis within the picture box for the vectors to be plotted on.	Passes in the picture box and the constants for the scale of the graph.

Function	Public Function GetDistance(vectori() , vectorj()) as Double	Calculates the distance between 2 vectors.	Passes the vector coordinates into the function and the answer back as a double.
Procedure (If Clear Graph command button clicked)		Clears the graph and the explanation picture boxes.	No parameters passed
Procedure (If Print Graph command button clicked)		Prints the graph and the explanation box	No parameters passed.

NOTE

I have used various procedures and functions within my program to make sure that my system works correctly and carries out tasks efficiently and with a modular fashion using efficient coding, below I will now show the differences between the pseudo code that I used in my design stage and the actual coding I have used within my program.

Procedure WriteName

Procedure WriteName(StuScore,QuestionsAns) Name as String Filename = VectorQuestionsScore.text Name = frmMain.txtName Open Filename For Append Write, Name, StuScore, QuestionsAns Close File End Sub	Public Sub WriteName() Filename = App.Path & "\VectorQuestionScores.txt" Name = frmMain.txtName Open Filename For Append As #1 Print #1, Name Close #1 End Sub

Procedure DrawAxis

Procedure DrawAxis (xmin, xmax,ymin,ymax)	Public Sub DrawAxis(pctDraw1, xmax, xmin,

DrawScale = (xmin,xmax) – (ymin,ymax) DrawLine (xmin, 0)-(xmax, 0) For counter = (xmin+1) To (xmax-1) DrawLine (counter, -1)-(counter, 1) Next counter Draw Line (0, ymin)-(0, ymax For counter = (ymin+1) To (ymax-1) DrawLine (-1, counter)-(1, counter Next counter ForeColour=black End Procedure	ymax, ymin) Dim counter As Integer Dim x As Integer Dim y As Integer pctDraw1.Scale (xmin, xmax)-(ymax, ymin) pctDraw1.Line (xmin, 0)-(xmax, 0) For counter = (xmin + 1) To (xmax - 1) pctDraw1.Line (counter, -1)-(counter, 1) Next counter pctDraw1.Line (0, ymin)-(0, ymax) For counter = (ymin + 1) To (ymax - 1) pctDraw1.Line (-1, counter)-(1, counter) Next counter pctDraw1.ForeColor = vbBlack End Sub

Procedure Draw Vectors

Procedure DrawVectors (vectori(), vectorj()) Draw Line (vectori(1), 0)-((vectori(1) + vectori(1)), vectorj(1)), Black Call Vector Anim(pass variables) End Procedure	Public Sub DrawVectors(pctDraw1) pctDraw1.Line (vectori(1), 0)-((vectori(1) + vectori(1)), vectorj(1)), vbBlack Call VectorAnim(vectori(), vectorj()) End Sub

Procedure Vector Anim

Procedure VectorAnim (vectori(),vector()) If vectori(1) < vectori(2) And vectorj(1) < vectorj(2) Then vectori(1) = vectori(1) + 1 vectorj(1) = vectorj(1) + 1 Start Timer Call VectorAnim(vectori(),vectorj()) ElseIf vectori(1) = vectori(2) And vectorj(1) < vectorj(2) Then vectorj(1) = vectorj(1) + 1 Start Timer Call VectorAnim(vectori(),vector()) ElseIf vectori(1) < vectori(2) And vectorj(1) = vectorj(2) Then vectori(1) = vectori(1) + 1	Public Sub VectorAnim(ByRef vectori() As Integer, ByRef vectorj() As Integer) If vectori(1) < vectori(2) And vectorj(1) < vectorj(2) Then vectori(1) = vectori(1) + 1 vectorj(1) = vectorj(1) + 1 Timer1.Enabled = True pctDraw1.Line (vectori(1), 0)-(vectori(1) + vectori(1), vectorj(1)), vbRed Call VectorAnim(vectori(), vectorj()) ElseIf vectori(1) = vectori(2) And vectorj(1) <

Start Timer Call VectorAnim(vectori(),vectorj()) Stop Timer End If End Procedure	vectorj(2) Then vectorj(1) = vectorj(1) + 1 Timer1.Enabled = True pctDraw1.Line (vectori(1), 0)-(vectori(1) + vectori(1), vectorj(1)), vbRed Call VectorAnim(vectori(), vectorj()) ElseIf vectori(1) < vectori(2) And vectorj(1) = vectorj(2) Then vectori(1) = vectori(1) + 1 Timer1.Enabled = True pctDraw1.Line (vectori(1), 0)-(vectori(1) + vectori(1), vectorj(1)), vbRed Call VectorAnim(vectori(), vectorj()) ElseIf vectori(1) > vectori(2) And vectorj(1) > vectorj(2) Then vectori(1) = vectori(1) - 1 vectorj(1) = vectorj(1) - 1 Timer1.Enabled = True pctDraw1.Line (vectori(1), 0)-(vectori(1) + vectori(1), vectorj(1)), vbRed Call VectorAnim(vectori(), vectorj()) ElseIf vectori(1) = vectori(2) And vectorj(1) > vectorj(2) Then vectorj(1) = vectorj(1) - 1 Call VectorAnim(vectori(), vectorj()) Timer1.Enabled = True pctDraw1.Line (vectori(1), 0)-(vectori(1) + vectori(1), vectorj(1)), vbRed ElseIf vectori(1) > vectori(2) And vectorj(1) = vectorj(2) Then

	vectori(1) = vectori(1) - 1 Timer1.Enabled = True pctDraw1.Line (vectori(1), 0)-(vectori(1) + vectori(1), vectorj(1)), vbRed Call VectorAnim(vectori(), vectorj()) ElseIf vectori(1) > vectori(2) And vectorj(1) < vectorj(2) Then vectori(1) = vectori(1) - 1 vectorj(1) = vectorj(1) + 1 Timer1.Enabled = True pctDraw1.Line (vectori(1), 0)-(vectori(1) + vectori(1), vectorj(1)), vbRed Call VectorAnim(vectori(), vectorj()) ElseIf vectori(1) < vectori(2) And vectorj(1) > vectorj(2) Then vectori(1) = vectori(1) + 1 vectorj(1) = vectorj(1) + 1 Timer1.Enabled = True pctDraw1.Line (vectori(1), 0)-(vectori(1) + vectori(1), vectorj(1)), vbRed Call VectorAnim(vectori(), vectorj()) End If **End Sub**

Function Get Angle

Function GetAngle (answer) GetAngle = InverseTan(-answer/SquareRoot(-answer*answer+1))+2*InverseTan(1))*(180/Pi)	Public Function GetAngle(ByRef Answer As Double, ByRef Pi As Double) As Double

Format GetAngle “#0.00” End Function	On Error Resume Next If Answer = 1 Then GetAngle = 0 Exit Function End If GetAngle = (Atn(-Answer / Sqr(-Answer * Answer + 1)) + 2 * Atn(1)) * (180 / Pi) On Error GoTo 0 GetAngle = Format(GetAngle, "#0.00") End Function

Function Get Answer

Function GetAnswer (vectori(),vectorj())
answer1 = ((vectori(1) * vectori(2)) + (vectorj(1) * vectorj(2)))
answer2 = (SquareRoot(vectori(1) ^ 2 + vectorj(1) ^ 2)) * (SquareRoot(vectori(2) ^ 2 + vectorj(2) ^ 2))
GetAnswer = answer1 / answer2
End Function
	Public Function GetAnswer(ByRef vectori() As Integer, ByRef vectorj() As Integer) As Double
	Dim answer1 As Double
	Dim answer2 As Double
	answer1 = ((vectori(1) * vectori(2)) + (vectorj(1) * vectorj(2)))
	answer2 = (Sqr(vectori(1) ^ 2 + vectorj(1) ^ 2)) * (Sqr(vectori(2) ^ 2 + vectorj(2) ^ 2))
	GetAnswer = (answer1 / answer2)
	End Function

Function Get Parallel

Function GetParallel (vectori(),vectorj()) If (vectori(1) * vectorj(2)) - (vectorj(1) * vectori(2)) = 0 Then GetParallel = True Else	Public Function GetParallel(ByRef vectori() As Integer, ByRef vectorj() As Integer) As Boolean If (vectori(1) * vectorj(2)) - (vectorj(1) *

GetParallel = False End If End Function	vectori(2)) = 0 Then GetParallel = True Else GetParallel = False End If End Function

Function Get Distance

Function GetDistance (vectori(),vectorj()) answer = Square Root(((vectori(1) - vectori(2)) ^ 2) + ((vectorj(1) - vectorj(2)) ^ 2)) End Function	Public Function GetDistance(ByRef vectori() As Integer, ByRef vectorj() As Integer) GetDistance = Sqr(((vectori(1) - vectori(2)) ^ 2) + ((vectorj(1) - vectorj(2)) ^ 2)) End Function

Procedure Clear Graph (If Clear Graph command button clicked)

Command Clear Click Clear Picture Box Clear Explanation Box Call DrawAxis End	Private Sub cmdClear_Click() pctDraw1.Cls pctExplanation.Cls Call frmDistanceVectors.DrawAxis(pctDraw1, xmax, xmin, ymax, ymin) End Sub

Procedure Print Graph (If Print Graph command button clicked on problem pages)

Procedure PrintGraph Initialise Printer Font=Bold FontSize = 18 Print Vector Problem	Private Sub cmdPrint_Click() iniprinter Printer.FontBold = True Printer.FontSize = 18 Printer.Print Spc(5); "Angle Vector Problem"

FontSize = 12 Print StudentName Font = Normal Print Details of Problem Print StudentAnswer Correct Print VectorGraph (using function from API) End Procedure	Printer.Print Printer.FontSize = 12 Printer.Print "Student Name:"; Spc(3); StudentName Printer.Print Printer.FontBold = False Printer.Print "The vector graph of "; vectori(1); "i + "; vectorj(1); "j and "; vectori(3); "i + "; vectorj(3); "j" Printer.Print Printer.Print Printer.Print "The answer is " & Angle & " degrees because "; "" Printer.Print Tab(43); "(" & vectori(1) & " x " & vectori(3) & ") + (" & vectorj(1) & " x " & vectorj(3); ")"; vbCrLf; Tab(43); "______________"; vbCrLf; Tab(43); vectori(1); "² + " & vectorj(1); " ² " & vectori(2); " ² + "; vectorj(2); "² " Printer.Print If correct = True Then Printer.Print "Well done you correctly answered this question" Else Printer.Print "Unfortunately your answer was wrong, either try again or make sure to contact your teacher " Printer.Print "with the relevant questions." End If AddPictureBox pctDraw1, Printer.ScaleWidth / 2 - pctDraw1.Width / 2, Printer.ScaleHeight / 2 - pctDraw1.Height / 2 Printer.EndDoc Call frmDistanceVectors.DrawAxis(pctDraw1, xmax, xmin, ymax, ymin) End Sub

Procedure ReadStudentResults (If Print Student Results command button is clicked on Main Menu)

Private Sub PrintResults Initialise Printer	Private Sub cmdPrintResults_Click() iniprinter

Tmp as String Filename = VectorQuestionScores.txt Font=Bold FontSize=18 Print "Student Scores" Font=Normal FontSize=10 Print"The results of how the students have scored on the various topics are as follows:" Font = Italic Open Filename For Reading While EOF(1) = 0 Input Line, tmp Print tmp End Loop Close File	Dim tmp As String Filename = App.Path & "\VectorQuestionScores.txt" Printer.FontBold = True Printer.FontSize = 18 Printer.Print Spc(15); "Student Scores" Printer.Print Printer.Print Printer.FontBold = False Printer.FontSize = 10 Printer.Print ; Spc(15); "The results of how the students have scored on the various topics are as follows:" Printer.Print Printer.Print Printer.FontItalic = True Open Filename For Input As #1 While EOF(1) = 0 Line Input #1, tmp Printer.Print Spc(5); tmp Printer.Print Wend Close #1

For a full overview of all the code used in the program I have created and not just the main procedures/functions please refer to page .

Whilst implementing my program of course various variables are needed in order to make the program work efficiently and effectively; below I will list the variables I have used, their data type and what they are used for.

Variable List

Variable	**Data Type**	**Used For**	**Comment**
Vectori	Integer (Array)	Used for the i coordinates of the vector (x axis)	Only whole vector coordinates stored so no need for real, array holds 2 values, must be between -15 & 15.

Vectorj	Integer (Array)	Used for the j coordinates of the vector (y axis)	Only whole vector coordinates stored so no need for real, array holds 2 values, must be between -15 & 15.
Scalar	Integer	Used as the multiplier to see whether vector 1 is parallel to vector 2.	Only whole values need to be stored, must be between -15 & 15.
Name	String	Stores the Student's Name.	Input in on the Main Form and Written to File
XMin	Integer	The minimum value on the x axis.	A constant value that does not change, declared before run time.
XMax	Integer	The maximum value on the x axis.	A constant value that does not change, declared before run time.
YMin	Integer	The minimum value on the y axis.	A constant value that does not change, declared before run time.
YMax	Integer	The maximum value on the y axis.	A constant value that does not change, declared before run time.
Parallel	Boolean	The result of a function to decide whether 2 vectors are parallel to one another.	
Answer	Double	The result of a function that is then input into the GetAngle function to calculate the angle between 2 vectors.	Needs to be a double so is precise when entered into the GetAngle Function.
Angle	Double	The result of the GetAngle function, the	Is a double for precision, then

		angle between 2 vectors.	formatted to 2 decimal places.
StuAns	Double	The answer the student inputs to see if they are correct.	Compared to the actual answer.
StuCorrect	Boolean	Set to True/False, used as an IF statement as to whether to increment StuScore.	
StuScore	Integer	Used to store how many answers the student has answered correctly.	Result written to text file.
QuestionsAns	Integer	Used to store how many questions the student has answered.	Result written to text file.
Distance	Double	Used to store the distance between 2 vectors, calculated from the GetDistance Function.	

Appraisal

Appraisal

Comparison of original objectives against finished system

1. *The learning system must cover the 3 most difficult topics of vectors : working out if two vectors are parallel to one another, working out angles between vectors and working out the distance between 2 non-parallel vectors.*

   Result
   This objective has been completed effectively, the 3 main topics of vectors are covered on three separate forms on the system : Parallel Vectors, Angles & Vectors & Vector Distances. Each form is clearly labelled and covers problems on each of the topics, so the student is easily able to identify which problem they want to solve and learn effectively from it.

2. *The proposed system must include diagrams of the vectors drawn in real time along with an explanation of what is occurring to help the students.*

   Result
   Every time the student enters their vector problem into any of the pages a step by step diagram is drawn on the appropriate graph complete with a mapped animation to show the movement between the 2 vectors, this is shown as the red-drawn lines on the graph, with the starting and finishing vector being mapped as 2 black lines. On the distance vector page the graph containing the diagrams also contains a blue line that shows the distance between the 2 vectors. Along with this an explanation is written to the form containing the formulae, the relevant answer, an explanation and feedback for the student at the same time.

3. *The proposed system must then produce transformation in real time of these vectors for the topic if these vectors are parallel to one another.*

Result
As stated above this objective has been fulfilled not only on the parallel vectors form but on all of the problem forms, the transformation is mapped as an animation using a recursive algorithm that maps the transformation of one vector to another by drawing a series of red lines to the form.

4. *The proposed system must also draw a line between the 2 non-parallel vectors to show students what distance the computer is working out for them.*

   Result
   As also stated above this has been achieved using a blue line that is drawn between the starting and finishing vector to show the distance between the 2.

5. *The student will be able to input the coordinates of the vectors, the computer will then draw these out and show the student how they would work out the given problem.*

Result
This objective has been managed, the students enter their vector coordinates in the relevant text boxes on each of the forms, once the calculate button has been clicked the computer then draws these out on the graph and provides them with an explanation & formulae as appropriate.

6. *The system must be easy to navigate around and will consist of 3 main sections for each of the 3 main topics.*

   Result
   The system is laid out in an easy to manage way, the main form clearly links to each of the separate 3 sections, that are all laid out on separate forms for the student to access as and when they are needed.

7. *Colours must be incorporated to make sure the system isn't too mundane, as vectors aren't particularly a fun subject.*

   Result
   The splash screen that loads before the Main Menu is full of colour and once the main menu is reached each page is differentiated by a colour theme for each problem page; once onto the main pages obviously too much colour couldn't be used as it could pose a distraction from the actual problem solving but colours were incorporated onto the vector graphs to show the transformations of vectors & the distances between them, to highlight the different nature of them compared to the original vectors.

8. *Have an option for students to print out certain vector diagrams/explanations.*

   Result
   This objective has been completed & fulfilled, by clicking the print graph button located on any of the 3 main problem forms the student can obtain a hard copy of their vector graph complete with the relevant explanation & feedback.

9. *For the student to be able to quit the system at any time if they do not wish to continue.*

   Result
   The student can exit the system at any time by clicking any of the Quit buttons, located on the Main Menu & the 3 separate problem pages.

10. *For the student to be able to input their answer to the question and be provided with feedback if their answer is wrong/right.*

    Result
    By inputting their answer into the relevant answer text box the student will be provided with feedback as to if their answer is correct along with the formulae & explanation that is also displayed.

11. *For the scores, the students' score on the program will be written to a small file so Mr Mason can see how each student is performing on the different topics.*

    Result
    On each of the separate problem pages the students' score is recorded along with how many questions they have answered, this is written to a file along with their name so Mr Mason can see which students are performing best on which topics. This file can be accessed by Mr Mason at any time.

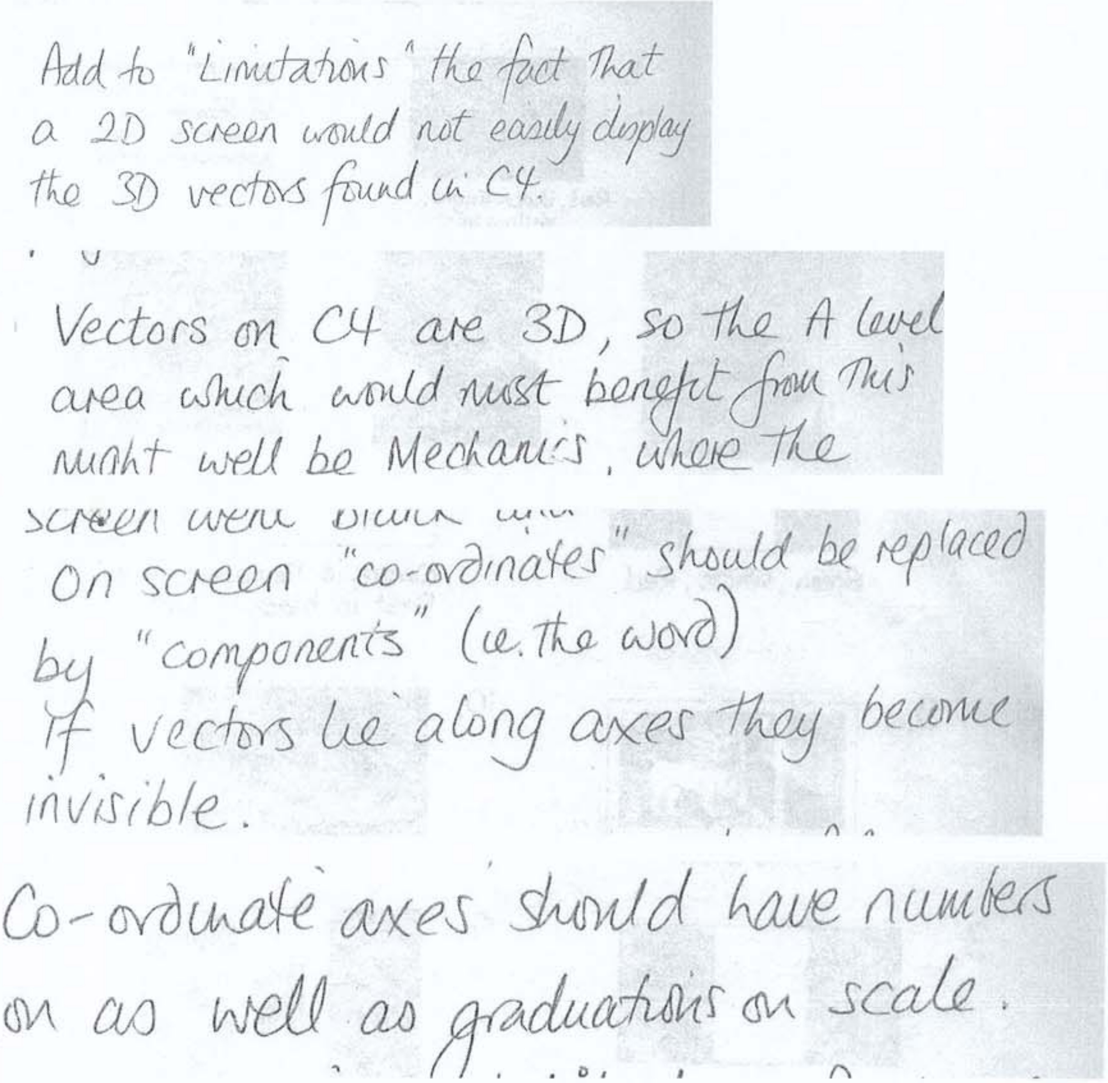
12. *For Mr Mason to be able to obtain a hard-copy of the students' results.*

Result

This objective has been fulfilled, located on the Main Menu form Mr M can click the Print Students Results button which will read from the scores file and print them out for him in order for him to obtain the necessary hard copy.

User Appraisal

A questionnaire filled out by Mr M & one of the students can be found attached on the following page; here they have outlined their use with the system and any improvements that could be made to the system. In addition to this I had a verbal discussion with Mr M where he went into more detail with some of the aspects of the system he may have liked changing slightly and provided me with some rough notes that can be seen below.



Add to "Limitations" the fact that a 2D screen would not easily display the 3D vectors found in C4.

Vectors on C4 are 3D, so the A level area which would most benefit from this might well be Mechanics, where the

screen were black

On screen "co-ordinates" should be replaced by "components" (ie. the word)

If vectors lie along axes they become invisible.

Co-ordinate axes should have numbers on as well as graduations on scale.

Although ultimately Mr M was happy with this system; the students interacted with the system well and it supported the standard teaching methods that he would use to teach this topic of C4 Vectors; print outs & hard copies were easy for him to obtain and apart from "a few minor hiccoughs" the program worked well; there were a few suggestions that both he & the student made that could have improved my program further.

1. To use 3d Vectors instead of 2d vectors

The use of a 3rd component, z, is used in A Level maths to represent a 3rd dimension that is incorporated into vector calculations; however due to limitations on the program it was impossible for me to represent this on the graphs that the system draws; therefore I believed that it would be less confusing for the student to incorporate this 3rd dimension when they wouldn't be able to see it. However if I was to redo the program then maybe the z component could be incorporated in calculations but explained to the student that the graphs drawn would only be drawn using the, i, and j coordinates so as not to confuse them.

2. A little more instruction to introduce the student to the program

Mr M and the students although managing to use the system relatively easily commented that it may have been beneficial for there to be more instruction in certain areas of the program sp they knew what to do; I believe on doing this again this could be achieved by as well has having instructions in the user manual on each of the separate vector pages there could be a small paragraph of text that detailed what the student should do on each of these pages to ensure that the program works correctly.

3. To use the word "component" instead of "co-ordinate" in the system

Mr M believed it would be better to use the correct terminology throughout the system and instead of using the word co-ordinate to describe vectors I & j he believed the word component would be more suitable. If editing was to be done I would make sure to change this within my system; however I believed that the word co-ordinate would confuse the student less and give them a better idea of exactly what they were dealing with.

4. Co-ordinate axis should have numbers on as well as graduations on scale

The axis would have benefitted from numbers on to try and save the student having to count along to ensure that their vector graphs had been drawn correctly; I did try to incorporate numbers onto the scales of the graph but I believe it may have looked slightly cluttered as the numbers would have to be incredibly small so as not to deter attention away from the task in hand. If amendments were to be made I may have to look at a different option of trying to fit numbers onto the axis without the system and graph looking cluttered.

5. If vectors lie along axis they become invisible

Obviously in my system the vectors are drawn in black, apart from the animation vectors & the distance line, there evidently is a problem here if the vector lies along the axis, which Is also drawn in black, as the vector can no longer be seen. To improve this I would simply change the colour of the vectors that were to be drawn to ensure they would stand out from the axis.

Ultimately if I was to re-do the project I would make sure to incorporate the above things into my system to improve how it works & how it interacts effectively with the student, obviously some of the extensions would be virtually impossible to code in VB6 such as the use of 3d vectors but many of the improvements could be easily done via re-coding and editing to the system allowing for more efficient use and an improved experience for both Mr M and his pupils.

User Appraisal Questionnaire

How easy did you find the system to use?

Very easy, buttons were well laid out. Maybe some more instructions on problem pages.

Do you think the new system is an effective new way to learn about vector problems?

Yes, makes it more fun and very helpful.

Did you enjoying using the system?

Yes, made learning vectors more fun.

Have you encountered any problems with the system during the time you have been using it?

No.

Have you easily been able to retrieve hard copies of your results?

Yes, I obtained print outs which I liked.

In terms of the layout of the system do you feel it was appropriate for the task in hand?

Yes I liked how graphs were drawn.

In terms of what the system does do you feel there is anything that could be improved upon or anything that you feel was missing from the system?

Vectors are 3D but this would probably be too confusing because they couldn't be drawn on the graph.
More instructions on problem pages.

4.4.11 C

User Appraisal Questionnaire

How easy did you find the system to use?

The system was user-friendly and required only a small amount of mathematical knowledge to use. A little more instruction would have helped in places.

Do you think the new system is an effective new way to teach the students about vector problems?

I think it would support standard teaching methods and allow the students to interact with the vectors.

Do the students appear to be enjoying using the system?

Yes, when given clear instructions to supplement the instructions in the program.

Have you encountered any problems with the system during the time you have been using it?

A few minor hiccoughs, but basically the software worked quite well.

Have you easily been able to retrieve hard copies of the students' results?

Hard copies for each student were easy to obtain.

In terms of the layout of the system do you feel it was appropriate for the task in hand?

Yes, again backed up by explanation from the teacher. No software is totally "stand alone".

In terms of what the system does do you feel there is anything that could be improved upon or anything that you feel was missing from the system?

Clearly 3D vectors would be better, but almost impossible to represent on a flat screen. Otherwise, please see comments and suggestions made in rest of questionnaire.

C [redacted] G. [redacted]

User Manual

User Manual Contents

Introduction to Vectors Solved Program :

Vectors Solved is a program aimed towards A Level maths students studying Core 4 maths, it is based on Chapter 5, Vector Problems and allows the user to interact with the system by answering questions in order to reinforce their learning of key topics within the Vector Chapter. The student can answer questions on various problems on different forms such as; are two vectors parallel; the angle between 2 vectors and the distance between two vectors. The student interacts with the program by inputting their chosen vector coordinates and what they think the answer is, they are then provided with feedback, formulae and a graphical drawing of their chosen vectors. In accordance with this the students results of how they perform on the questions are written to a file in order for their teacher to assess what topics are causing more of a problem with the students.

Print outs can be obtained of both the problems that the students have attempted to solve and the students' scores as a hard copy for the teacher.

Requirements of System :

In order for the program to run efficiently the following are recommended :

Windows XP (or higher)
Pentium 4 2 GHz Processor
1Gb RAM
40Gb Hard Drive

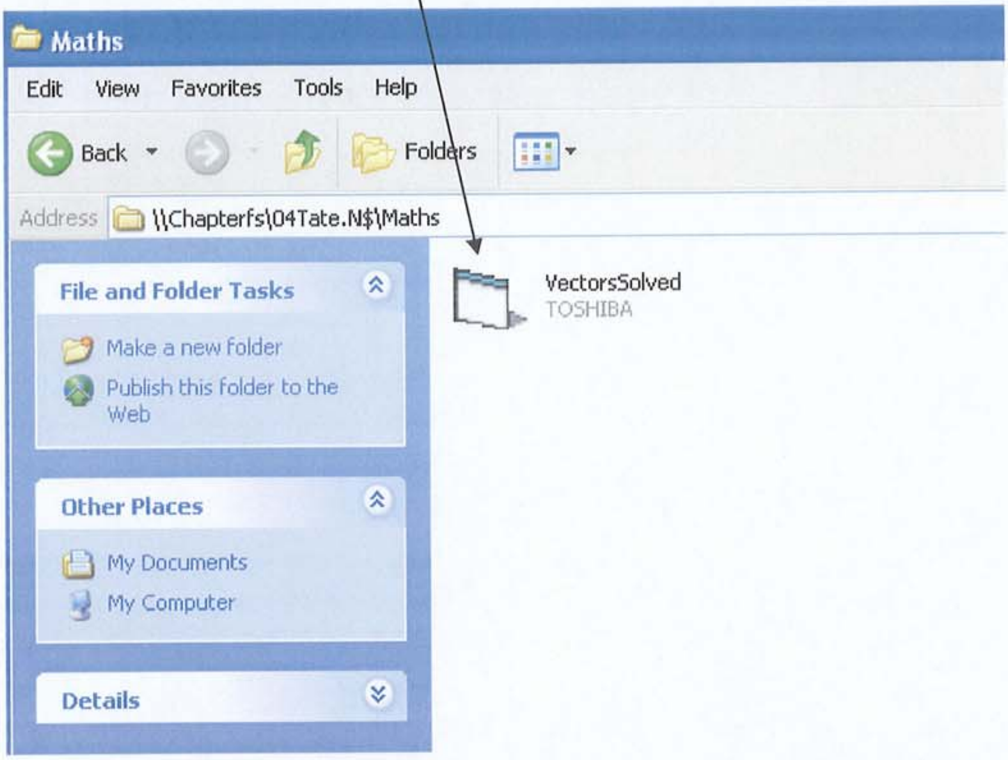
Necessary Peripherals :

The following peripherals will be needed to make sure the user can interact correctly with the program :

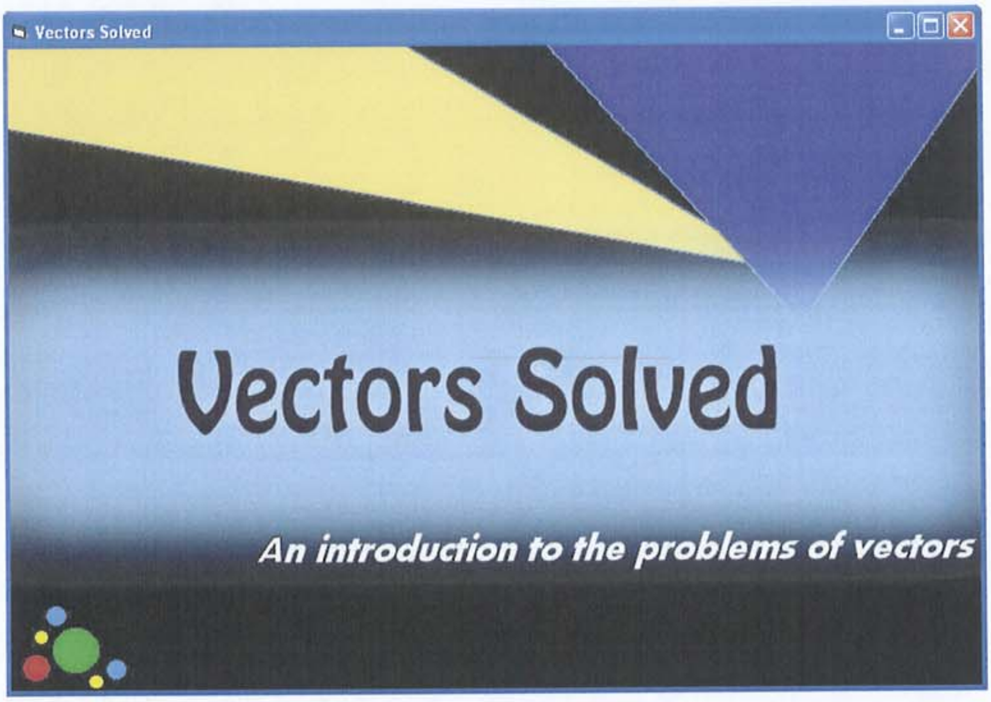
Keyboard
Monitor
Mouse/Touch Screen Monitor
Printer (Colour or Black & White)

Installing the Program

To install and use the program is extremely simple as the program is a compiled executable file that can be ran simply by double clicking on the icon that will be located in the relevant maths folder on the School network/system.

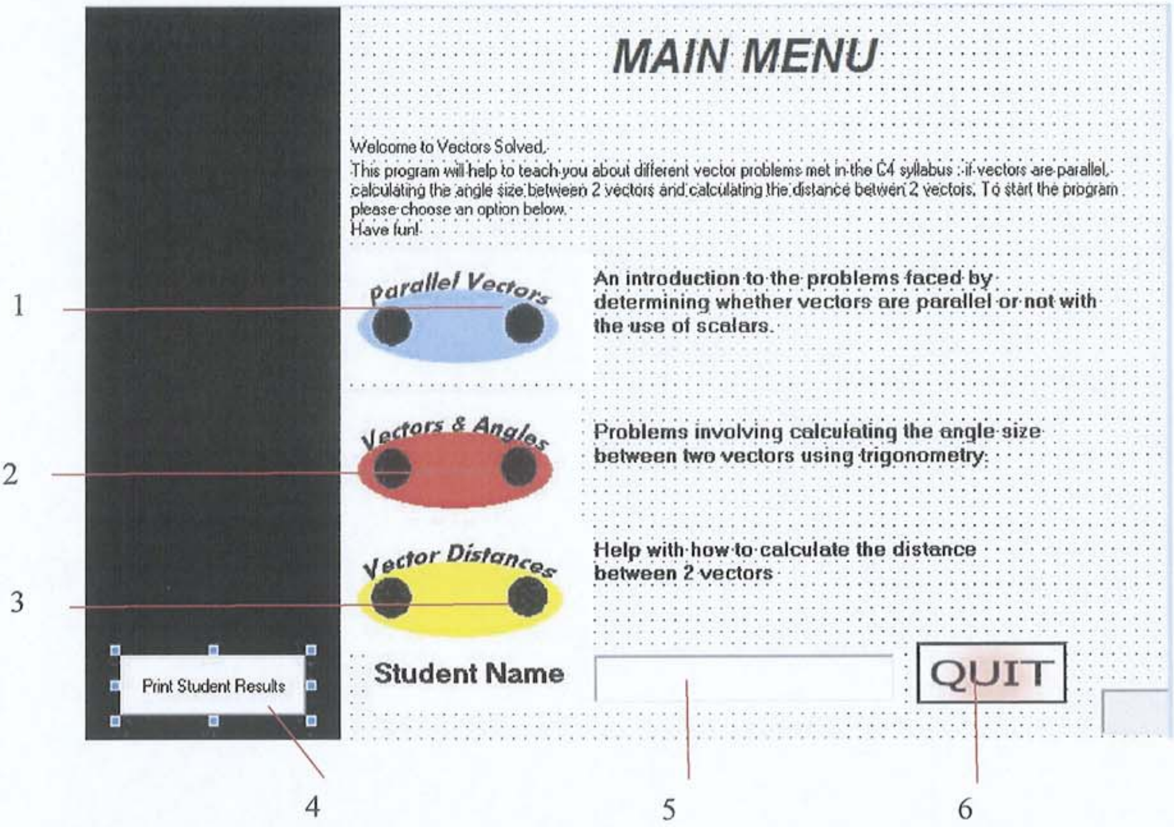


On double clicking the icon the program will open and introduce you to the program.



Layout of the system

Main Menu



Welcome to Vectors Solved.
This program will help to teach you about different vector problems met in the C4 syllabus : if vectors are parallel, calculating the angle size between 2 vectors and calculating the distance betwen 2 vectors. To start the program please choose an option below.
Have fun!

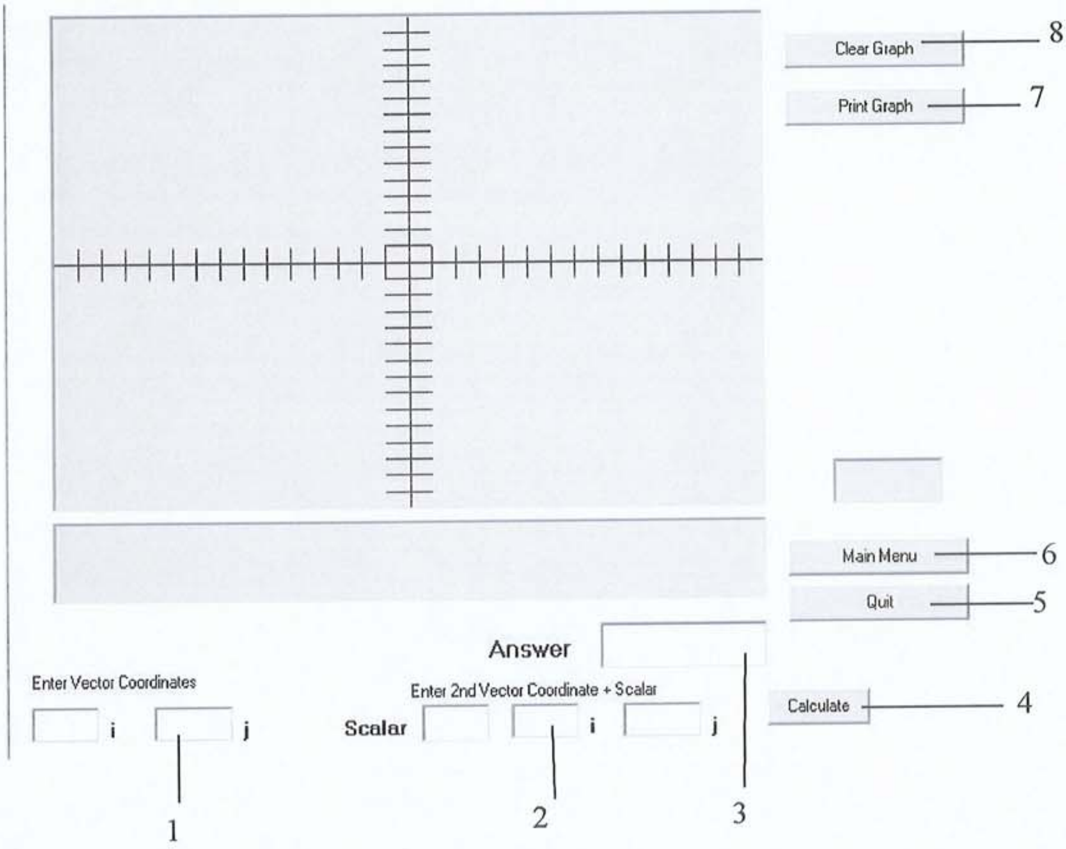
An introduction to the problems faced by determining whether vectors are parallel or not with the use of scalars.

Problems involving calculating the angle size between two vectors using trigonometry.

Help with how to calculate the distance between 2 vectors

1.Parallel Vectors Link Button – This button will take you to the Parallel Vectors problem page.
2. Vectors & Angles Link Button – This button will take you to the Vectors & Angles problem page.
3. Vector Distances Link Button – This button will take you to the Vector Distances problem page.
4. Print Student Results Button – This allows the teacher to print off Student Results of how they score on the problem pages.
5. Student Name – Where the student is required to input their name.
6. Quit Button – This button will allow you to exit and close the program.

Parallel Vectors Page



1. 1st Vector Coordinate Input Area – Where you input your 1st vector I and vector J coordinates.

2. 2nd Vector Coordinate Input Area – Where you input your 2nd vector I and vector J coordinates along with a scalar.

3. Answer Text Box – Input your answer to your given problem here.

4. Calculate Button – Once all text boxes have been filled make sure to click this button to begin the problem.

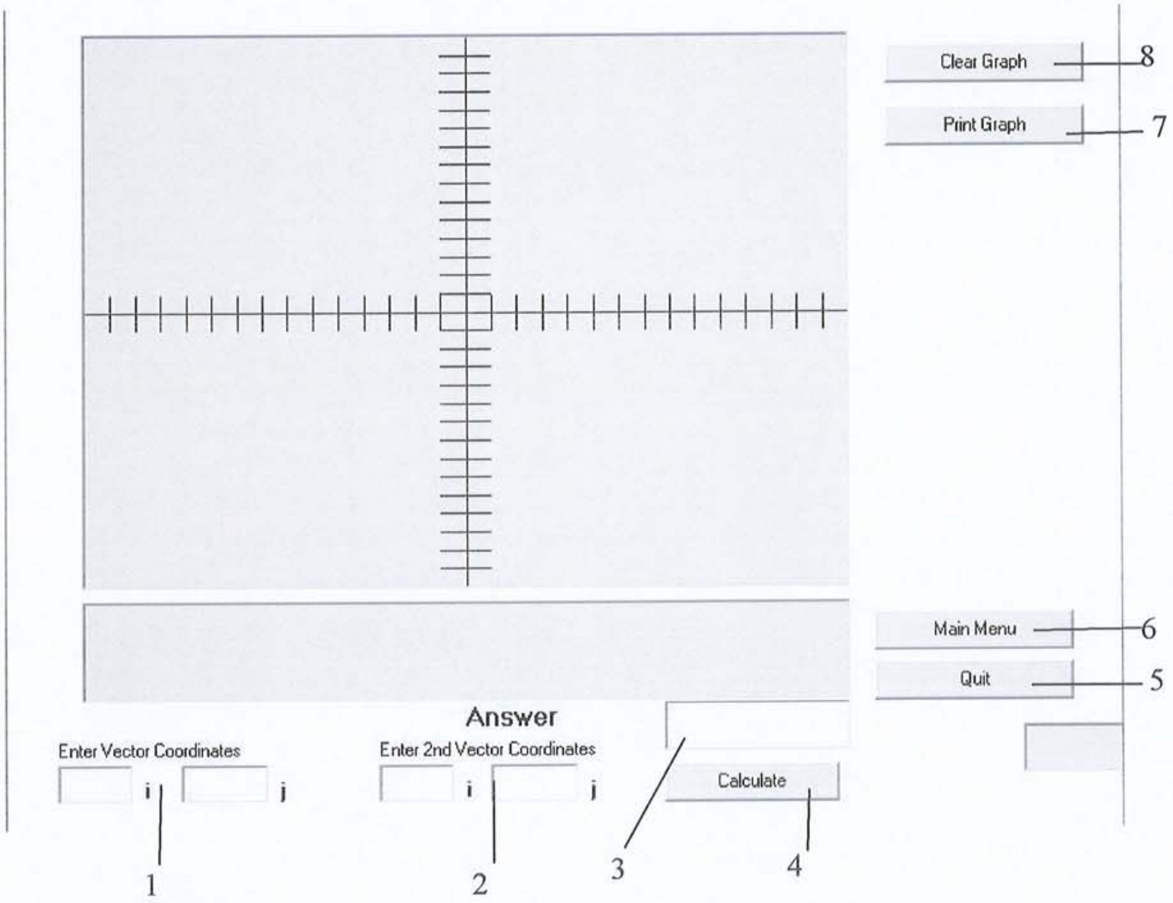
5. Quit Button – This button allows you to close and exit the program.

6. Main Menu Button – This button will navigate you back to the Main Menu.

7. Print Graph Button – This button will initiate the print process and produce a hard copy of the graph and answer for you.

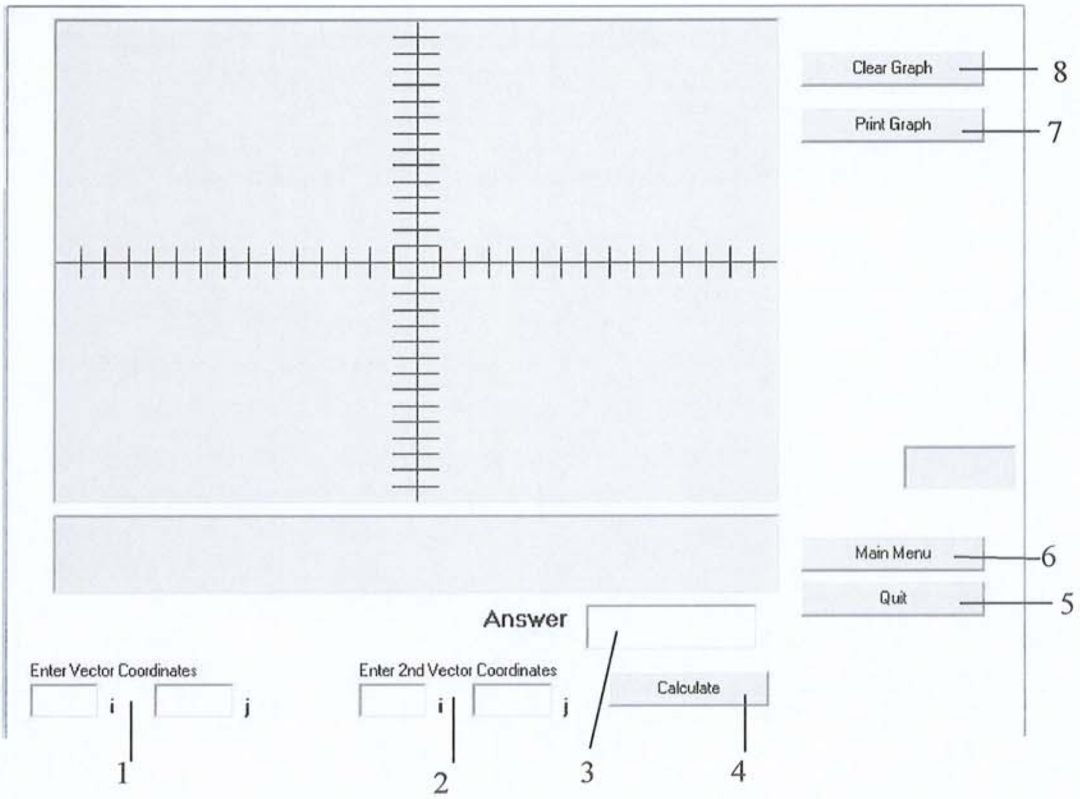
8. Clear Graph Button – This button clears the graph & the explanation box of your problem and allows you to have another go.

Vectors & Angles Page



1. **1st Vector Coordinate Input Area** – Where you input your 1st vector I and vector J coordinates.
2. **2nd Vector Coordinate Input Area** – Where you input your 2nd vector I and vector J coordinates.
3. **Answer Text Box** – Input your answer to your given problem here, to 2d.p.
4. **Calculate Button** – Once all text boxes have been filled make sure to click this button to begin the problem.
5. **Quit Button** – This button allows you to close and exit the program.
6. **Main Menu Button** – This button will navigate you back to the Main Menu.
7. **Print Graph Button** – This button will initiate the print process and produce a hard copy of the graph and answer for you.
8. **Clear Graph Button** – This button clears the graph & the explanation box of your problem and allows you to have another go.

Vector Distances Page



1. 1st Vector Coordinate Input Area – Where you input your 1st vector I and vector J coordinates.

2. 2nd Vector Coordinate Input Area – Where you input your 2nd vector I and vector J coordinates.

3. Answer Text Box – Input your answer to your given problem here, to 2d.p.

4. Calculate Button – Once all text boxes have been filled make sure to click this button to begin the problem.

5. Quit Button – This button allows you to close and exit the program.

6. Main Menu Button – This button will navigate you back to the Main Menu.

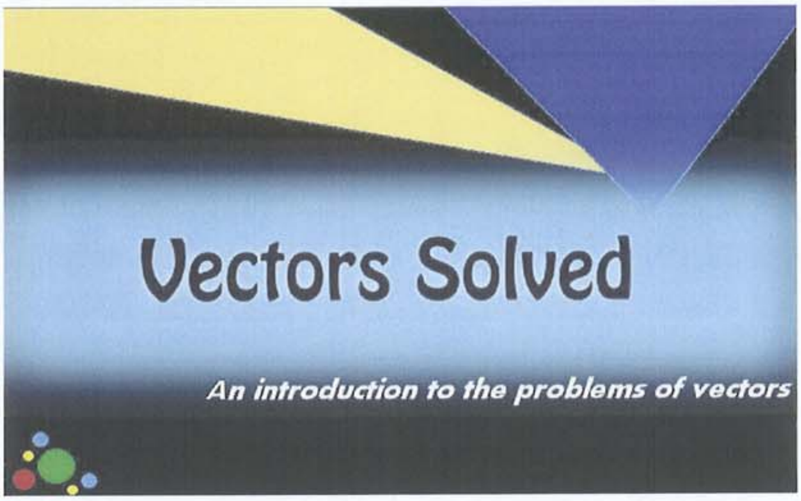
7. Print Graph Button – This button will initiate the print process and produce a hard copy of the graph and answer for you.

8. Clear Graph Button – This button clears the graph & the explanation box of your problem and allows you to have another go.

Using the Program

Launching the program :

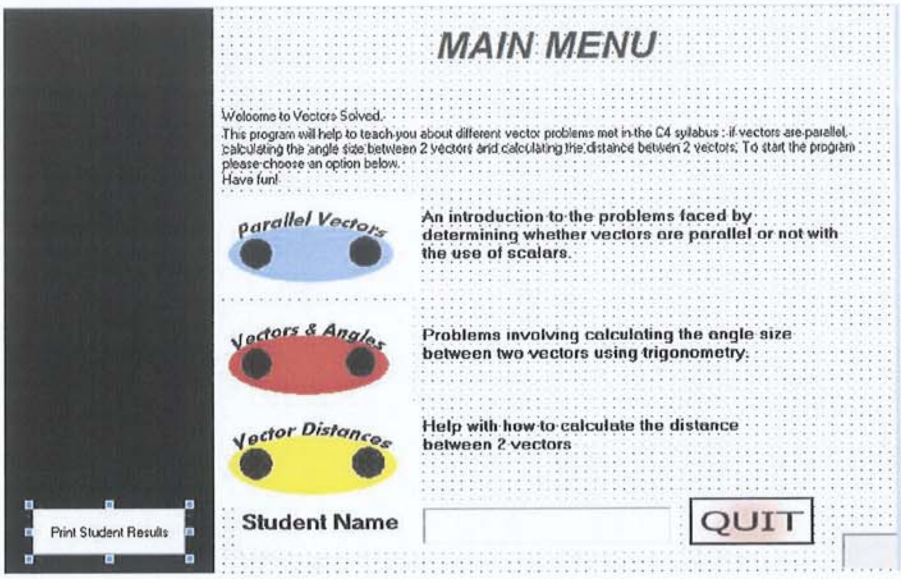
On launching the program a splash screen loads initially to introduce you to the program before the Main Menu appears with the various links to the different Vector Problem Pages.



Student Manual

Main Menu :

Before you are able to load any of the separate Vector pages make sure you enter your Name into the correct Student Name text box located at the bottom of the page. This will ensure that your name is correctly stored along with your scores. If you do not enter your name into the system an error message will appear, if this happens simply click ok and make sure you enter your name into the text box.



Welcome to Vectors Solved.
This program will help to teach you about different vector problems met in the C4 syllabus : if vectors are parallel, calculating the angle size between 2 vectors and calculating the distance betwen 2 vectors. To start the program please choose an option below.
Have fun!

An introduction to the problems faced by determining whether vectors are parallel or not with the use of scalars.

Problems involving calculating the angle size between two vectors using trigonometry.

Help with how to calculate the distance between 2 vectors

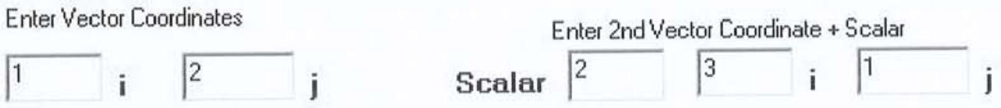
Once you have filled this in you are free to move on to any of the Vector Problem Pages you want : Parallel Vectors, Vectors & Angles or Vector Distances.

Every time you return to the Main Menu page, even if from a Vector Problem Page make sure to re-enter your name so that your results from each separate page will be stored.

Parallel Vectors Form :

The layout of all 3 problem pages is fundamentally the same with only slight differences between them depending on the type of problem you are trying to solve. On this form you are able to answer questions on whether 2 vectors are parallel to one another.

To begin with enter your first vector coordinates into the 2 text boxes at the far left side of the form, once these have been filled in you can then move onto entering your 2nd vector coordinates and the scalar you wish to multiply them by.



Enter Vector Coordinates

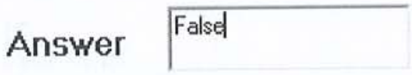
1 i 2 j

Enter 2nd Vector Coordinate + Scalar

Scalar 2 3 i 1 j

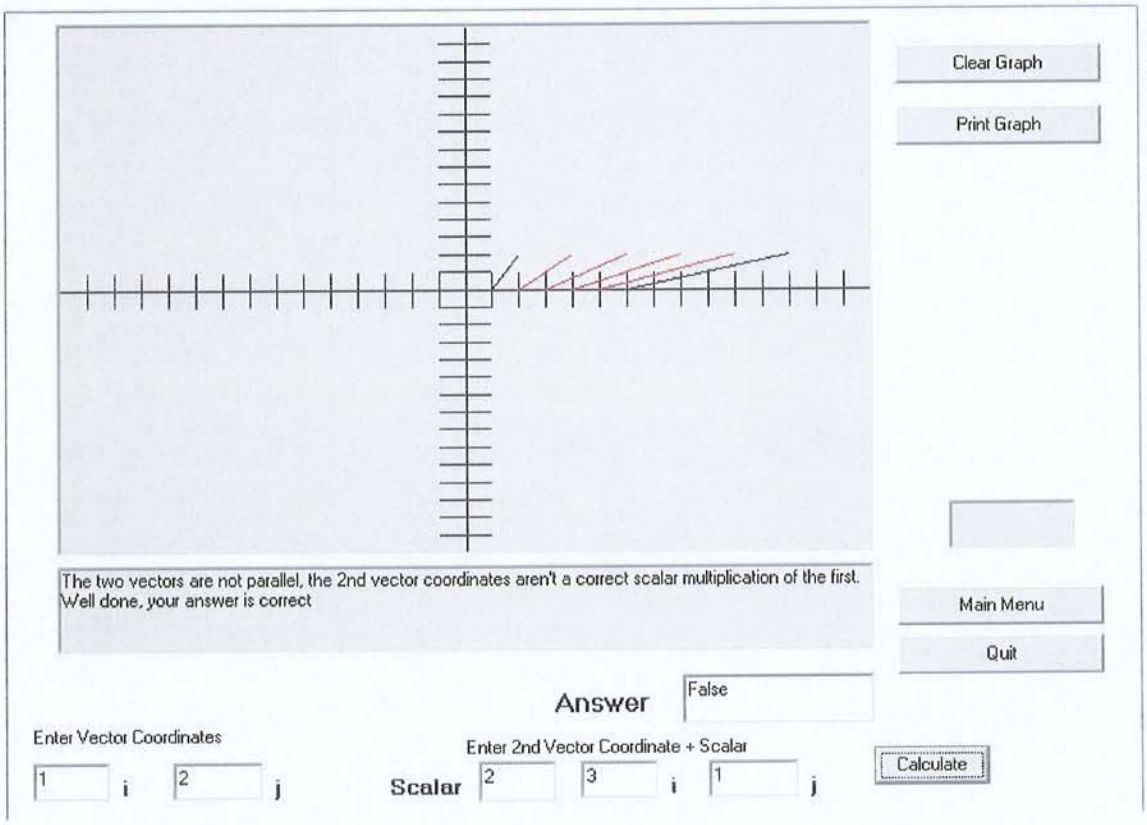
All of these must be numerical values and be between the values of -15 & 15 in order for the program to work correctly, if you do not enter valid data into the text boxes an error message will appear.

On completing these then make sure to enter your answer in the above text box marked 'Answer', for Parallel Vectors this will either be 'True' or 'False' depending on whether you think the vectors are parallel or not.



Answer False

After all text boxes have been filled in click the 'Calculate' button to see your problem drawn out along with an explanation, formulae and feedback as to whether you have answered the question correctly or not. The red lines on the vector graph represent the mapped movement between the 2 vectors with the ending black lines being your 1st & 2nd vector coordinates.



The two vectors are not parallel, the 2nd vector coordinates aren't a correct scalar multiplication of the first.
Well done, your answer is correct

Answer: False

Enter Vector Coordinates: 1 i 2 j

Enter 2nd Vector Coordinate + Scalar: Scalar 2; 3 i 1 j

Trying again:

If you wish to try another problem and clear the explanation box and the graph of the drawn vectors make sure to click the 'Clear Graph' button located at the top of the form, this will give you the chance to try a different problem or to have another go at your previous problem.

Printing your graph :

If you wish to obtain a print out of your chosen problem and explanation click the 'Print Graph' button, this will send a message to the printer and print your graph along with your explanation, the print out will also contain a copy of your name to prevent mix ups.

Navigating around the form :

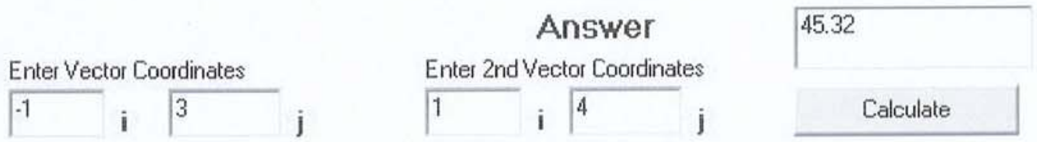
If you want to navigate away from the Parallel Vectors page and try another different type of problem click the 'Main Menu' button, this will take you back to the home page in which you can select a different form.

However if you have had enough of the program and wish to exit it click the 'Quit' button.

Vectors & Angles Form :

This form looks extremely similar to the Parallel Vectors form apart from this form deals with finding the size of the angle between 2 vectors. This form does not contain the scalar text box however all other text boxes are the same.

To begin your problem enter your first vector coordinates and second vector coordinates, once completed calculate what you think the correct size of the angle will be, to 2.d.p and enter this into the Answer text box. Once again make sure all your values are numeric and for the vector coordinates between -15 & 15 to avoid an error message.

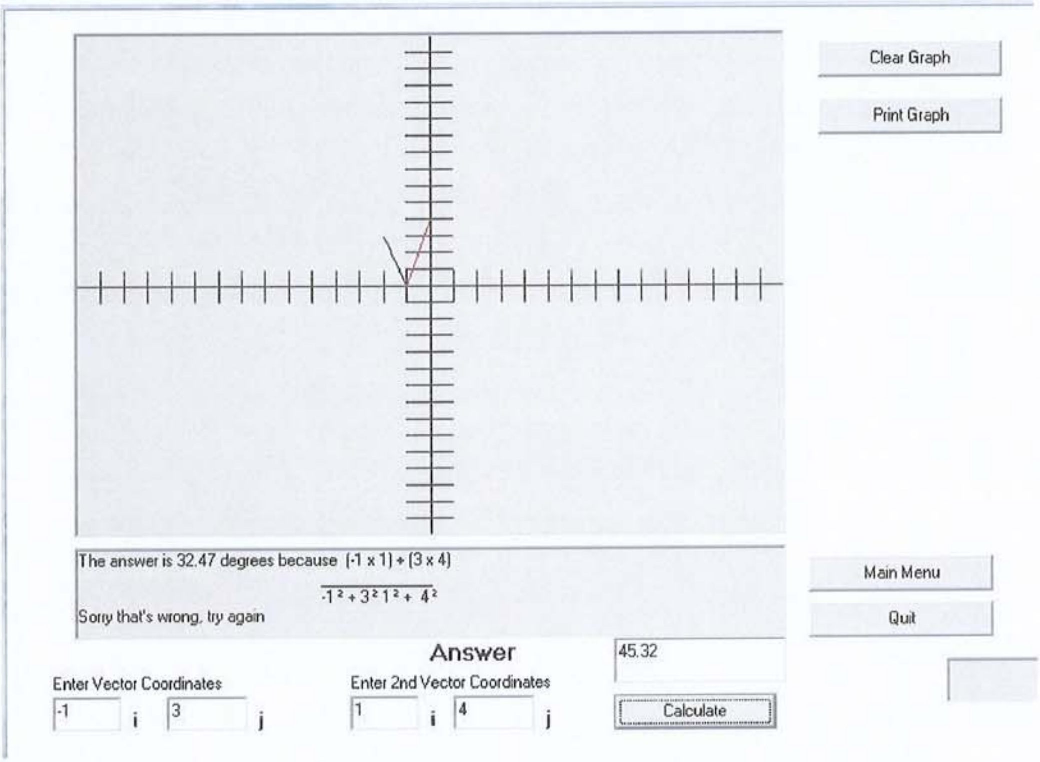


Answer: 45.32

Enter Vector Coordinates: -1 i 3 j

Enter 2nd Vector Coordinates: 1 i 4 j

On completion of all the text boxes click the 'Calculate' button to begin your problem, this will draw out your vector graph for you and provide you with feedback and an explanation of to whether you have answered the question correctly or not.



The answer is 32.47 degrees because (-1 x 1) + (3 x 4)

-1 ² + 3 ² 1 ² + 4 ²

Sorry that's wrong, try again

Answer: 45.32

Enter Vector Coordinates: -1 i 3 j

Enter 2nd Vector Coordinates: 1 i 4 j

Trying again:

If you wish to try another problem and clear the explanation box and the graph of the drawn vectors make sure to click the 'Clear Graph' button located at the top of the form, this will give you the chance to try a different problem or to have another go at your previous problem.

Printing your graph :

If you wish to obtain a print out of your chosen problem and explanation click the 'Print Graph' button, this will send a message to the printer and print your graph along with your explanation, the print out will also contain a copy of your name to prevent mix ups.

Navigating around the form :

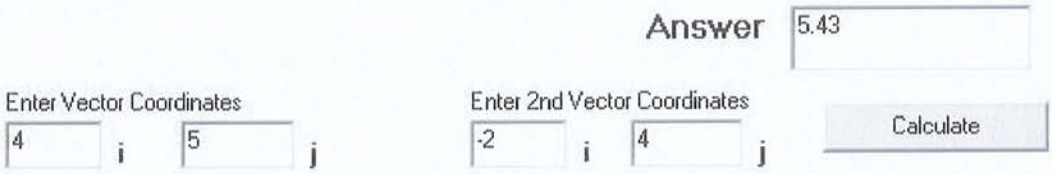
If you want to navigate away from the Vectors & Angles page and try another different type of problem click the 'Main Menu' button, this will take you back to the home page in which you can select a different form.

However if you have had enough of the program and wish to exit it click the 'Quit' button.

Vector Distances Form :

This is the 3rd and final problem page that you are able to use throughout the program, this deals with calculating the distance between 2 vectors, it contains the fundamental text boxes to enter your 1st & 2nd vector coordinates and your appropriate answer and all the relevant buttons to carry out specific tasks.

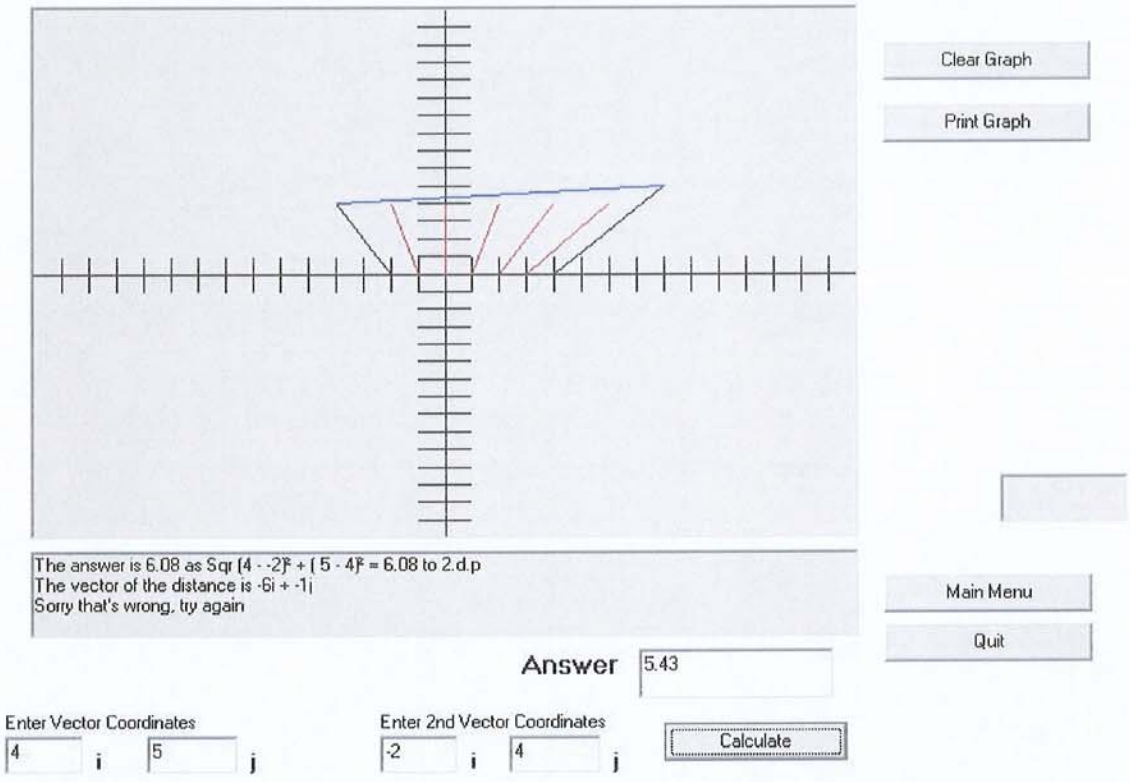
To begin your problem enter your first vector coordinates and second vector coordinates, once completed calculate what you think the correct distance between them will be, to 2.d.p and enter this into the Answer text box. Once again make sure all your values are numeric and for the vector coordinates between -15 & 15 to avoid an error message.


Answer 5.43

Enter Vector Coordinates 4 i 5 j

Enter 2nd Vector Coordinates -2 i 4 j

Clicking the 'Calculate' button will then draw out your vector problem for you and provide an explanation, feedback to your answer and details of the vector distance. On this vector graph, similar to the Parallel Vectors form the red lines represent the mapped movement between the 2 vectors, with the 2 black lines representing your 1st & 2nd vector coordinates. The difference on this graph however is the presence of a blue line, this line represents the distance between the 2 vectors.



The answer is 6.08 as Sqr (4 - -2)^2 + (5 - 4)^2 = 6.08 to 2.d.p
The vector of the distance is -6i + -1j
Sorry that's wrong, try again

Answer 5.43

Enter Vector Coordinates 4 i 5 j

Enter 2nd Vector Coordinates -2 i 4 j

Trying again:

If you wish to try another problem and clear the explanation box and the graph of the drawn vectors make sure to click the 'Clear Graph' button located at the top of the form, this will give you the chance to try a different problem or to have another go at your previous problem.

Printing your graph :

If you wish to obtain a print out of your chosen problem and explanation click the 'Print Graph' button, this will send a message to the printer and print your graph along with your explanation, the print out will also contain a copy of your name to prevent mix ups.

Navigating around the form :

If you want to navigate away from the Vector Distances page and try another different type of problem click the 'Main Menu' button, this will take you back to the home page in which you can select a different form.

However if you have had enough of the program and wish to exit it click the 'Quit' button.

Teacher Manual

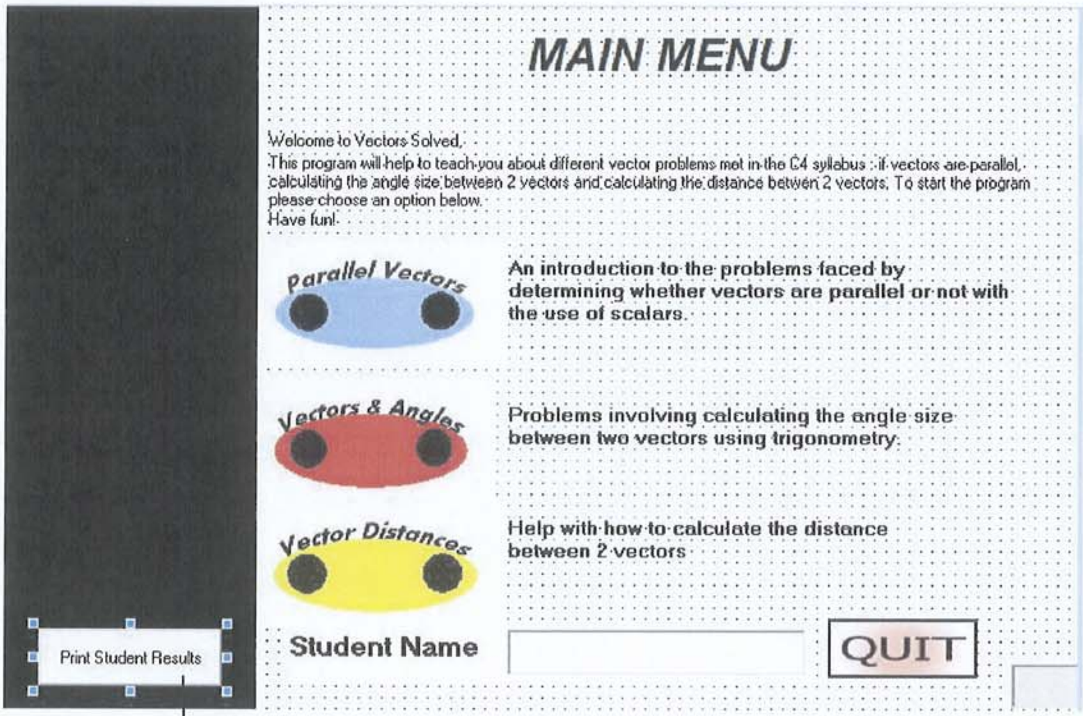
The only form that you will really need to access is the Main Menu, this will allow you to print out the relevant student scores to see how they are performing.

Obtaining student scores :

Every time the student answers a problem their results will be written to a file, in this file the students name, how many questions they have answered on what topic and the amount of questions they have answered correctly will be stored.

In order to do this just allow the Main Menu to load after the Splash page on launching the program, once the Main Menu has loaded simply click the 'Print Student Results' button located in the bottom left hand corner of the form.

This will send a message to the printer to print out a copy of the students' results, containing their name, the amount of questions they have answered on a certain topic and the number they have answered correctly.



Welcome to Vectors Solved,
This program will help to teach you about different vector problems met in the C4 syllabus : if vectors are parallel, calculating the angle size between 2 vectors and calculating the distance betwen 2 vectors. To start the program please choose an option below.
Have fun!

An introduction to the problems faced by determining whether vectors are parallel or not with the use of scalars.

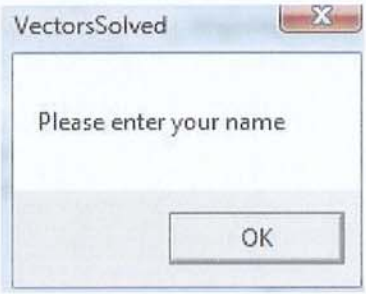
Problems involving calculating the angle size between two vectors using trigonometry.

Help with how to calculate the distance between 2 vectors

Click this button

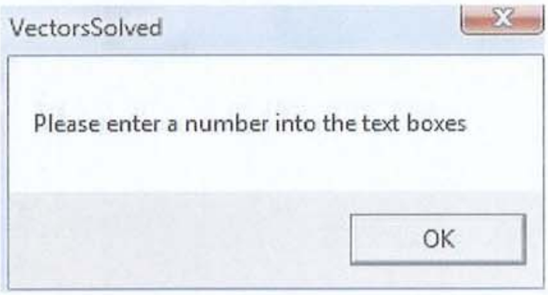
Error Messages

Various error messages can appear throughout using the system, below is a brief outline on what they are and how you can handle them :



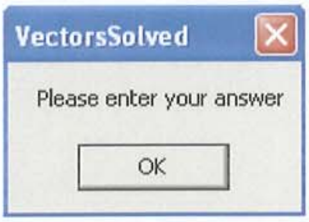
Please enter your name

This error occurs on the Main Menu if you have not entered your name into the appropriate text box, simply click ok and enter your name into the text box before moving on.



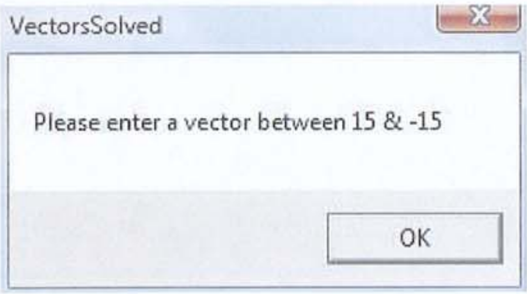
Please enter a number into the text boxes

This error can appear on any of the problem pages, it occurs if you have not entered a number into any of the text boxes. To move on from this simply click on and make sure all the text boxes are filled with the appropriate values.



Please enter your answer

This error occurs on any of the problem pages when you have not entered a number into the answer box; simply enter your answer and click the Calculate Button again.



Please enter a vector between 15 & -15

Again this error can appear on any of the problem pages and happens when you do not enter a vector coordinate between the valid ranges. To correct this click ok and enter a vector coordinate between -15 & 15.