GradeMax

Please check the examination details below before entering your candidate information

Candidate surname	Other names

Centre Number | Candidate Number

Pearson Edexcel International GCSE (9–1)

Monday 5 – Wednesday 7 June 2023

Time 3 hours

Paper reference 4CP0/02

Computer Science

PAPER 2: Application of Computational Thinking

You must have: A computer workstation with appropriate programming language code editing software and tools, including a code interpreter/compiler, CODES folder containing code and data files, and pseudocode command set (enclosed)

Total Marks

Instructions

- Use **black** ink or ball-point pen.
- **Fill in the boxes** at the top of this page with your name, centre number and candidate number.
- Answer **all** questions.
- Answer the questions **requiring a written answer** in the spaces provided – *there may be more space than you need.*
- Only **one** programming language (Python, C# or Java) must be used throughout the examination.
- Carry out practical tasks on the computer system and save new or amended code using the name given in the question with the appropriate file extension.
- Do **not** overwrite the original code and data files provided to you.
- You must **not** use the internet during the examination.

Information

- The total mark for this paper is 80.
- The marks for **each** question are shown in brackets – *use this as a guide as to how much time to spend on each question.*
- This paper covers Python, C# and Java.
- The CODES folder in your user area includes all the code and data files you need.
- The invigilator will tell you where to store your work.

Advice

- Read each question carefully before you start to answer it.
- Save your work regularly.
- Check your answers if you have time at the end.

Turn over ▶

P72938A

©2023 Pearson Education Ltd.
N:1/1/1/1/1/E2/1/

P 7 2 9 3 8 A 0 1 1 6

Pearson

Answer all questions.

Answer the questions requiring a written answer in the spaces provided.

Some questions must be answered with a cross in a box ☒. If you change your mind about an answer, put a line through the box ☒ and then mark your new answer with a cross ☒.

Carry out practical tasks on the computer system and save new or amended code using the name given with the appropriate file extension.

Use only ONE programming language throughout the examination.

Indicate the programming language that you are using with a cross in a box ☒.

C#	☒	Java	☒	Python	☒

1 Programs are used to handle financial transactions.

(a) Monthly account statements are created by a program.

Figure 1 shows a statement.

Date	Description	Debit (£)	Credit (£)	Balance (£)
01 Jan	Opening balance			128.35
04 Jan	Music store	10.23		118.12
07 Jan	Salary		1,515.28	1,633.40
11 Jan	Electricity company	50.00		1,583.40
19 Jan	Uniform shop	12.56		1,570.84
27 Jan	Refund		25.00	1,595.84
29 Jan	Rent	750.00		845.84
31 Jan	Interest		0.25	846.09

Figure 1

Complete the table to identify an input, an output and a process used by the program to generate the **furthest right column**.

(3)

Input	
Processing	
Output	

DO NOT WRITE IN THIS AREA
DO NOT WRITE IN THIS AREA
DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA
DO NOT WRITE IN THIS AREA
DO NOT WRITE IN THIS AREA

(b) A tax rate is applied to a gross value to give a net value.

Open **Q01b** in the code editor.

Use the code to answer these questions.

(i) Give the contents of a comment.

(1)

.....

(ii) Identify a logical operator used in this program.

(1)

.....

(iii) Give the name of a global variable.

(1)

.....

(iv) Give the name of a constant.

(1)

.....

(v) Give the name of a parameter.

(1)

.....

(c) Variables and constants are used in program code.

(i) State the purpose of a constant.

(1)

.....

.....

(ii) Give **one** benefit of using a constant.

(1)

.....

.....

(Total for Question 1 = 10 marks)

2 Solutions to problems are made up of many different components.

(a) Programmers use subprograms when developing code.

(i) Give **two** reasons to use **subprogram libraries** when developing code.

(2)

1 .....

.....

2 .....

.....

(ii) Identify which **one** of these must return a result.

(1)

☒ **A** Function

☒ **B** Iteration

☒ **C** Procedure

☒ **D** Selection

(iii) State what is meant by the term **local variable**.

(1)

.....

.....

DO NOT WRITE IN THIS AREA
DO NOT WRITE IN THIS AREA
DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA
DO NOT WRITE IN THIS AREA
DO NOT WRITE IN THIS AREA

(b) A school is planning a fish and chip dinner for the students and their families.

(i) Tickets are only sold to families.

A family is one to four adults and one to six children.

A program is being written to help manage ticket sales.

Complete the table to show examples of normal, boundary and erroneous numeric test data for the program.

(3)

	Adult	**Children**
Normal		
Boundary		
Erroneous		

(ii) The kitchen staff use a program to determine whether they have enough chips or how many more they need to order.

Open **Q02bii** in the code editor.

There are **four** errors in the code.

Amend the code to correct the errors.

Use this test data to help you find the errors.

Chips in stock (kilograms)	**Number of adults**	**Number of children**	**Expected output**
19	100	150	Stocks available
15	100	150	Order 4.0 kilograms

Save your amended code as **Q02biiFINISHED** with the correct file extension for the programming language.

(4)

(Total for Question 2 = 11 marks)

3 Algorithms can be represented in flowcharts, pseudocode or program code.

(a) Trace tables can be used with flowcharts or pseudocode.

Give **two** characteristics of a trace table.

(2)

1 .....

.....

2 .....

.....

(b) An algorithm has been written to validate numbers entered by the user.

Figure 2 shows the pseudocode for the algorithm.

```
 1 RECEIVE num1 FROM (INTEGER) KEYBOARD
 2 RECEIVE num2 FROM (INTEGER) KEYBOARD
 3 IF (((num1 < 16) AND (num2 < 23)) OR (num2 = 13)) THEN
 4     SEND "State 1" TO DISPLAY
 5 ELSE
 6     IF ((num2 > 12) AND (num1 > 20)) THEN
 7         SEND "State 2" TO DISPLAY
 8     ELSE
 9         IF ((num1 = 88) OR NOT (num2 = 18)) THEN
10             SEND "State 3" TO DISPLAY
11         ELSE
12             SEND "State 4" TO DISPLAY
13         END IF
14     END IF
15 END IF
```

Figure 2

Complete the table to show the output for each set of inputs.

(3)

num1	num2	Output
88	18	
17	18	
12	19	

DO NOT WRITE IN THIS AREA
DO NOT WRITE IN THIS AREA
DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA
DO NOT WRITE IN THIS AREA
DO NOT WRITE IN THIS AREA

(c) A program is required to calculate the result of raising one integer (the base) to the power of another (the exponent).

Figure 3 shows a flowchart for the algorithm.

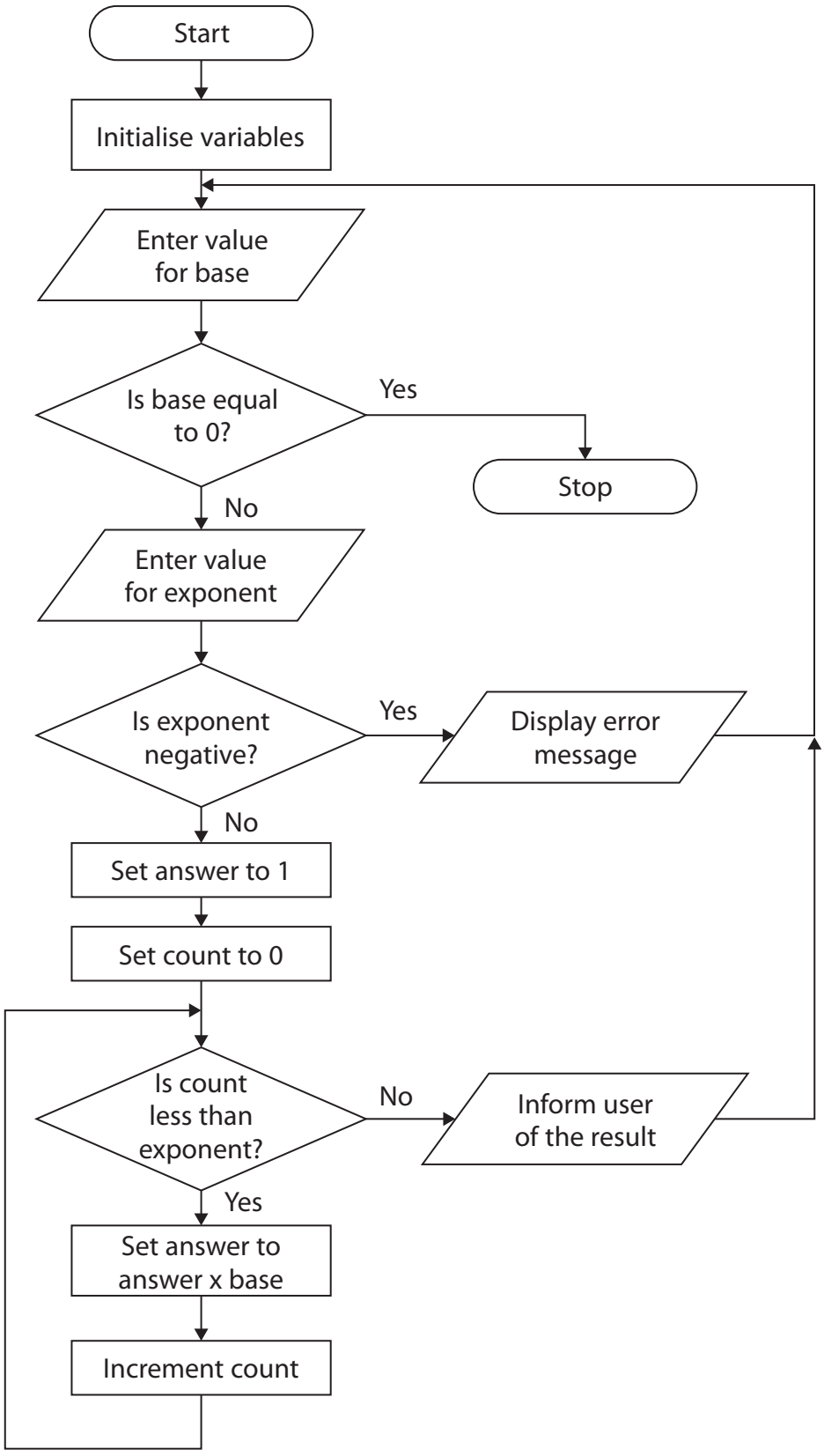

Figure 3

The program has these requirements:

- outputs meaningful error messages
- outputs the final answer with the base and the exponent.

Open **Q03c** in the code editor.

Write a program to implement the logic in the flowchart.

Do not add any further functionality.

Save your code as **Q03cFINISHED** with the correct file extension for the programming language.

(10)

(Total for Question 3 = 15 marks)

DO NOT WRITE IN THIS AREA
DO NOT WRITE IN THIS AREA
DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA
DO NOT WRITE IN THIS AREA
DO NOT WRITE IN THIS AREA

4 Data is encoded for encryption and for identifying records in databases.

(a) A Caesar cipher is one method of encryption.

(i) Complete the table to show the result of applying a Caesar cipher.

(2)

Plaintext	Shift	Ciphertext
PIXEL	–4	
CLOUD		FORXG

(ii) Plaintext is encrypted using a Caesar cipher.

Compare the ciphertexts produced for the plaintext **GOLD** by:

- a shift of –6 followed by a shift of +8
- a single shift of +2.

(2)

(iii) Ann has written a Caesar cipher algorithm.

When the algorithm encrypts the plaintext **BYTE** with a shift of +5 the result is **GVYJ**.

This is incorrect.

Explain the error.

(2)

(b) A program is needed to create a key for a database.

A user enters a two-letter word, a whole number and a decimal number.

The program must ensure the word is only two characters long.

The program must display an error message when the word is not two characters long.

A key is generated from the whole number, the reversed word and the whole number part of the decimal number.

Figure 4 shows the input values and a valid key.

```
Enter a word: at
Enter a whole number: 12
Enter a decimal number: 7.89
12ta7
```

Figure 4

Open **Q04b** in the code editor.

Write the program.

Do not add any further functionality.

Save your code as **Q04bFINISHED** with the correct file extension for the programming language.

(8)

(Total for Question 4 = 14 marks)

DO NOT WRITE IN THIS AREA
DO NOT WRITE IN THIS AREA
DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA
DO NOT WRITE IN THIS AREA
DO NOT WRITE IN THIS AREA

5 A marine scientist is conducting a study of tuna in the world's oceans.

(a) A list of tuna species needs to be sorted into **descending** alphabetical order using a merge sort algorithm.

Figure 5 shows the lists created at the end of the splitting process.

Each list is a single tuna species.

Bigeye	Blackfin	Albacore	Longtail	Bluefin

Figure 5

It will require three passes to merge the lists into a single sorted list in **descending** order.

Complete the merge sort using the space provided.

(2)

(b) A bubble sort could be used to sort the list of tuna species into **ascending** order.

Figure 6 shows an algorithm for a bubble sort.

```
 1 SET myTuna TO ["Bigeye", "Blackfin", "Albacore",
 2                "Longtail", "Bluefin"]
 3 SET tmp TO 0
 4 SET swaps TO True
 5 SET length TO LENGTH (myTuna)
 6
 7 WHILE (swaps = True) DO
 8     SET swaps TO False
 9     FOR ndx FROM 0 TO length - 1 DO
10         IF myTuna[ndx] > myTuna[ndx + 1] THEN
11             SET tmp TO myTuna[ndx + 1]
12             SET myTuna[ndx] TO myTuna[ndx + 1]
13             SET myTuna[ndx + 1] TO tmp
14             SET swaps TO True
15         END IF
16     END FOR
17 END WHILE
```

Figure 6

There is an error in the loop between line 9 and line 16.

Complete the table to give the line number with an error and a corrected line of pseudocode.

(2)

Line number with error	
Corrected line of pseudocode	

DO NOT WRITE IN THIS AREA
DO NOT WRITE IN THIS AREA
DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA
DO NOT WRITE IN THIS AREA
DO NOT WRITE IN THIS AREA

(c) The scientist is collecting and storing data about tuna.

The data collected is:

- species
- length in centimetres
- weight in kilograms
- age in years.

The data is stored in an array.

The collected data needs to be written to a file named **TunaData.txt**

Each record stored in the file must have a code number in the first field.

Code numbers must start at 101

Each field in a record should be separated by a comma.

Figure 7 shows the contents of the file.

```
101,Yellowfin,105,15,3
102,Albacore,90,15,5
103,Skipjack,50,3,4
104,Bigeye,105,25,4
105,Atlantic Bonito,50,4,2
106,Northern Bluefin,190,120,11
107,Southern Bluefin,190,120,11
108,Tongol,90,20,4
```

Figure 7

Open **Q05c** in the code editor.

Write the program.

You must use the structure given in **Q05c** to write the program.

Do not add any further functionality.

Save your code as **Q05cFINISHED** with the correct file extension for the programming language.

(6)

(Total for Question 5 = 10 marks)

6 An agricultural college has a herd of dairy cows.

Data collected for the cows is stored in arrays.

The data stored is:

- the name of the breed
- the ease of care rating for that breed (1 is highest and 3 is lowest)
- the number of cows in the herd of that breed
- the volume of milk per day for a cow of that breed.

The college wants to present the data to farmers and to recommend the best breed of cows for them.

The best breed has the highest care rating and the largest volume of milk per day for a cow of that breed.

Open the file **Q06** in the code editor.

Write a program to:

- calculate the daily volume of milk produced by each breed
- add this daily volume to the data structure tbl_dailyVolume
- display a message informing the user what each field holds, such as breed, rating, volume per cow, count and total volume
- display the data for each breed
- calculate and display the total volume of milk produced each day by the herd
- find the recommended breed
- display the recommended breed by name.

Figure 8 shows the output from a functional program.

```
Fields are: Breed, Rating, Volume (cow), Count, Volume (day)
Red Chittagong 1 7.5 6 45.0
Sussex 2 5.7 3 17.1
Dexter 3 11.4 8 91.2
Abondance 2 11.4 7 79.8
Sahiwal 3 22.0 6 132.0
Vorderwald 1 15.2 4 60.8
Ayrshire 2 21.0 3 63.0
Jersey 1 18.3 7 128.1
Randall 2 19.0 3 57.0
Alderney 1 9.0 3 27.0
Carora 3 23.1 4 92.4
Gloucester 2 16.0 7 112.0
Total: 905.4 litres
Recommended breed: Jersey rating: 1 volume 18.3
```

Figure 8

DO NOT WRITE IN THIS AREA
DO NOT WRITE IN THIS AREA
DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA
DO NOT WRITE IN THIS AREA
DO NOT WRITE IN THIS AREA

Your program should function correctly even if the number of breeds represented in the data is changed.

Save your amended code as **Q06FINISHED** with the correct file extension for the programming language.

You may use this space for planning/design work.

The content of this space will **not** be assessed.

(Total for Question 6 = 20 marks)

TOTAL FOR PAPER = 80 MARKS

BLANK PAGE

DO NOT WRITE IN THIS AREA
DO NOT WRITE IN THIS AREA
DO NOT WRITE IN THIS AREA

GradeMax

Pearson Edexcel International GCSE (9–1)

Monday 5 – Wednesday 7 June 2023

Paper reference **4CP0/02**

Computer Science

Component 2
Pseudocode command set

Resource Booklet
Do not return this Booklet with the question paper.

Turn over ▶

P72938A

©2023 Pearson Education Ltd.
N:1/1/1/1/1/1/E2/1/

P 7 2 9 3 8 A

Pearson

Pseudocode command set

Questions in the written examination that involve code will use this pseudocode for clarity and consistency. However, students may answer questions using any valid method.

Data types

INTEGER

REAL

BOOLEAN

CHARACTER

Type coercion

Type coercion is automatic if indicated by context. For example 3 + 8.25 = 11.25 (integer + real = real)

Mixed mode arithmetic is coerced like this:

	INTEGER	REAL
INTEGER	INTEGER	REAL
REAL	REAL	REAL

Coercion can be made explicit. For example, RECEIVE age FROM (INTEGER) KEYBOARD assumes that the input from the keyboard is interpreted as an INTEGER, not a STRING.

Constants

The value of constants can only ever be set once. They are identified by the keyword CONST. Two examples of using a constant are shown.

CONST REAL PI

SET PI TO 3.14159

SET circumference TO radius * PI * 2

Data structures

ARRAY

STRING

Indices start at zero (0) for all data structures.

All data structures have an append operator, indicated by &.

Using & with a STRING and a non-STRING will coerce to STRING. For example, SEND 'Fred' & age TO DISPLAY, will display a single STRING of 'Fred18'.

Identifiers

Identifiers are sequences of letters, digits and '_', starting with a letter, for example:
MyValue, myValue, My_Value, Counter2

Functions

LENGTH()

For data structures consisting of an array or string.

RANDOM(n)

This generates a random number from 0 to n.

Comments

Comments are indicated by the # symbol, followed by any text.

A comment can be on a line by itself or at the end of a line.

Devices

Use of KEYBOARD and DISPLAY are suitable for input and output.

Additional devices may be required, but their function will be obvious from the context.
For example, CARD_READER and MOTOR are two such devices.

Notes

In the following pseudocode, the < > indicates where expressions or values need to be supplied. The < > symbols are not part of the pseudocode.

Variables and arrays		
Syntax	**Explanation of syntax**	**Example**
SET Variable TO <value>	Assigns a value to a variable.	SET Counter TO 0 SET MyString TO 'Hello world'
SET Variable TO <expression>	Computes the value of an expression and assigns to a variable.	SET Sum TO Score + 10 SET Size to LENGTH(Word)
SET Array[index] TO <value>	Assigns a value to an element of a one-dimensional array.	SET ArrayClass[1] TO 'Ann' SET ArrayMarks[3]TO 56
SET Array TO [<value>, ...]	Initialises a one-dimensional array with a set of values.	SET ArrayValues TO [1, 2, 3, 4, 5]
SET Array [RowIndex, ColumnIndex] TO <value>	Assigns a value to an element of a two dimensional array.	SET ArrayClassMarks[2,4] TO 92

Selection		
Syntax	**Explanation of syntax**	**Example**

IF <expression> THEN
  <command>
END IF
	If <expression> is true then command is executed.	IF Answer = 10 THEN
		  SET Score TO Score + 1
		END IF

IF <expression> THEN
  <command>
ELSE
  <command>
END IF
	If <expression> is true then first <command> is executed, otherwise second <command> is executed.	IF Answer = 'correct' THEN
		  SEND 'Well done' TO DISPLAY
		ELSE
		  SEND 'Try again' TO DISPLAY
		END IF

Repetition		
Syntax	**Explanation of syntax**	**Example**

WHILE <condition> DO
<command>
END WHILE
	Pre-conditioned loop. Executes <command> whilst <condition> is true.	WHILE Flag = 0 DO
		SEND 'All well' TO DISPLAY
		END WHILE

REPEAT <command> UNTIL <expression>	Post-conditioned loop. Executes <command> until <condition> is true. The loop must execute at least once.	REPEAT SET Go TO Go + 1 UNTIL Go = 10
REPEAT <expression> TIMES <command> END REPEAT	Count controlled loop. The number of times <command> is executed is determined by the expression.	REPEAT 100-Number TIMES SEND '*' TO DISPLAY END REPEAT

FOR <id> FROM <expression> TO
<expression> DO
<command>
END FOR
	Count controlled loop. Executes <command> a fixed number of times.	FOR Index FROM 1 TO 10 DO
		SEND ArrayNumbers[Index] TO DISPLAY
		END FOR

FOR <id> FROM <expression> TO
<expression> STEP <expression> DO
<command>
END FOR
	Count controlled loop using a step.	FOR Index FROM 1 TO 500 STEP 25 DO
		SEND Index TO DISPLAY
		END FOR

FOR EACH <id> FROM <expression> DO <command> END FOREACH	Count controlled loop. Executes for each element of an array.	SET WordsArray TO ['The', 'Sky', 'is', 'grey'] SET Sentence to '' FOR EACH Word FROM WordsUArray DO SET Sentence TO Sentence & Word & ' ' END FOREACH

Input/output		
Syntax	**Explanation of syntax**	**Example**
SEND <expression> TO DISPLAY	Sends output to the screen.	SEND 'Have a good day.' TO DISPLAY
RECEIVE <identifier> FROM (type) <device>	Reads input of specified type.	RECEIVE Name FROM (STRING) KEYBOARD RECEIVE LengthOfJourney FROM (INTEGER) CARD_READER RECEIVE YesNo FROM (CHARACTER) CARD_READER

File handling		
Syntax	**Explanation of syntax**	**Example**
READ <File> <record>	Reads in a record from a <file> and assigns to a <variable>. Each READ statement reads a record from the file.	READ MyFile.doc Record
WRITE <File> <record>	Writes a record to a file. Each WRITE statement writes a record to the file.	WRITE MyFile.doc Answer1, Answer2, 'xyz 01'

Subprograms		
Syntax	**Explanation of syntax**	**Example**
PROCEDURE <id> (<parameter>, …) BEGIN PROCEDURE <command> END PROCEDURE	Defines a procedure.	PROCEDURE CalculateAverage (Mark1, Mark2, Mark3) BEGIN PROCEDURE SET Avg to (Mark1 + Mark2 + Mark3)/3 END PROCEDURE

FUNCTION <id> (<parameter>, …)
BEGIN FUNCTION
  <command>
RETURN <expression>
END FUNCTION
	Defines a function.	FUNCTION AddMarks (Mark1, Mark2, Mark3)
		BEGIN FUNCTION
		  SET Total to (Mark1 + Mark2 + Mark3)/3
		  RETURN Total
		END FUNCTION

<id> (<parameter>, …)	Calls a procedure or a function.	Add (FirstMark, SecondMark)

Arithmetic operators	
Symbol	**Description**
+	Add
-	Subtract
/	Divide
*	Multiply
^	Exponent
MOD	Modulo
DIV	Integer division

Relational operators	
Symbol	**Description**
=	equal to
<>	not equal to
>	greater than
>=	greater than or equal to
<	less than
<=	less than or equal to

Logical operators	
Symbol	**Description**
AND	Returns true if both conditions are true.
OR	Returns true if any of the conditions are true.
NOT	Reverses the outcome of the expression; true becomes false, false becomes true.

GradeMax

BLANK PAGE