1. José works for a company that provides loans to its customers. When customers take out a loan they decide how much money to borrow and for how many years.

The interest rate is currently 10% but it may change in the future.

José writes the following program to calculate the monthly payment for a loan.

```
01  PROGRAM LoanCalculator
02
03      CONST InterestRate = 10
04
05  BEGIN
06      INPUT Amount
07      INPUT Years
08      AnnualInterest = Amount * InterestRate / 100
09      TotalToPay = (AnnualInterest * Years) + Amount
10      MonthlyPayment = TotalToPay / (Years * 12)
11      OUTPUT MonthlyPayment
12  END
```

Parentheses have been used in lines 09 and 10.

(i) State why the parentheses in line 09 are **not** essential.

---

[1]

(ii) Explain why the parentheses in line 09 are useful.

---

[2]

(iii) Explain why the parentheses in line 10 are essential.

---

[2]

2. A company organises a word guessing game to be played using text messages. Players have to guess a six letter word and send it to the company's computer. The program which processes the message contains several subroutines.

The code for one of these subroutines is shown below.

```
01  PROCEDURE ReceiveMessage(Message, PhoneNumber)
02    Message = UPPERCASE(Message)
03    IF LENGTH(Message) <> 6 THEN
04      Result = "INCORRECT LENGTH"
05    ELSE IF NotInDictionary(Message) THEN
06      Result = "UNKNOWN WORD"
07    ELSE
08      Result = CheckAnswer(Message)
09    END IF
10    SendMessage(Result, PhoneNumber)
11  END PROCEDURE
```

The subroutine ReceiveMessage has two parameters.

(i) Define the term parameter.

---

[2]

(ii) State the names of the **two** parameters of the procedure ReceiveMessage. For each parameter, state the most appropriate data type.

---

[3]

3(a). Some whole numbers are known by mathematicians as evil numbers.
One way to find out if a number is evil, is to use the integer division operators DIV and MOD.

Describe how iteration has been used in this function.

---

[2]

(b). 0 is an evil number.

Describe each step of the execution of the call IsEvil(0), showing that it returns the value TRUE.

---

[3]

(c). Using the trace table below, show what happens in the execution of the call IsEvil(2), showing that 2 is not an evil number.

You should use a new row in the table for every line that is executed, and show any values that are changed during the execution of that line. Yo u may not need every row in the table. The first two rows have been filled in for you.

Line Number	n	Temp	Comment
01	2		Call IsEvil(2)
02		TRUE	Temp = TRUE

[6]

All numbers that are not evil are known as odious numbers.

The following function determines whether a number is odious.

```
01    FUNCTION IsOdious(n : INTEGER)
02      IF n = 0 THEN
03        RETURN FALSE
04      ELSE
05        IF n MOD 2 = 0 THEN
06          RETURN IsOdious(n DIV 2)
07        ELSE
08          RETURN NOT(IsOdious(n DIV 2))
09        END IF
10      END IF
11    END FUNCTION
```

4.

(i) A procedural programming language may use procedures.

Explain the term procedural programming language.

---[2]

(ii) The same variable name may be used in more than one procedure in a program.

Explain how a variable named result may be used in different procedures without causing errors.

---[2]

(iii) Explain parameter passing.

---[5]

5. Variables are used in programming.

(i) Describe the use of local variables.

---

[4]

(ii) State **two** features of global variables that distinguish them from local variables.

1

2

---

[2]

6. A variable can be declared as global or local and is said to have scope.

(i) Explain what is meant by the term ‘variable’.

---

[2]

(ii) Explain what is meant by ‘scope’ in relation to global and local variables.

---

[2]

7. A company employs a systems analyst to create a program for calculating wages. The data that he analyses has the following items:

- Hourly rate.
- Hours worked.
- Tax paid.
- National Insurance paid.

Various variables are used in the execution of the program.

Explain the term ‘scope’ in relation to variables within a program that calls several different procedures.

[6]

8. Programming languages consist of three basic programming constructs. For each construct, state its name and give a working example.

Construct 1: _____

Example: _____

Construct 2: _____

Example: _____

Construct 3: _____

Example: _____

---[6]

9(a). The code below uses a procedure:

```
name = "Sam"
addMessage(name)
print(name)

procedure addMessage(inText:byVal)
     inText = "Hello " + inText
endprocedure
```

Explain why this program outputs `Sam` rather than `Hello Sam`.

---

[2]

(b). Explain the advantages of writing an application using a modular approach.

---

[6]

10. A group of A-level students are working together to program a computer game.

In the game, the player controls a character who moves through a virtual world. The game starts with a load-up screen. The player can select which area to move to on an on-screen map, and then they control the movements of their character using a keyboard to solve puzzles on the screen.

The following pseudocode algorithm is for the sub-procedure of the game created `characterMovement`.

```
    procedure characterMovement(inputKey:byVal, characterx:byRef,
characterY:byRef)
        if inputKey == 37 then
            characterx = characterx + 1
        elseif inputKey == 38 then
            charactery = charactery + 1
        elseif inputKey == 39 then
            characterx = characterx - 1
        elseif inputKey == 40 then
            charactery = charactery - 1
        endif
    endprocedure
```

(i) Identify the **three** parameters in the procedure `characterMovement`.

1 ---

2 ---

3 ---

[3]

(ii) Describe the decision that is made in this procedure and how the decision affects the flow through the procedure.

---

[3]

(iii) Explain why `characterx` and `charactery` are passed `byRef` and not `byVal`.

[3]

11. A programmer is developing an ordering system for a fast food restaurant. When a member of staff inputs an order, it is added to a linked list for completion by the chefs.

The programmer is writing the program using an IDE.

Identify **three** features of an IDE that the programmer would use when writing the code and describe how the features benefit the programmer.

1 ----------

---

2 ----------

---

3 ----------

---

[6]

12. A function, `readMessage`:

- takes the file name as a parameter
- reads and returns the line of text

Complete the pseudocode algorithm for `readMessage`:

```
function ……(fileName)
    messageFile = openRead(……)
    message = messageFile.readLine()
    messageFile. ……
    return ……
endfunction
```

[4]

13(a). A procedure takes as input a number between 1 and 100. It calculates and outputs the square of each number starting from 1, to the number input. The square of a number is the result of multiplying a number by itself.

```
procedure squares()
    do
        number = int(input("Enter a number between 1 and 100"))
    until number >= 1 AND number <= 100

    for x = 1 to number
        print(x * x)
    next x
endprocedure
```

The procedure uses one programming construct twice.

State whether the construct that is used twice, is iteration or branching.

---

[1]

(b). State why the algorithm is a procedure and not a function.

---

[1]

14. A user enters whole numbers into a computer program. Each number entered is placed onto a stack. The stack is created using an array with a maximum of 20 elements.

Part of the array, `numStack`, is shown when one number has been input.

top	1

index	stackItem
9	
8	
7	
6	
5	
4	
3	
2	
1	
0	20

The pointer, `top`, points to the next free space in the stack.

A function, `addItem`, takes a number as a parameter and adds the number to the stack. The function returns `true` if this was successful, and `false` if the stack is already full.

(i) Give **one** reason why a function is used instead of a procedure in this scenario.

---

[1]

(ii) The parameter can be passed by value or by reference.

Describe what is meant by passing a parameter by value and by reference.

By value ---

By reference ----------

---

[4]

(iii) The function `addItem` is written but is incomplete.

Complete the function, `addItem`.

```
function addItem (number)
      if top == ..................................... then
           return false
      else
           numStack [.....................] = .....................
           top = ............................. + 1
           ..................................................
      endif
endfunction
```

[5]

(iv) The procedure, `calculate`, takes each item in turn from the stack. It alternately adds then subtracts the numbers until there are none left.

For example, if `numStack` contains:

2
6
5
12

It would perform 2 + 6 – 5 + 12 and output 15.

```
01   procedure calculate()
02        total = 0
03        add = true
04        if top == 0 then
05             print("Stack empty")
06        else
07             total = numStack[top - 1]
08             top = top ? 1
09             while top != 0
10                  if add == true then
11                       total = total + numStack[top - 1]
12                       add = false
13                  else
14                       total = total ? numStack[top - 1]
15                       add = true
16                  endif
17                  top = top - 1
18             endwhile
19             print(total)
20        endif
21   endprocedure
```

Complete the trace table for the procedure `calculate`. The current array and pointer values when the procedure is called are on the first line of the trace table.

top	numstak						total	add	output
	0	1	2	3	4	5			
5	20	2	6	12	8				

[6]

15(a). A games company has developed a game called Kidz Arrowz. The players throw an arrow at a target board and are awarded different points depending on which circle the arrow lands. Fig. 1 shows the board.

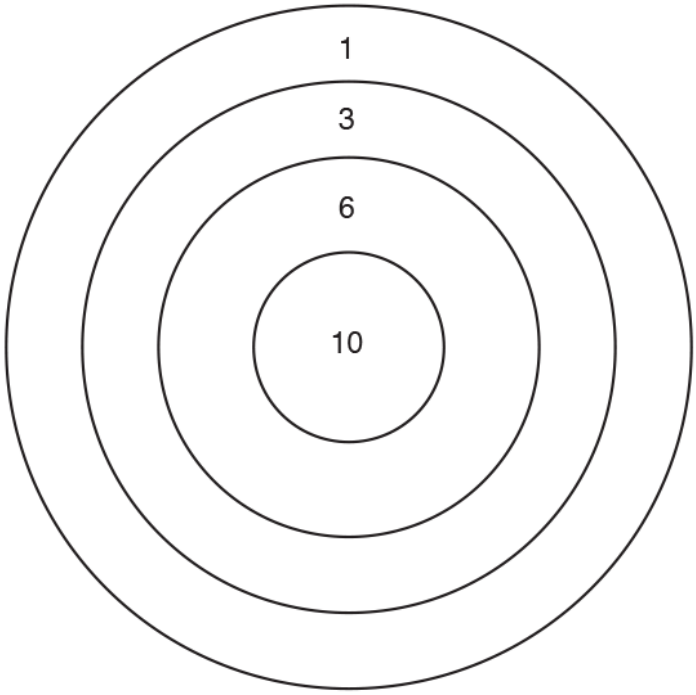


Fig. 1

A computer program is required to keep track of the scores for each competition. The user will enter the number of players, and the name of each player, in that competition to a maximum of 10.

The program is decomposed into multiple sub-programs, that each perform a specific task.

The array, `scores`, is declared as a global array of type record:

```
global array scores[10] of player
```

Explain why the array `scores` has been declared as global instead of local.

---

[2]

(b). The programmer uses an Integrated Development Environment (IDE) to develop the program.

Describe how the IDE could be used to create the Kidz Arrowz program.

---

[3]

16(a) A recursive function, `generate`, is shown.

.

```
function generate(num1:byval)
     if num1 > 10 then
          return 10
     else
          return num1 + (generate(num1 + 1) DIV 2)
     endif
endfunction
```

Trace the algorithm to show the value returned when `generate(7)` is called. Show each step of your working.

---

[6]

(b). The parameter, `num1`, is passed by value.

Explain why the parameter was passed by value instead of by reference.

[2]

(c). * Parameters can be used to reduce the use of global variables.

Compare the use of parameters to global variables in recursive functions.

[9]

17(a). Identify and describe **three** features commonly found in an IDE that will help programmers to find any bugs in their code.

1

2

3

[6]

(b). Describe what is meant by the term *IDE (Integrated Development Environment)*.

[2]

18. Describe **one** difference between a global and a local variable.

[2]

19. Nobugs is a software development company that produces enterprise-wide management software for large companies. Its software products are built up from many program functions.

The managers of Nobugs enforce standard rules on their programmers about how program functions should be written.

The following are some of the rules that they insist upon:

- no function may be longer than a single page of code
- variable identifiers must conform to a standard convention
- each function must have a single entry point
- variables must not be set up outside the scope of a function
- hardware-specific code must be avoided
- embedded documentation must be adequate.

Describe what is meant by a function.

---

[2]

20. Dexter is leading a programming team who are creating a computer program that will simulate an accident and emergency room to train hospital staff.

Dexter's team is using an integrated development environment (IDE).

Describe how the programmers could make use of the following IDE tools:

Breakpoints _____

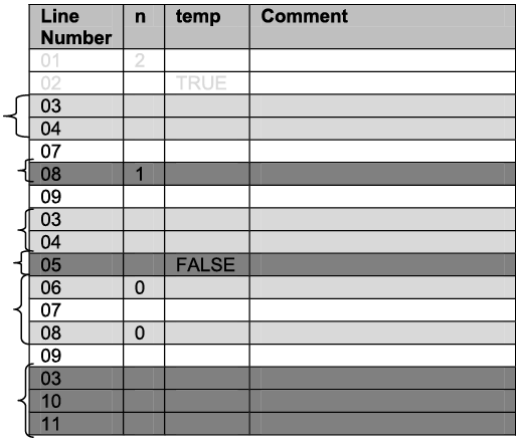
Stepping _____

---

[4]

END OF QUESTION PAPER

Question			Answer/Indicative content	Marks	Guidance
1		i	• * has higher precedence than +	1	Allow follows rules of BODMAS / BIDMAS Examiner's Comments Well answered with reference to BODMAS/BIDMAS.
		ii	• Make the formula / line easier to understand / read… • … as contents of bracket stands for total interest	2	Do not allow code / program Examiner's Comments The main reason given was that it made the code easier to read rather than the line. Only a few made the connection to the fact that the contents of the bracket calculated total interest.
		iii	• So the formula is calculated correctly • The * is done before the \	2	The 2nd bullet is worth two marks if that is the only point made
			Total	5	
2		i	• (A description of) an item of data • That is passed to a subroutine (when it is called)… • …is used as a variable within the subroutine	2	Examiner's Comments The wording used by some candidates did not make it clear that the parameter was passed to the subroutine. Words such as 'fed', 'allocated', 'put', etc. were used. Few mentioned that the parameter was used as a variable within the subroutine.
		ii	- Message: String - PhoneNumber: String *One mark for correct names of parameters + one mark each for the data types.*	3	*Parameter names cao* *If parameters are misspelt / wrong case, data type marks can be awarded* Examiner's Comments Most candidates picked up full marks, but again the incorrect use of case caused a problem along with the continued idea that any digit string that starts with a zero can be held in an Integer.
			Total	5	

Question			Answer/Indicative content	Marks	Guidance
3	a		• (WHILE) loop (on line 03) will repeat lines 04 to 08 or 03 to 09 • as long as n > 0 / until n is not > 0	2	**Examiner's Comments** Many good answers but a large minority tried to explain the principle of a while loop not how it was used in this algorithm.
	b		• (line 02) Temp is set to TRUE • Condition in line 3 is false (so it skips the loop) • in line 10, it returns TEMP (which was set to TRUE)	3	Do not accept code line i.e. Temp=TRUE **Examiner's Comments** Many candidates lost the first point by simply copying the Temp = true line from the given algorithm.
	c		 *One mark for each bulleted group of rows*	6	If the value of n & temp does not change allow copy down. Ignore comments Start marking from top until wrong, then start marking from bottom until wrong **Examiner's Comments** A large number of candidates scored 1 mark almost by accident as they simply wrote the line numbers in ascending order. There were, however, many responses getting 5 or 6 marks here.
			Total	11	

Question			Answer/Indicative content	Marks	Guidance
4		i	High-level language / 3GL / imperative language Gives a series of instructions in a (logical) order / line by line / what to do and how to do it	2	Examiner's Comments A mixed bag of answers for this question, a good example of candidates not reading the question. About half gave a perfect answer and the other half said something about using procedures and functions or that it used sequence, selection and iteration which was not what was required.
		ii	Declare (result) as a local variable in each procedure Accessible within one procedure (at a time) / the scope of the variable is for one procedure at a time / only exists as long as the procedure is running	2	Examiner's Comments Most candidates were able to give a complete answer to this question and it was good to see candidates talking about scope of the variables.
		iii	Parameters passed by value or by reference By value, local copy of data is used then discarded… …so value of (original) data is unchanged By reference, location of data is used… …so changes may be made to value of data	5	Examiner's Comments Some candidates knew this and were able to reel off the answers easily, some managed to get half way and gave vague answers on the detail and some missed the point entirely. This question was designed to cover a range of grades and this was demonstrated in the range of answers given.
			Total	9	

Question			Answer/Indicative content	Marks	Guidance
5		i	• Defined within one module… • … accessible only in that module / Any mention of scope • Can be used as parameters • Data is lost at end of module • Same variable name can be used in other modules without overwriting values/causing errors • Can overwrite global variables (with the same name)	4	For module allow procedure / function / sub routine / block of code Examiner's Comments Well answered by most candidates.
		ii	• Defined at start of program • Exists throughout program / in all modules • Allows data to be shared by modules	2	Examiner's Comments Nearly all candidates were able to get at least one mark on this.
			Total	6	
6		i	• Identifier/name of a … • Memory location used to store data	2	Examiner's Comments Generally well answered.
		ii	• A range of statements/procedure/function/method that a variableis valid for • A local variable takes precedence over a global variable of thesame name/allow the same identifier to be used for different purposes without conflict	2	Accept block of code Examiner's Comments Most gained a mark, although many vague references to code were given which were rescued by definitions of global and local variables.
			Total	4	

Question			Answer/Indicative content	Marks	Guidance
7			• Global variables are (usually) defined at the start of a program • Global variables can be seen / used everywhere in the program • Local variables can only be seen / used in a procedure / function / sub routine in which they are declared • Local variables cease to exist once the procedure / function / sub routine they are in is finished • Local variables with the same name as global variables… • …will overwrite / take precedence over the values in the global variable • Local variables within two different procedures will not interfere with one another	6	For 4th bullet accept construct Examiner's Comments A large majority of very good answers to this question with candidates writing confidently.
			Total	6	
8			• Selection / Branching (1) (AO1.1) • Working selection example (1) (AO1.2) e.g. ```if a>b then c=b+42 endif``` • Iteration (1) (AO1.1) • Working iteration example (1) (AO1.2) e.g. ```for count=1 to 10 print(count) next count``` • Sequence (1) (AO1.1) • Working Sequence example (1) (AO1.2) e.g. ```qty = input() total = qty * price```	6	Max 6 marks Do not penalise pseudocode if it is does not conform to the specification pseudocode guidelines. Examiner's Comments The programming constructs of sequence, iteration and branching are specifically identified within the specification. Many candidates were unaware of these named constructs. Of those who were, many then failed to give a working example as required by the question, but went on to describe rather than exemplify. Reponses such as looping were too vague as candidates are expected to know the correct technical vocabulary at AS Level.
			Total	6	

Question			Answer/Indicative content	Marks	Guidance
9	a		Parameter / name is passed by value... ...rather than by reference / by value does not change the original variable value		**Examiner's Comments** Few candidates understood the concept of passing by reference and passing by value which is in the specification. Greater programming experience using both methods would pay dividends for many candidates.
	b		• Work is easier to divide between a team • each team member just needs to know what values go into their subroutine and the expected functionality • Saves time as work takes place in parallel • each team member can work on their area of expertise. • Breaks problems into smaller areas. • Easier to test/ debug/ read • each subroutine can be tested before integration. • Code can be reused in the project/ future projects	6	Maximum 6 marks **Examiner's Comments** Many candidates achieved some credit for this answer, but few could identify and expand upon a number of different points regarding the advantages of using a modular approach. This highlighted a lack of exam technique whereby candidates did not think about the number of separate points that they were expected to give to achieve the full six marks.
			Total	8	

Question			Answer/Indicative content	Marks	Guidance
10		i	1 mark for each parameter – case sensitive • `inputKey` [1] • `characterx` [1] • `charactery` [1]	3 AO1.2 (3)	No spaces allowed in parameter names **Examiner's Comment:** Most candidates had a good understanding of what parameters were and could hence answer the question well.
		ii	1 mark per bullet • Decision is based on the value of `inputKey`… [1] • …and the values of `characterx` or `charactery` are changed [1] • Description of a condition and what it will do e.g. If the input key equals value 37 [1], then the x coordinate is increased [1]	3 AO1.2 (1) AO2.1 (2)	**Examiner's Comment:** Many candidates answered vaguely and could not describe in detail the condition that was implemented.
		iii	1 mark per bullet to max 3 • ByRef changes the value in the variable passed (`characterx` and `charactery`) [1] • ByRef passes the address/location [1] • ByVal only a copy of the data is passed [1] • ByVal the change would be lost when the procedure ended [1]	3 AO2.1	**Examiner's Comment:** ByRef and ByVal continue to be an area that candidates struggle with. Those with experience of languages that implement this tended to do better.
			Total	9	

Question			Answer/Indicative content	Marks	Guidance
11			1 mark for feature, 1 for benefit. Max 2 per feature. e.g. • Auto-complete • Can view identifiers / avoid spelling mistakes • Colour coding text / syntax highlighting • Can identify features quickly / use to check code is correct • Stepping • Run one line at a time and check result • Breakpoints • Stop the code at a set point to check value of variable(s) • Variable watch / watch window • Check values of variables and how they change during the execution • Error diagnostics • Locate and report errors / give detail on errors	6 AO1.1 (3) AO1.2 (3)	Question states when writing the code, therefore use of compiler / producing .exe etc. are not awarded marks Accept any suitable features e.g. traces, crash dump, stack contents, cross-references, line numbers, auto-indent Examiner's Comment: Most candidates achieved some credit for factual recall. However, weaker candidates often answered debugger rather than explaining the specific features of the debugger which would have been creditworthy.
			Total	6	

Question			Answer/Indicative content	Marks	Guidance
12			1 mark each `function readMessage(fileName)` `  messageFile = openRead(fileName)` `  message = messageFile.readLine()` `  messageFile.close()` `  return message` `endfunction`	4 AO2.1 (4)	We are not testing pseudocode knowledge – answers that work but do not match the pseudo code given should still be credited full marks. readMessage and fileName and message are case sensitive Examiner's Comment: Many candidates struggled to produce good answers which could have been calculated and did not require factual recall.
			Total	4	
13	a		Iteration [1]	1 AO2.1 (1)	Examiner's Comment: Well answered by most candidates.
	b		It does not return a value [1]	1 AO2.1 (1)	Examiner's Comment: A number of candidates clearly did not appreciate how functions differ from procedures.
			Total	2	

Question			Answer/Indicative content	Marks	Guidance
14		i	A procedure does not return a value / a function has to return a value	1 AO1.2 (1)	Examiner's Comments Many candidates answered well and understood the difference between functions and procedures, knowing that functions have to return a value.
		ii	1 mark per bullet, max 2 for by value, max 2 for by reference by value: • A local copy of the data is used • Data is discarded when the subprogram exits • Does not override/change the original data by reference: • Memory location of data is sent • Changes are made to the original data • Changes remain after the subprogram exits	4 AO1.2 (4)	Examiner's Comments Parameter passing by value and by reference continue to prove problematic to candidates, with many having a poor grasp of the concept. Those candidates who have used a variety of programming languages including those that allow for parameter passing by reference often had the practical experience to draw upon.
		iii	1 mark for each completed space to max 5 `function addItem (number)` `  if top = 20 then` `  return false` `  else` `  numStack[top] = number` `  top = top + 1` `  return true` `  endif` `endfunction`	5 AO2.2 (2) AO3.2 (3)	Accept `numStack.length()` instead of `20` Examiner's Comments Candidates are best prepared for this paper by having had practical experience of implementing data structures such as stacks and queues. A number of candidates did not read the whole stem of the question and assumed that the total number of elements was 10 instead of 20. The concept that the stack pointer points to the next space to be used in the stack was poorly understood. Those candidates with practical experience and the ability to read and interpret code did answer well.

<table>
<tr><th colspan="3">Question</th><th>Answer/Indicative content</th><th>Marks</th><th>Guidance</th></tr>
<tr><td></td><td></td><td>iv</td><td>1 mark for each bullet to max 6

• Initialising <code>total</code> to 0 and <code>add</code> to true
• <code>top</code> = 4 and <code>total</code> = 8
• <code>total</code> = 20 and <code>add</code> = false
• <code>top</code> = 3, <code>total</code> = 14, <code>add</code> = true
• <code>top</code> 2, 1. <code>Total</code> 16, -4, <code>add</code> false, true
• Output = -4

<table><tr><th></th><th colspan="6">numStack</th><th></th><th></th><th></th></tr><tr><th>top</th><th>0</th><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th><th>total</th><th>add</th><th>Output</th></tr><tr><td>5</td><td>20</td><td>2</td><td>6</td><td>12</td><td>8</td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>0</td><td>true</td><td></td></tr><tr><td>4</td><td></td><td></td><td></td><td></td><td></td><td></td><td>8</td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>20</td><td>false</td><td></td></tr><tr><td>3</td><td></td><td></td><td></td><td></td><td></td><td></td><td>14</td><td>true</td><td></td></tr><tr><td>2</td><td></td><td></td><td></td><td></td><td></td><td></td><td>16</td><td>false</td><td></td></tr><tr><td>1</td><td></td><td></td><td></td><td></td><td></td><td></td><td>-4</td><td>true</td><td></td></tr><tr><td>0</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>-4</td></tr></table></td><td>6
AO1.2
(3)
AO2.2
(3)</td><td><u>Examiner's Comments</u>

Tracing code execution is an area that continues to prove challenging to candidates. Candidates need to have experience of completing dry-runs of code and setting out a trace table in a logical manner. Where candidates did perform well the initialisation of the variables total and add before the main body of the loop was entered was often omitted.</td></tr>
<tr><td></td><td></td><td></td><td>Total</td><td>16</td><td></td></tr>
<tr><td>15</td><td>a</td><td></td><td>1 mark per bullet to max 2

• It does not need to be passed between subroutines
• It can be accessed/updated at any point/place in the program
• It allows it to be updated as a running total</td><td>2
AO2.1
(1)
AO2.2
(1)</td><td><u>Examiner's Comments</u>

Most candidates successfully identified that a global variable would have scope throughout the program and would therefore be available within each subroutine. Fewer could expand on this in context.</td></tr>
<tr><td></td><td>b</td><td></td><td>1 mark per bullet to max 3
e.g.

• Provides a text editor / allows the code to be written
• Provides debugging tools / allows the code to be tested
• Provides a translator/compiler/interpreter / provides a run-time environment / allows the code to be run
• Description of key feature e.g. colour coding keywords, autocomplete, breakpoints etc.</td><td>3
AO1.2
(3)</td><td><u>Examiner's Comments</u>

It was clear that nearly all candidates had experience of using an IDE and that they could successfully identify a number of features that an IDE provides.</td></tr>
<tr><td></td><td></td><td></td><td>Total</td><td>5</td><td></td></tr>
</table>

Question			Answer/Indicative content	Marks	Guidance
16	a		1 mark per bullet for working to max 6 • generate(7) return 7 + (generate(8) DIV 2) • generate(8) return 8 + (generate(9) DIV 2) • generate(9) return 9 + (generate(10) DIV 2) • generate(10) return 10 + (generate(11) DIV 2) • generate(11) return 10 • Rewinding: return 10 + (10 DIV 2) = 10 + 5 = 15 • return 9 + (15 DIV 2) = 9 + 7 = 16 return 8 + (16 DIV 2) = 8 + 8 = 16 • return 7 + (16 DIV 2) = 7 + 8 = 15	6 AO1.2 (1) AO2.2 (5)	Examiner's Comments A significant number of candidates struggled to clearly present a program trace for a recursive algorithm. Those who used diagrammatic or clearly indented structures showing the recursive levels and the return values faired best. Candidates need to be encouraged to work on producing a logical layout for a recursive function trace.
	b		• If the value is sent by value, num1 will not be overridden / it is a copy of the parameter that is used (1) and this will produce the correct output (1) • if the parameter had been passed by reference it would not produce the correct result (1) as num1 would be overridden / because it is a pointer to the address of the variable (1)	2 AO2.1 (1) AO2.2 (1)	Examiner's Comments Many candidates either defined passing by value or passing by reference and did not answer the question. Few could demonstrate a deeper understanding of the implications of the method of parameter passing chosen within the context of a recursive function.

Question			Answer/Indicative content	Marks	Guidance
	c		**Mark Band 3 – High level** **(7-9 marks)** The candidate demonstrates a thorough knowledge and understanding of parameters and global variables; the material is generally accurate and detailed. The candidate is able to apply their knowledge and understanding directly and consistently to the context provided. Evidence/examples will be explicitly relevant to the explanation. *There is a well-developed line of reasoning which is clear and logically structured. The information presented is relevant and substantiated.* **Mark Band 2 – Mid level** **(4-6 marks)** The candidate demonstrates reasonable knowledge and understanding of parameters and global variables; the material is generally accurate but at times underdeveloped. The candidate is able to apply their knowledge and understanding directly to the context provided although one or two opportunities are missed. Evidence/examples are for the most part implicitly relevant to the explanation. The candidate provides a reasonable discussion, the majority of which is focused. Evaluative comments are, for the most part appropriate, although one or two opportunities for development are missed. *There is a line of reasoning presented with some structure. The information presented is in the most part relevant and supported by some evidence.* **Mark Band 1 – Low Level** **(1-3 marks)** The candidate demonstrates a basic knowledge of parameters and global variables with limited understanding shown; the material is basic and contains some inaccuracies. The candidates makes a limited attempt to apply acquired knowledge and understanding to the context provided.	9 AO1.1 (2) AO1.2 (2) AO2.1 (2) AO3.3 (3)	**AO1: Knowledge and Understanding** **Indicative content** • Parameter allows a value to be sent to a sub-program • Global variables can be accessed throughout the scope of the program • Local variables can only be accessed within the scope of the sub-program it's defined within ¡V a parameter becomes a local variable in the function **AO2: Application** • If global, equivalent of by reference -value would be over-ridden • Global variable takes more memory than a local variable/parameter • In recursion, each call produces a new local variable for num1 **AO3: Evaluation** Candidates will need to evaluate the benefits and drawbacks of each algorithm • Global would require altering the algorithm as the value would be over-ridden on each call • Global would mean that memory space is kept throughout the running of the program, not just the sub-program • Parameter enables memory to be reallocated • Many more memory spaces needed for parameter in recursion, 1 for each call <u>Examiner's Comments</u> Most candidates produced responses limited to the scope of global variables being accessible throughout the program or a discussion of the different methods of parameter passing available. Few made any references to either recursive functions or to the implications to memory usage of using parameters instead of global variables.

Question			Answer/Indicative content	Marks	Guidance
			The candidate provides a limited discussion which is narrow in focus. Judgements if made are weak and unsubstantiated. *The information is basic and comunicated in an unstructured way. The information is supported by limited evidence and the relationship to the evidence may not be clear.* 0 marks No attempt to answer the question or response is not worthy of credit.		
			Total	17	
17	a		• Debugging tools allow inspection of variable values (1 – AO 1.1) this can allow run-time detection of errors (1 – AO 1.2). • Code can be examined as it is running (1 – AO 1.1) which allows logical errors to be pinpointed (1 – AO 1.2). • IDE debugging can produce a crash dump (1 – AO 1.1), which shows the state of variables at the point where an error occurs (1 – AO 1.2). • It can display stack contents (1 – AO 1.1) which show the sequencing through procedures / modules (1 – AO 1.2). • It can step through code (1 – AO 1.1), which allows the programmer to watch the effects each line of code (1 – AO 1.2). • The insertion of a break-point (1 – AO 1.1) allows the program to be stopped at a predetermined point in order to inspect its state (1 – AO 1.2).	6	1 mark (AO 1.1) for each correct identification up to a maximum of three identifications plus up to a further 1 mark (AO 1.2) for each of three valid descriptions.
	b		• A (single) program (1) used for developing programs (1) made from a number of components (1).	2	Up to 2 marks for a valid description.
			Total	8	

Question			Answer/Indicative content	Marks	Guidance
18			• Global variable is visible throughout a program / may be accessed from more than one part of the program (1), local variable is visible only in module / construct where it is created / declared (1).	2	Up to 2 marks for a valid description.
			Total	2	
19			• A function is a named section of program (1) that performs a specific task (1). • It returns a value (1), it is often called inline (1).	2	Up to 2 marks for a valid description.
			Total	2	
20			1 mark per bullet, max 2 for each tools Breakpoints • Use to test the program works up to/at specific points • Check variable contents at specific points • Can set a point where the program stops running Stepping • Can set the program to run line by line • Slow down/watch execution • Find the point where an error occurs	4	
			Total	4	