

Example Candidate Responses – Paper 4

Cambridge International AS & A Level Computer Science 9618

For examination from 2021



Contents

Introduction.....	4
Question 1	6
Example Candidate Response – high	6
Example Candidate Response – middle	10
Example Candidate Response – low	14
Question 2	17
Example Candidate Response – high	17
Example Candidate Response – middle	21
Example Candidate Response – low	23
Question 3	26
Example Candidate Response – high	26
Example Candidate Response – middle	31
Example Candidate Response – low	36

Introduction

The main aim of this booklet is to exemplify standards for those teaching Cambridge International AS & A Level Computer Science 9618, and to show how different levels of candidates' performance (high, middle and low) relate to the subject's curriculum and assessment objectives.

In this booklet candidate responses have been chosen from the June 2021 exam series to exemplify a range of answers.

For each question, the response is annotated with a clear explanation of where and why marks were awarded or omitted. This is followed by examiner comments on how the answer could have been improved. In this way, it is possible for you to understand what candidates have done to gain their marks and what they could do to improve their answers. There is also a list of common mistakes candidates made in their answers for each question.

This document provides illustrative examples of candidate work with examiner commentary. These help teachers to assess the standard required to achieve marks beyond the guidance of the mark scheme. Therefore, in some circumstances, such as where exact answers are required, there will not be much comment.

The questions, mark schemes and inserts used here are available to download from the School Support Hub. These files are:

9618 June 2021 Question Paper 42

9618 June 2021 Mark Scheme 42

Past exam resources and other teaching and learning resources are available on the School Support Hub:

www.cambridgeinternational.org/support

How to use this booklet

This booklet goes through the paper one question at a time, showing you the high-, middle- and low-level response for each question. The candidate answers are set in a table. In the left-hand column are the candidate answers, and in the right-hand column are the Examiner comments.

Example Candidate Response – high	Examiner comments
Question 3 Part 3(a) 1 2 <pre>Public Class TreasureChest Private question As String Private answer As Integer Private points As Integer Sub New(ByVal ques As String, ByVal ans As Integer, ByVal pts As Integer) Me.answer=ans Me.question=ques Me.points=pts End Sub End Class</pre> Answers are by real candidates in exam conditions. These show you the types of answers for each level. Discuss and analyse the answers with your learners in the classroom to improve their skills.	1 The candidate declares the class <code>TreasureChest</code> and includes <code>End Class</code>, so they are awarded marking point 1. 2 The candidate declares <code>question</code> as a string and within the class, so they are awarded marking point 2. They declare <code>'answer'</code> as an integer and within the class so are awarded marking point 3. They declare <code>points</code> as an integer and Examiner comments are alongside the answers. These explain where and why marks were awarded. This helps you to interpret the standard of Cambridge exams so you can help your learners to refine their exam technique.

How the candidate could have improved their answer

- The candidate's answers met all the given requirements. They could have included comments in their code to support their logic and any inbuilt functions (such as the file reading) they used.
- (c)(iv) A value was read incorrectly from the user but was not validated. The candidate could have validated the input because it needed to be either 1, 2, 3, 4 or 5 so any other number or character would have been invalid and would have caused the program to report a runtime error.

This section explains how the candidate could have improved each answer. This helps you to interpret the standard of Cambridge exams and helps your learners to refine their exam technique.

Common mistakes candidates made in this question

- (a) Some candidates did not declare the attributes as private. This could be done in different ways depending on the programming language. When writing in Python, candidates needed to make clear the data type of the attributes. This could be done using comments or by another method, provided this was clear.
- (b) Some candidates did not make the data type clear in their answer, when an array needed to be declared of a given data type. In Python, a comment could have been used to identify the number of elements and the data type, or empty elements could have been declared of the correct number and type.

Often candidates were not awarded marks because they misread or misinterpreted the questions.

Lists the common mistakes candidates made in answering each question. This will help your learners to avoid these mistakes and give them the best chance of achieving the available marks.

Question 1

Example Candidate Response – high	Examiner comments
Question 1 Part 1(a) 1 <pre>Structure node Dim data As Integer Dim nextNode As Integer End Structure</pre> Part 1(b) 2 <pre>Dim linkedList(9) As node linkedList(0).data=1 linkedList(0).nextNode=1 linkedList(1).data=5 linkedList(1).nextNode=4 linkedList(2).data=6 linkedList(2).nextNode=7 linkedList(3).data=7 linkedList(3).nextNode=-1 linkedList(4).data=2 linkedList(4).nextNode=2 linkedList(5).data=0 linkedList(5).nextNode=6 linkedList(6).data=0 linkedList(6).nextNode=8 linkedList(7).data=56 linkedList(7).nextNode=3 linkedList(8).data=0 linkedList(8).nextNode=9</pre>	1 A Structure is a suitable Visual Basic feature for declaring a record type. The candidate is awarded the first mark for declaring the structure with the correct name. They declare the fields with suitable data types within the record and are awarded the second marking point. Mark for (a) = 2 out of 2 2 The candidate declares an array of the correct structure type from their previous answer, and is awarded marking point 1. They assign the correct data to the array and they use the <code>data</code> and <code>nextNode</code> fields appropriately to be awarded marking points 2 and 3. The candidate does not declare the two pointers <code>emptyList</code> and <code>startPointer</code> and does not initialise these to the required values. Mark for (b) = 3 out of 4

Example Candidate Response – high, continued	Examiner comments
<pre>linkedList(9).data=0 linkedList(9).nextNode=-1 linkedList(9).data=0 linkedList(9).nextNode=-1</pre> Part 1(c)(i) 3 <pre>Sub outputNode(ByVal startPointer As Integer , ByVal list() As node) Dim currentNodePointer As Integer currentNodePointer=startPointer While (currentNodePointer<>-1) Console.WriteLine(list(currentNodePointer).data) 4 currentNodePointer=list(currentNodePointer).nextNode End While End Sub</pre> Part 1(c)(ii)  5	3 The candidate declares the procedure with the correct name, and is awarded marking point 1. The procedure takes the pointer and the list as parameters and the candidate gives the appropriate data types of each so is awarded marking point 2. 4 The program loops until the pointer at the node being accessed is -1 (null) so the candidate is awarded marking point 3. Within the loop, the program outputs the data at the current node using the appropriate field access. This pointer is updated appropriately so the candidate is awarded marking point 4. The pointers are updated by accessing the pointer to the next node and updating the loop control so the candidate is awarded marking point 5. The program accesses the fields of the appropriate record data throughout the answer and the candidate is awarded marking point 6. Mark for (c)(i) = 6 out of 6 5 The candidate's screenshot shows the data output in the correct order and the node's pointers are followed. Mark for (c)(ii) = 1 out of 1

Example Candidate Response – high, continued	Examiner comments
Part 1(d)(i) <pre>Function addNode(ByVal list() As node , byref emptyList As Integer,byval startPointer As Integer) As Boolean Dim data As Integer Dim previousNode As Integer=startPointer While (list(previousNode).nextNode<>-1) 7 previousNode=list(previousNode).nextNode End While Console.WriteLine("Enter the data to be inserted in list:") 8 data=Console.ReadLine() Dim currentNode As Integer currentNode=emptyList If(emptyList<>-1) Then emptyList=list(emptyList).nextNode End If If(currentNode<>-1) Then list(previousNode).nextNode=currentNode 10 End If list(currentNode).data=data 11 list(currentNode).nextNode=-1 Return True Else Return False 12 End If End Function</pre>	6 The function is declared with the correct name and both pointers are taken as parameters, with the correct data type for each. The answer gives the return data type as Boolean which is correct so the candidate is awarded marking point 1. 7 The linked list is traversed to find the last node in the list. 8 A value is taken as input from the user and stored appropriately, so the candidate is awarded marking point 2. 9 The candidate uses an appropriate check against <code>emptyList</code> to check if the list is full. The value of <code>emptyList</code> is stored in <code>currentNode</code> so they are awarded marking point 3. 10 The previous last node in the list is updated to point to the new node (the first node in the empty list) so the candidate is awarded marking point 7. 11 The data input is stored in the first node in the empty list so the candidate is awarded marking point 5. 12 False is returned when the list is full following through from marking point 3, so the candidate is awarded marking point 4. Mark for (d)(i) = 7 out of 7

Example Candidate Response – high, continued

Examiner comments

Part 1(d)(ii)

```
Dim emptyList As Integer=5
Dim startPointer As Integer=0
```

```
Console.WriteLine("List before entering data:")
outputNode(startPointer,linkedList)
Console.WriteLine("")
```

```
If(addNode(linkedList,emptyList,startPointer)=True) Then
    Console.WriteLine("")
    console.WriteLine("Data is inserted successfully.")
```

```
Else
    Console.WriteLine("")
    Console.WriteLine("There is no empty space in list.")
End If
```

```
Console.WriteLine("")
Console.WriteLine("List after entering data:")
outputNode(startPointer,linkedList)
```

Part 1(d)(iii)

C:\Users\lglab\Desktop\EG147_9618_42_4011\question 1\q

List before entering data:

```
1
5
2
6
56
7
```

Enter the data to be inserted in list:

5

Data is inserted successfully.

List after entering data:

```
1
5
2
6
56
7
5
```

Press any key to continue . . .

13 The function `addNode` is called with the appropriate parameters that follows on from the previous answer. The return value is used in the comparison and the candidate is awarded marking point 1.

14 The return value is used in the selection statement to output an appropriate message if the value is entered and if it is not entered, so the candidate is awarded marking point 2.

15 The procedure `outputNode` is called before and after the `addNode` function call. Both calls have the appropriate parameters following on from the previous answers so the candidate is awarded marking point 3.

Mark for (d)(ii) = 3 out of 3

16 The candidate's screenshot shows the initial data in the list. A number is taken as input and an appropriate message is output. The subsequent list is then output. The candidate also uses appropriate messages to identify the 'before insertion' and 'after insertion' and they are awarded a mark for this.

Mark for (d)(iii) = 1 out of 1

**Total mark awarded =
23 out of 24**

How the candidate could have improved their answer

- (b) The candidate correctly assigned the data to the linked list, but they did not declare the two pointers as instructed in the question. The candidate needed to declare `startPointer` as 0 and `emptyList` as 5.
- (c)(ii) The screenshot is present and can be magnified, but it would have been helpful for the candidate to have enlarged this in their document to make sure it was clear because it may not have scanned clearly at this size.
- (d)(i) The candidate could have included comments to make it clear which array elements were being accessed. This would have demonstrated their understanding and support a third party was following their code.
- (d)(ii) The candidate could have shown more of their solution to make it clear that this code was inserted within the main program.

Example Candidate Response – middle

Examiner comments

Question 1

Part 1(a)

{Copy and paste program code listing for question 1(a) here}

```
class node:
```

```
    data: int
```

```
    nextNode: int
```

```
def __init__(self, data: int, nextNode: int):
```

```
    self.data = data
```

```
    self.nextNode = nextNode
```

Part 1(b)

{Copy and paste program code listing for question 1(b) here}

```
startPointer = 0
```

```
emptyList = 5
```

```
linkedList = [node(1, 1), node(5, 4), node(6, 7), node(7, -1),
```

```
               node(2, 2), node(0, 6), node(0, 8), node(56, 3),
```

```
               node(0, 9), node(0, -1)]
```

Part 1(c)(i)

1 The candidate uses a class to represent a record type. This is an appropriate use and they include the constructor so they are awarded marking point 1.

2 The constructor initialises the two fields that are required in the record. The candidate uses an appropriate method to identify the data type of each field. A comment would also be appropriate and they are awarded marking point 2.

Mark for (a) = 2 out of 2

3 The candidate declares both pointers with the appropriate values, so they are awarded marking point 4.

4 The candidate declares a list with the correct identifier. They declare each element in the list as a node object which is appropriate for the data type node and are awarded marking point 1.

5 Within the list declaration, each item is of type node, so the candidate is awarded marking point 2. Each of these sends the data and the `nextNode` as parameters, which matches the constructor from (a) so they are awarded marking point 3.

Mark for (b) = 4 out of 4

Example Candidate Response – middle, continued	Examiner comments
Part 1(c)(i) 6 <pre>{Copy and paste program code listing for question 1(c)(i) here} def outputNodes(array: list(node), startPointer: int): for node in array: print(node.nextNode)</pre> 7	6 The candidate declares the procedure with the correct identifier so is awarded marking point 1. They take the list and pointer as parameters so they are awarded marking point 2.
Part 1(c)(ii) <pre>{Copy and paste the screenshot for question 1(c)(ii) here} 16 def outputNodes(array: list(node), startPointer: int): 17 for node in array: 18 print(node.nextNode) 19 20 outputNodes(linkedList, 0) 21 22 23</pre> 8	7 When accessing the data, the candidate uses the appropriate field access, e.g. <code>node.nextNode</code>, so they are awarded marking point 6.
Part 1(d)(i) <pre>{Copy and paste the program code listing for question 1(d)(i) here} def addNode(array: list(node), startPointer: int, emptyList: int): inputData = node(input()) array.append(inputData) startPointer += 1 emptyList += 1 if len(array) > 0: return False</pre> 9	8 This output follows on from (i) where the output is the data in the list, in the order it appears in the array, instead of following the pointers. Mark for (c)(ii) = 0 out of 1 9 The function has the appropriate identifier and takes the list, <code>startpointer</code> and <code>emptylist</code> as parameters. The data types of the parameters are given, which are suitable so the candidate is awarded marking point 1. Mark for (d)(i) = 1 out of 7

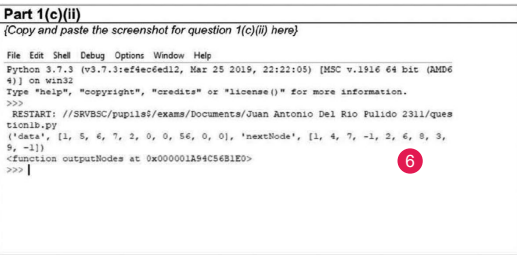
Example Candidate Response – middle, continued	Examiner comments
Part 1(d)(ii) <pre>{Copy and paste program code listing for question 1(d)(ii) here} outputNodes() outputNodes() addNode = addNode() if addNode == False: print("All the nodes has been added.") else: print("Not all nodes were added.") outputNodes() outputNodes()</pre> 10 11	10 <code>addnode</code> is not called with the appropriate values as parameters. 11 The candidate is awarded follow through marks for not using the parameters again when calling the <code>outputNodes</code> method before and after. They are awarded marking point 3. Mark for (d)(ii) = 1 out of 3

Example Candidate Response – middle, continued	Examiner comments
Part 1(d)(iii) (Copy and paste the screenshot for question 1(d)(iii) here) <pre>30 outputNodes() 31 outputNodes() 32 addNode = addNode() 33 if addNode == False: 34 print("All the nodes has been added.") 35 else: 36 print("Not all nodes were added.") 37 outputNodes() 38 outputNodes() 39 40 41 42</pre> PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE 5 PS C:\Users\cbt\Desktop\New folder> "All nodes has been added" 12 The output shows the number 5 and then a message stating 'All the nodes has been added', but this does not display all of the required output. Mark for (d)(iii) = 0 out of 1 Total mark awarded = 11 out of 24	

How the candidate could have improved their answer

- (c)(i) and (c)(ii) The candidate output all the data items in the array, but did this in the order that they appeared in the array, instead of following the pointers to output the list. The first node data output should have been that at the start pointer, then subsequent output should have been determined by the current node pointer until a null pointer (-1) was encountered. This also led to an incorrect output screenshot that showed all array elements, instead of just the data in the linked list.
- (d)(i) The candidate used the list as a data structure, instead of the linked list it was designed to be. The code appended the data item to the end of the list, instead of making use of the linked list and emptyList pointer. The code needed to access the pointers to follow the nodes from the first entry at startPoint until an empty node (-1) was found. The data then needed to be inserted into the node emptyList pointed to, and the relevant pointers updated.
- (d)(ii) The candidate called the function addNode(), but did not send the parameters to the function. In their response to the function declaration question, the list and pointers were identified as parameters. The candidate needed to pass these values in this function call.
- (d)(iii) The screenshot showed data being entered and an output message relating to the node being added, but it did not include the output of the list contents before and after the node was entered.

Example Candidate Response – low	Examiner comments
Question 1 Part 1(a) (Copy and paste program code listing for question 1(a) here) <pre>def node: type node: data : Integer nextNode : Integer</pre> Part 1(b) (Copy and paste program code listing for question 1(b) here) <pre>linkedList = ("data", [1,5,6,7,2,0,0,56,0,0], "nextNode", [1,4,7,-1,2,6,8,3,9,-1]) print(linkedList) alternative: linkListdata = [1,5,6,7,2,0,0,56,0,0] linkListnextNode = [1,4,7,-1,2,6,8,3,9,-1] print(linkListdata) print(linkListnextNode)</pre> Part 1(c)(i) (Copy and paste program code listing for question 1(c)(i) here) <pre>x = int(input("Search the data value By entering node value")) def outputNodes (linkedList, startPoint): if data[j] == nextNode[j]: print(data)</pre>	1 The user uses a command word 'type' which is a Python function to check the type of an object and does not declare a record structure. Mark for (a) = 0 out of 2 2 The candidate uses linkedList which is a dictionary structure instead of the required array. 3 The data is assigned to two separate arrays which does not meet the requirements of the question. Mark for (b) = 0 out of 4 4 The procedure outputNodes is declared so the candidate is awarded marking point 1. The procedure declaration includes the linked list and the start pointer so they are awarded marking point 2. 5 The code within the function compares data in two lists and outputs the contents of a list which does not output the data of the list by following the node's pointers. Mark for (c)(i) = 2 out of 6

Example Candidate Response – low, continued	Examiner comments
Part 1(c)(ii) {Copy and paste the screenshot for question 1(c)(ii) here} 	6 This is the output of the content stored in the dictionary <code>linkedList</code> rather than the output of the data of the linked list in order. Mark for (c)(ii) = 0 out of 1
Part 1(d)(i) {Copy and paste the program code listing for question 1(d)(i) here} <pre>y = int(input("Enter pointers to the linkedList")) linkListdata.append(y) print(linkListdata)</pre>	7 Data is read as input from the user and stored in the variable <code>y</code>, so the candidate is awarded the second marking point. Mark for (d)(i) = 1 out of 7
Part 1(d)(ii) {Copy and paste program code listing for question 1(d)(ii) here} <pre>call addNodes: call outputNodes() call outputNodes() call outputNodes(): call addNodes:</pre>	8 The candidate uses the 'call' functions inaccurately here. There is no evidence that <code>addNodes</code> is defined and the output call should be before and after this function call. Mark for (d)(ii) = 0 out of 2 Mark for (d)(iii) = 0 out of 1
Part 1(d)(iii) {Copy and paste the screenshot for question 1(d)(iii) here}	Total mark awarded = 3 out of 24

How the candidate could have improved their answer

- (a) The candidate needed to define a record structure. In Python this would most commonly be done using a class.
- (b) The candidate declared `linkedList`, but not as an array or list as per the requirements and instead they defined a dictionary inaccurately. They then declared the data into individual arrays instead of nodes of the record type. The candidate's answer needed to declare the two pointers.
- (c)(i) The content of the procedure performed a comparison of data in arrays. This needed to output the first node value at `startPointer` and then follow the pointers to output the contents of the linked list.
- (c)(ii) Following on from the output from (c)(i), the output was the contents of the dictionary declared and not the data of the list following the pointers.
- (d)(i) The solution to this question should have been written within a function. The candidate's answer appended the data input to the end of the list, but they needed to use the empty list pointer to determine the position of the new node. The response then output the contents of a list which was not a requirement of the question. The linked list was being treated as a list where the data was appended to the end, instead of using pointers to follow the nodes in order.
- (d)(ii) The candidate attempted to call the `outputNodes()` procedure, but this did not meet the requirements of the question. They needed to call this before and after `addNodes()` was called. The candidate did not define `addNodes()`, so was unable to call it.
- (d)(iii) The candidate did not give an answer to this question.

Common mistakes candidates made in this question

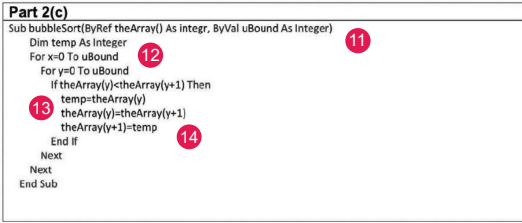
- (a) Many candidates declared a function instead of a record type (or equivalent) for the node. Some did not declare two arrays; one to store the data and one to store the next node pointers.
- (b) Some candidates missed the required declaration of the start pointer and empty list pointers and some incorrectly entered the data or node values.
- (c)(i) Some candidates output the contents of the array in the order it was in the array, instead of in the correct order, e.g. following the pointers from one node to the next.
- (d)(i) Some candidates appended the new data value to the end of the linked list array, instead of making use of the empty list pointer to identify the first free node. Some candidates did not update the last node's pointer to the new node's position.

Question 2

Example Candidate Response – high	Examiner comments
Question 2 Part 2(a) <pre>Dim arrayData(10) As Integer arrayData(0)=10 arrayData(1)=5 arrayData(2)=6 arrayData(3)=7 arrayData(4)=1 arrayData(5)=12 arrayData(6)=13 arrayData(7)=15 arrayData(8)=21 arrayData(9)=8</pre> Part 2(b)(i) <pre>Function linearSearch(ByVal searchitem As Integer, ByVal arrayData() As Integer) As Boolean Dim returnFlag As Boolean For i=0 To 9 If (arrayData(i)=searchitem) Then returnFlag= True End If Next If (returnFlag<>True) Then Return False Else Return True End If End Function</pre> Part 2(b)(ii)	1 The array is declared with the correct identifier with 10 data items (although this will create 11, it still meets the requirement of 10 elements) and of the appropriate data type. The candidate is awarded marking point 1. 2 The 10 array elements are assigned the correct values, so the candidate is awarded marking point 2. Mark for (a) = 2 out of 2 3 The function <code>linearSearch</code> has been defined with the correct identifier so the candidate is awarded marking point 1. The function has the search value as a parameter so they are awarded marking point 2. The array should be declared as global, but passing it as a parameter does not interfere with the functionality of the program and therefore is acceptable. 4 The candidate's loop runs from index 0 to 9 which covers all 10 array elements and they are awarded marking point 3. 5 Within the loop, the array data item is compared to the parameter so the candidate is awarded marking point 4. 6 The candidate sets a flag to signal whether the value being searched for is found or not. If this shows it has been found, when the loop ends 'True' is returned so the candidate is awarded marking point 5. Otherwise, 'False' is returned and the candidate is awarded marking point 6. Mark for (b)(i) = 6 out of 6

Example Candidate Response – high, continued	Examiner comments
<pre>Sub Main() Dim arrayData(10) As Integer arrayData(0)=10 arrayData(1)=5 arrayData(2)=6 arrayData(3)=7 arrayData(4)=1 arrayData(5)=12 arrayData(6)=13 arrayData(7)=15 arrayData(8)=21 arrayData(9)=8 Dim item As Integer Console.WriteLine("Please input an integer value.") item=Console.ReadLine() Dim result As Boolean result=linearSearch(item, arrayData) If (result=True) Then Console.WriteLine("Search value was found.") Else Console.WriteLine("Search value was not found.") End If End Sub</pre> Part 2(b)(iii)	7 Data is read in from the user, so the candidate is awarded marking point 1. 8 The function <code>linearSearch</code> is called. The value input is passed as a parameter, as well as the array which follows through from the answer to (b)(ii) so the candidate is awarded marking point 3. The value returned is stored in the variable <code>result</code>. 9 The value in <code>result</code> is compared to the boolean <code>True</code> so the candidate is awarded marking point 4. A suitable message is then output if the return value is 'True' and if it is 'False', so they are awarded marking point 5. Mark for (b)(ii) = 4 out of 4

Example Candidate Response – high, continued	Examiner comments
 10	10 The candidate gives two screenshots. Each shows the data value being input and the correct value output. The first shows the number 5 is found so the candidate is awarded marking point 1. The second shows the number 20 not found, so they are awarded the second marking point. The screenshots have a lot of excess space and are not clear without magnification. The candidate needs to make sure these are clearly legible. Mark for (b)(iii) = 2 out of 2

Example Candidate Response – high, continued	Examiner comments
 11 12 13 14	11 The code does not match the pseudocode because of the passing of parameters which were not in the given procedure, so the candidate is not awarded marking point 6. 12 The value of uBound is not clear because this is passed to bubbleSort as a parameter. Therefore, it is assumed that this is the upper bound of the array, 9. This is the correct missing statement for the outer loop and the candidate is awarded the first marking point. The second loop should stop at 1 less than the upper bound so they are not awarded this mark. 13 The candidate uses the correct logical comparison < in the selection and they are awarded marking point 3. 14 The missing value, theArray(y+1) is correct and temp is correctly given on the next line so the candidate is awarded marking point 5. Mark for (c) = 4 out of 6 Total mark awarded = 18 out of 20

How the candidate could have improved their answer

- (b)(i) The candidate needed to declare the array globally and it did not need to be passed as a parameter but this did not impact their answer and was accepted. They set a flag within the loop to indicate whether the search item was found. This would work functionally, but to improve efficiency the candidate could have returned 'True' at that point, and then 'False' outside the loop.
- (b)(ii) The candidate needed to validate the user's input to make sure it was acceptable. In this case, there were restrictions on what was acceptable and in this case the user's input needed to be an integer, otherwise the program would crash.
- (b)(iii) The candidate's screenshots were legible when magnified, but they could have cropped these and enlarged them to make sure they were still legible after scanning.
- (c) The candidate followed parts of the given pseudocode procedure but made further modifications to the functionality by taking values as parameters when the given procedure did not.

Example Candidate Response – middle

Question 2**Part 2(a)***(Copy and paste program code listing for question 2(a) here)*

```
arrayData = [10, 5, 6, 7, 1, 12, 13, 15, 21, 8]
print(arrayData)
```

1

Part 2(b)(i)*(Copy and paste program code listing for question 2(b)(i) here)*

```
def linearSearch(self, integer, arrayData):
    self.integer = 0
    self.arrayData = [10, 5, 6, 7, 1, 12, 13, 15, 21, 8]
    x = int(input("Enter a value for search: "))
    if integer in arrayData:
        return True
    else:
        return False
```

3

Part 2(b)(ii)*(Copy and paste program code listing for question 2(b)(ii) here)*

```
x = 0
arrayData = [10, 5, 6, 7, 1, 12, 13, 15, 21, 8]
print(arrayData)

x = int(input("Enter a value for search"))
print(arrayData[x], "the data was found")
```

4

5

Examiner comments

1 The candidate declares a list with the correct identifier so they are awarded the first marking point. The correct data is assigned to the list so the candidate is awarded the second marking point.

Mark for (a) = 2 out of 2

2 The candidate declares the function with the correct identifier, so they are awarded the first marking point. The search value (integer here) is passed as a parameter, and the array which although it should be globally declared will not impact the result and is therefore accepted. The candidate is awarded the second marking point.

2 The candidate's code does not perform a linear search, instead it uses an in-built Python function to identify if the result is in the array, which does not meet the requirements. It also re-inputs a value to search for in the array, but then the candidate does not use this in their answer. The spelling mistake in the array name will also stop the code functioning.

Mark for (b)(i) = 2 out of 6

4 A value is taken as input from the user so the candidate is awarded marking point 1. This value is cast as an integer, which is appropriate because the array contains integer values. This will also stop an error occurring, in some circumstances. The candidate is awarded marking point 2.

5 The function the candidate declares previously is not used to search for the data value.

Mark for (b)(ii) = 2 out of 4

Example Candidate Response – middle, continued

Examiner comments

Part 2(b)(iii)*(Copy and paste the screenshots for question 2(b)(iii) here)*

```
x = 0
arrayData = [10, 5, 6, 7, 1, 12, 13, 15, 21, 8]
print(arrayData)

x = int(input("Enter a value for search"))
print(arrayData[x], "the data was found")
if x not in arrayData:
    print("Data not found")
```

6

```
type TreasureChest:
    arrayTreasure = []
    arrayTreasure.append(treasureChest)
except
    print("File was not found")
```

Part 2(c)*(Copy and paste program code listing for question 2(c) here)*

```
def bubbleSort():
    temp : integer
    for y in range(0:11):
        if theArray[y] == theArray[y+1]:
            temp = theArray[y]
            theArray[y] == temp
            theArray[y+1] == theArray
```

7

6 The answer to this question should be a screenshot of the code being tested, so this does not meet the requirements of the question.

Mark for (b)(iii) = 0 out of 2

7 The candidate's answer does not follow the pseudocode given in the question to search the correct values. The first loop uses invalid Python syntax so the iteration will not work. The completed spaces are incorrect so the procedure will not work as required.

Mark for (c) = 0 out of 6

Total mark awarded = 6 out of 20

How the candidate could have improved their answer

- **(b)(i)** The candidate used an in-built Python function to check whether a value was within the array, but this did not meet the requirements of the question. The candidate needed to write a function that performed the actions given in the question, to search each array element in turn to identify if the parameter was in the list or not. They needed to declare the array as global from the previous question and so it would not need to be passed as a parameter to this function.
- **(b)(ii)** The candidate's code did not make use of the previous function they declared to identify if the data item input was in the array. Instead, it output a message which stated the data was found regardless of what data item was input. To improve this, the candidate needed to call their previously defined function with the data input, then check the return value and output a message according to its return value.
- **(b)(iii)** The candidate needed to present evidence of testing their program code. Instead they copied code from part of Question 2 and part of Question 3.
- **(c)** The candidate needed to test their program to find the syntax errors within their coded solution. They could then have tested their solution with the given array to refine the algorithm by identifying and correcting the errors. The candidate was given a pseudocode algorithm to follow and this was not followed accurately with elements missed out, for example, a loop.

Example Candidate Response – low	Examiner comments
Question 2 Part 2(a) Module Program Sub Main() Dim arraydata(10) As Integer Dim i As Integer i = 0 Console.WriteLine("Please enter the elements") For i = 1 to 10 Console.ReadLine() Next Console.WriteLine("Press any key to continue . . . ") Console.ReadKey(True) End Sub Part 2(b)(i) Module Program Sub Main() Dim arraydata(10) As Integer Dim i As Integer Dim X As Integer X = 1 i = 0	1 The candidate declares the array but local to the main program so they are not awarded the first marking point. The given data is not stored in the array so they are not awarded marking point 2. Mark for (a) = 0 out of 2 2 There is no evidence of a function named <code>linearSearch</code>. Mark for (b)(i) = 0 out of 6

Example Candidate Response – low, continued	Examiner comments
<pre>Console.WriteLine("Please enter the elements") For i = 1 to 10 Console.ReadLine() Next For i = 1 To 10 If (arraydata(10) = X) Then Console.WriteLine("True") Else Console.WriteLine("false") End If Next Console.WriteLine("Press any key to continue . . . ") Console.ReadKey(True) End Sub End Module</pre>	3 The loop does not check each data item in turn. It checks the data item in index 10 each time, so it does not perform a linear search. 4 The code outputs 'True' and 'False' instead of being returned, because there is no function declaration. Mark for (b)(ii) = 0 out of 4
Part 2(b)(ii) <pre>Module Program Sub Main() Dim arraydata(10) As Integer Dim i As Integer Dim X As Integer X = 1 i = 0 Console.WriteLine("Please enter the elements") For i = 1 to 10 Console.ReadLine() Next For i = 1 To 10 If (arraydata(10) = X) Then Console.WriteLine("True") Else Console.WriteLine("false") End If Next Console.WriteLine("Press any key to continue . . . ") Console.ReadKey(True) End Sub End Module</pre>	5 A value is read in using <code>Console.ReadLine()</code>, but this value is not stored in any form so it is not classed as an input. This needs to be stored or used in some way such as into a variable or sent to a function. The candidate does not declare the linear search function so it is not called. The output of an appropriate message could be awarded follow through marks for the search but the messages need to say more than 'True' or 'False' and they do not happen when the value is found or not found.

Example Candidate Response – low, continued	Examiner comments
Part 2(b)(iii) <pre> Module Program Sub Main() Dim arraydata(10) As Integer Dim i As Integer Dim X As Integer X = 1 Console.WriteLine("Please enter the elements") For i = 1 To 10 Console.ReadLine() Next For i = 1 To 10 If (arraydata(10) = X) Then Console.WriteLine("True") Else Console.WriteLine("false") End If Next Console.WriteLine("Press any key to continue ...") Console.ReadKey(True) End Sub End Module </pre> 6	6 The candidate needs to include a screenshot of the program being tested. Instead, they give another copy of their code. Mark for (b)(iii) = 0 out of 2
Part 2(c) <pre> Module Program Sub Main() Dim arraydata(10) As Integer Dim i As Integer Dim X As Integer X = 1 i = 0 Console.WriteLine("Please enter the elements") For i = 1 To 10 Console.ReadLine() Next For i = 1 To 10 If (arraydata(10) = X) Then Console.WriteLine("True") Else Console.WriteLine("false") End If Next Console.WriteLine("Press any key to continue ...") Console.ReadKey(True) End Sub End Module </pre> 7	7 The candidate does not show an attempt at this question. Instead they give the same code from their previous answers. Mark for (c) = 0 out of 6 Total mark awarded = 0 out of 20

How the candidate could have improved their answer

- (a) The candidate needed to declare the array globally but instead they showed the declaration within the main program. Where the candidate was given the data to store, they needed to use this accurately.
- (b)(i) The candidate should have declared a function for the linear search routine. Within this, a loop needed to check each array index in turn. Instead, the candidate looped the correct number of times (10) but checked index 10 each time and did not search each array element.
- (b)(ii) The candidate's code read a value using `Console.ReadLine()` but this value was not stored, so it was not classed as an input. The value needed to be stored or used in some way such as into a variable or sent to a function. The candidate did not declare the linear search function so it was not called. The output of an appropriate message could have been awarded a mark on follow through for the 'search' but the messages needed to say more than 'True' or 'False' and they did not happen when the value was found or not found.

Common mistakes candidates made in this question

- (a) Some candidates did not follow the instruction to use a global array or variable. Candidates needed to do this to allow the appropriate parameters to be used consistently.
- (b)(i) Some code output a message within the function instead of returning 'True' or 'False'.
- (b)(ii) Some code did not validate or cast the input data. This would reduce possible errors that could occur.
- (c) The question required the array to be sorted into descending order, this required a < (less than) instead of a > (greater than) in the third space. This was commonly given incorrectly, so some candidate's codes sorted the array into ascending order instead. The second loop counter was often too large, for example both from 0 to 9, which meant that when `theArray[y+1]` was accessed it would be out of bounds. This may not have been evident when testing if candidates declared an array (or list) with too many values.

Question 3

Example Candidate Response – high	Examiner comments
Question 3 Part 3(a) <pre> Public Class TreasureChest Private question As String Private answer As Integer Private points As Integer Sub New(ByVal ques As String, ByVal ans As Integer, ByVal pts As Integer) Me.answer=ans Me.question=ques Me.points=pts End Sub End Class </pre> 1 2	1 The candidate declares the class <code>TreasureChest</code> and includes <code>End Class</code>, so they are awarded marking point 1. 2 The candidate declares <code>question</code> as a string and within the class, so they are awarded marking point 2. They declare 'answer' as an integer and within the class so, are awarded marking point 3. They declare <code>points</code> as an integer and within the class so, are awarded marking point 4. The candidate declares all three attributes as private, so is awarded marking point 5. Mark for (a) = 5 out of 5

Example Candidate Response – high, continued

Part 3(b)

```

Sub readData()
    Try
        FileOpen(1,"C:\Users\lglab\Desktop\EG147_9618_42_4011\06_9618_42_Confidential Source Files
        June 2021\TreasureChestData.txt",openMode.input)
        Dim question As String
        Dim answer As Integer
        Dim points As Integer
        Dim line As String
        Dim arrayTreasure(4) As TreasureChest

        For i=0 To 4
            Input(1,question)
            Input(1,line)
            answer=Int(line)
            Input(1,line)
            points=Int(line)

            Dim obj As New TreasureChest(question,answer,points)
            arrayTreasure(i)=obj
        Next
        FileClose(1)
        Catch ex As Exception
            Console.WriteLine("cannot open file")
    End Try
End Sub

```

Examiner comments

3 The candidate declares the procedure with the correct identifier, so is awarded marking point 1.

4 The correct file is opened in read mode (input) so the candidate is awarded marking point 3.

5 The array is declared with 4 elements and the candidate uses the correct type of `TreasureChest` to be awarded marking point 2.

6 The candidate uses a `FOR` loop with 5 iterations to be awarded marking point 4.

7 Within the loop, three lines are read from the file in each iteration and each line read is stored in a variable, so the candidate is awarded marking point 5.

8 The candidate inserts the newly created object into the `arrayTreasure` array and uses the loop counter appropriately to control the index, so the candidate is awarded marking point 8.

9 The candidate declares a new object of type `TreasureChest` so they are awarded marking point 6. The 3 values read within the loop are sent as parameters to the constructor, so they are awarded marking point 7.

10 The five questions are all input within the loop and appended to the array correctly. The candidate is already awarded 8 marks, but otherwise they could be awarded marking point 9.

11 The opened file is closed appropriately. Although the candidate is already awarded 8 marks, this would meet marking point 12.

12 The candidate uses 'Try-Catch' as exception handling for opening and reading from the file. The candidate is already awarded 8 marks, but this would meet marking point 10. Within this 'Catch' an appropriate message is output which would meet marking point 11. Mark for (b) = 8 out of 8

Example Candidate Response – high, continued

Part 3(c)(i)

```

Public Function getQuestion() As String
    return Me.question
End Function

```

Part 3(c)(ii)

```

Public Function checkAnswer(ByVal inputAnswer As Integer) As Boolean
    If(inputAnswer=answer) Then
        Return True
    Else
        Return False
    End If
End Function

```

Part 3(c)(iii)

```

Function getPoints(ByVal attempts As Integer) As Integer
    Dim result As Integer
    If(attempts=1) Then
        result=points
    ElseIf (attempts=2) Then
        result=points/2
    Else If (attempts= 4 Or attempts=3 )Then
        result=points/4
    Else
        result=0
    End If
    return result
End Function

```

Examiner comments

13 The candidate declares the function with no parameter and it returns the correct attribute, so they are awarded the mark.

Mark for (c)(i) = 1 out of 1

14 The function has the correct identifier `checkAnswer`. It takes an integer parameter and the return data type is given correctly as boolean, so the candidate is awarded marking point 1.

15 The candidate's selection statement compares the parameter to the attribute `answer`, so they are awarded marking point 2.

16 'True' and 'False' are returned under the correct condition, so the candidate is awarded marking point 3. Mark for (c)(ii) = 3 out of 3

17 The function has the correct identifier. It takes an integer parameter and the return data type is given as `integer` which is accurate, so the candidate is awarded marking point 1.

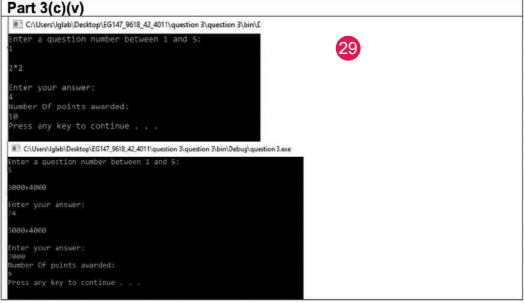
18 The condition uses the parameter and when this is 1 it stores `points` in `result` and then returns this at the end of the function, so the candidate is awarded marking point 2.

19 The candidate uses the correct operator for `DIV (/)`. They use this correctly when `attempts` is 2 and 3 or 4, so they are awarded marking points 3 and 4.

20 If the value is not 1, 2, 3 or 4, then 0 is stored in `result`. This is then returned, so they are awarded marking point 5.

Mark for (c)(iii) = 5 out of 5

Example Candidate Response – high, continued	Examiner comments
Part 3(c)(iv) <pre>Dim questionNum As Integer Dim answer As Integer Dim attemptsNum As Integer attemptsNum=0 readData() Console.WriteLine("Enter a question number between 1 and 5:") questionNum=Console.ReadLine()-1 Do attemptsNum=attemptsNum+1 Console.WriteLine("") Console.WriteLine(arrayTreasure(questionNum).getQuestion()) Console.WriteLine("") Console.WriteLine("Enter your answer:") answer=Console.ReadLine() Loop until (arrayTreasure(questionNum).checkAnswer(answer)=True) Console.WriteLine("Number Of points awarded:") Console.WriteLine(arrayTreasure(questionNum).getPoints(attemptsNum))</pre>	21 The procedure <code>readData</code> is called, so they are awarded marking point 1. 22 A data value is read in from the user and stored. They decrement this appropriately to give the suitable array index, so the candidate is awarded marking point 2. 23 The variable <code>attemptsNum</code> keeps track of the number of attempts by the user. This is initialised to 0 and then incremented within the loop each time an answer is given, so the candidate is awarded marking point 8. 24 The candidate uses the method <code>getQuestion</code> to access the question in the array at the index entered by the user (-1). This is output and so the candidate is awarded marking point 4. 25 The candidate uses the methods <code>getQuestion</code> and <code>checkAnswer</code> consistently and appropriately throughout their answer which would fulfil marking point 11 if they were not already awarded 7 marks. 26 A value is read in from the user as their answer, meeting marking point 5. 27 The loop includes a comparison of the answer given and the correct answer to the question, so the candidate is awarded marking point 6. The use of it in a <code>DO UNTIL</code> loop allows the program to repeatedly ask the question and read the answer until the user answers correctly. The candidate is awarded marking point 7. 28 The <code>getPoints</code> method is used correctly by the number of attempts being passed as a parameter. The candidate is already awarded 7 marks, but this would fulfil marking point 9. The same statement outputs the return value which fulfils marking point 10. Mark for (c)(iv) = 7 out of 7

Example Candidate Response – high, continued	Examiner comments
Part 3(c)(v) 	29 The screenshots show the correct values are entered by the candidate. They clearly show the outputs which are all accurate. The candidate is awarded both marking points 1 and 2. Mark for (c)(v) = 2 out of 2 Total mark awarded = 31 out of 31

How the candidate could have improved their answer

- The candidate's answers met all the given requirements. They could have included comments in their code to support their logic and any inbuilt functions (such as the file reading) they used.
- (c)(iv) A value was read in correctly from the user but was not validated. The candidate could have validated the input because it needed to be either 1, 2, 3, 4 or 5, so any other number or character would have been invalid and would have caused the program to report a runtime error.

Example Candidate Response – middle	Examiner comments
Question 3 Part 3(a) <pre>class TreasureChest: def __init__(self, question, answer, points): self.__question = question #STRING self.__answer = answer #INTEGER self.__points = points #INTEGER</pre>	1 The candidate correctly declares <code>TreasureChest</code> as a class and is awarded marking point 1. 2 The candidate uses comments appropriately to give the data type of each attribute. <code>question</code> is declared as a string, so they are awarded marking point 2, <code>answer</code> is declared as an integer, so they are awarded marking point 3 and <code>points</code> is declared as an integer, so they are awarded marking point 4. 3 The candidate uses <code>self.__</code> appropriately to identify that these attributes are intended as private and they are awarded marking point 5. Mark for (a) = 5 out of 5

Example Candidate Response – middle, continued	Examiner comments
Part 3(b) <pre>filenames = ["123", "4", "10", "100/10", "10", "10", "1000/4", "4000", "20", "123/23", "5", "50", "5000+5000", "5000", "10"] questions = [] answers = [] points = [] arrayTreasure = [] def readData(filename): for i in range(0, len(filename)): if i % 3 == 0: questions.append(filename[i]) if i % 3 == 1: answers.append(filename[i]) if i % 3 == 2: points.append(filename[i]) count = 1 for i in range(0,4): questionName = "Question" + str(count) count += 1 questions = TreasureChest(questions[i], answers[i], points[i]) arrayTreasure.append(questionName)</pre> Part 3(c)(i) <pre>class TreasureChest: def __init__(self, question, answer, points): self.__question = question #STRING self.__answer = answer #INTEGER self.__points = points #INTEGER def getQuestion(self): return self.__question</pre>	4 The candidate enters the file data into a list manually, instead of reading the data in from the file. This prevents them from being awarded any of the remaining marking points because these all require the reading of data from the file. Although the candidate does not follow the requirements, using the file data in this way does allow them to continue with later question parts that they can produce answers for, with the required data. 5 The procedure is correctly declared as <code>readData</code>, so they are awarded marking point 1. Mark for (b) = 1 out of 8 6 The method has the correct identifier and does not take a parameter. The code returns the appropriate attribute, so the candidate is awarded marking point 1. Mark for (c)(i) = 1 out of 1

Example Candidate Response – middle, continued

Part 3(c)(ii)

```
class TreasureChest:
    def __init__(self, question, answer, points):
        self.__question = question $STRING
        self.__answer = answer $INTEGER
        self.__points = points $INTEGER

    def getQuestion(self):
        return self.__question

    def checkAnswer(self, answer):
        if answer == self.__answer:
            return True
        else:
            return False
```

9

Part 3(c)(iii)

```
class TreasureChest:
    def __init__(self, question, answer, points):
        self.__question = question $STRING
        self.__answer = answer $INTEGER
        self.__points = points $INTEGER

    def getQuestion(self):
        return self.__question

    def checkAnswer(self, answer):
        if answer == self.__answer:
            return True
        else:
            return False

    def getPoints(self, attempts):
        if attempts == 1:
            return self.__points
        elif attempts == 2:
            return (self.__points//2)
        elif (attempts == 3) or (attempts == 4):
            return (self.__points//4)
        else:
            return 0
```

14

10

11

12

13

Examiner comments

7 The candidate declares the method with an appropriate parameter, so they are awarded marking point 1.

8 The candidate uses selection to compare the parameter to the private attribute which meets marking point 2.

9 'True' and 'False' are both returned under the correct conditions, so the candidate is awarded marking point 3.

Mark for (c)(ii) = 3 out of 3

10 The candidate declares the function appropriately and takes the required parameter, so they are awarded marking point 2.

11 When the parameter is 1 the attribute is returned, so the candidate is awarded marking point 2.

12 The candidate uses the Python operator for DIV (//) appropriately in each calculation.

13 The return values for attempts being 2, and 3 or 4 both return the correct value, so the candidate is awarded marking points 3 and 4.

14 When none of the values are met the correct value of 0 is returned and the candidate is awarded marking point 5.

Mark for (c)(iii) = 5 out of 5

Example Candidate Response – middle, continued

Part 3(c)(iv)

```
questions = []
answers = []
points = []
arrayTreasure = []

def readData(fileData):
    for i in range(0, len(fileData)):
        if i % 3 == 0:
            questions.append(fileData[i])
        if i % 3 == 1:
            answers.append(fileData[i])
        if i % 3 == 2:
            points.append(fileData[i])

    count = 1
    for i in range(0,4):
        questionName = "Question" + str(count)
        count += 1
        questionName = TreasureChest(questions[i], answers[i], points[i])
        arrayTreasure.append(questionName)

readData(fileData)
questionNumber = int(input("Enter a question number between 1 and 5: "))
print("Question:", arrayTreasure[questionNumber-1].getQuestion())

questionCorrect = False
questionCounter = 1
while questionCorrect == False:
    answer = input("Enter answer: ")
    if arrayTreasure[questionNumber-1].checkAnswer(answer) == True:
        points = arrayTreasure[questionNumber-1].getPoints(questionCounter)
        print("Correct. Points awarded: ", points)
        questionCorrect = True
    else:
        print("Incorrect")
        questionCounter += 1
```

Examiner comments

15 The procedure is declared correctly and the value passed to follow through the candidate's previous response where the list is used instead of the file. They are awarded marking point 1.

16 The question number is taken as input from the user and the candidate is awarded marking point 2.

17 The candidate uses both the getQuestion and checkAnswer methods. They are already awarded 7 marks, but this would fulfil marking point 11.

18 The question in the array at the index entered by the user is output, so the candidate is awarded marking point 4.

19 The answer is input by the user and stored in the variable answer, so the candidate is awarded marking point 5.

20 The selection statement compares the answer using the correct checkAnswer method, sending the value as input. The candidate is awarded marking point 6. This is within the while loop that continues until the answer input is correct, so the candidate is awarded marking point 7.

21 The candidate uses the getPoints method when the correct answer is input. They are already awarded 7 marks but this would fulfil marking point 9. The return value is stored in the variable points that is then output and this would fulfil marking point 10.

22 The candidate starts the question counter at 1 and it goes up in increments each time the answer given is incorrect. This keeps track of the number of attempts and the candidate is awarded marking point 8.

Mark for (c)(iv) = 7 out of 7

Example Candidate Response – middle, continued	Examiner comments
Part 3(c)(v) <pre>===== RESTART: N:\Questions\question3.py ===== Enter a question number between 1 and 5: 1 Question: 2*2 Enter answer: 4 Correct. Points awarded: 10 >>> ===== RESTART: N:\Questions\question3.py ===== Enter a question number between 1 and 5: 5 Question: 3000+4000 Enter answer: 3 Incorrect Enter answer: 7000 Correct. Points awarded: 9 >>></pre>	23 The candidate gives two screenshots. Both show the correct inputs and outputs, and they are awarded marking points 1 and 2. Mark for (c)(v) = 2 out of 2 Total mark awarded = 24 out of 31

How the candidate could have improved their answer

- (b) The candidate needed to read the data from a file and then appropriately manipulate and store this. Instead, they entered their data manually which did not meet the requirements of the question. By doing this, however, they were able to continue with the further questions that would still work with the given data.
- (c)(iv) A value was read incorrectly from the user, but the candidate did not validate it. To improve their answer, the candidate could have validated the input because it needed to be either 1, 2, 3, 4 or 5 so any other number or character would have been invalid and would have caused the program to report a runtime error.

Example Candidate Response – low	Examiner comments
Question 3 Part 3(a) <pre>Public Class TreasureChest Private Dim question As String Private Dim answer As Integer Private Dim points As Integer Dim t1 As New TreasureChest t1 = (Me.question, Me.answer, Me.points) Public Function getQuestion() As String Return Me.question End Function Public sub checkAnswer(ByVal input As Integer) input = Console.ReadLine() Dim correct As Boolean If input = Me.answer Then correct = True Else correct = False End If Console.WriteLine(correct) End Sub Public Function getPoints(ByVal attempts As Integer) As Integer Dim input As Integer input = Console.ReadLine() Dim correct As Boolean Dim wrong As Integer If input = Me.answer Then correct = True Else correct = False wrong = wrong + 1 attempts = wrong End If return Me.points End Function End Class</pre>	1 The candidate correctly declares the class <code>TreasureChest</code> and they include 'end class', so they are awarded marking point 1. 2 The candidate declares <code>question</code> as a string and it is clearly within the class as an attribute, so they are awarded marking point 2. They declare <code>answer</code> as an integer within the class, so are awarded marking point 3. They declare <code>points</code> as an integer within the class and are awarded marking point 4. 3 The candidate declares all three attributes as private using the key word, so is awarded marking point 5. 4 The answer contains code for later questions, but this does not impact the answer. Only the relevant code is considered. Mark for (a) = 5 out of 5

Example Candidate Response – low, continued

Part 3(b)

```
Sub readData(question As String, answer As Integer, points As Integer)
    For i = 1 To 15
        For j = 2,4,6,8,10,12,14
            dataItem = question
            Next
        Try
            FileOpen(1,C:\Users\lglab\Desktop\EG100_9618_42_5023,OpenMode)
            If Not EOF Then
                question =
            End If
        Catch
        end try
    Next
```

Part 3(c)(i)

```
Public Function getQuestion() As String
    Return Me.question
End Function
```

Part 3(c)(ii)

```
Public sub checkAnswer(byval input As Integer)
    input = Console.ReadLine()
    Dim correct As Boolean
    If input = Me.answer Then
        correct = True
    Else
        correct = False
    End If
    Console.WriteLine(correct)
End Sub
```

Examiner comments

5 The candidate attempts exception handling with the file opening within the 'try catch' statement, so they are awarded marking point 10. They are not awarded marking point 11 because the 'try' statement is incomplete.

6 The filename is not given to demonstrate that the correct file is being opened.

Mark for (b) = 1 out of 8

7 The candidate declares the method with the correct identifier. There is no parameter and the attribute is returned, so the candidate is awarded the mark.

Mark for (c)(i) = 1 out of 1

8 The candidate declares a procedure instead of a function, which is required for the return of a Boolean value. Otherwise, the identifier and parameter are both correct.

9 A comparison is made of a value with the attribute answer, however this parameter has been overwritten by taking an input from the user which means this comparison is inaccurate.

10 The values assigned to correct are output at the end of the subroutine, but the question required these to be returned to the calling program.

Mark for (c)(ii) = 0 out of 3

Example Candidate Response – low, continued

Part 3(c)(iii)

```
Public Function getPoints(ByVal attempts As Integer) As Integer
    Dim input As Integer
    input = Console.ReadLine()
    Dim correct As Boolean
    Dim wrong As Integer
    If input = Me.answer Then
        correct = True
    Else
        correct = False
        wrong = wrong + 1
        attempts = wrong
    End If
    If wrong = 1 Then
        Console.WriteLine(points)
    ElseIf wrong = 2 Then
        Console.WriteLine(points / 2)
    ElseIf wrong = 3 Or 4 Then
        Console.WriteLine(points / 4)
    Else
        Console.WriteLine("0")
    End If
    return Me.points
End Function
```

Part 3(c)(iv)

```
readData(arraytreasure())
    console.WriteLine("Choose number of question")
    Dim chosen As Integer
    If chosen = question Then
        console.WriteLine(question(chosen))
    Else
        console.WriteLine("Invalid question number!")
    End If
    checkAnswer(chosen)
    console.WriteLine(getpoints(attempts))
```

Part 3(c)(v)

{Copy and paste screenshots of the two tests for question 3(c)(v) here}

Examiner comments

11 The candidate declares function with the correct identifier. A parameter is sent of the correct data type. The return data type is given as Integer which is correct, so the candidate is awarded marking point 1.

12 The candidate's algorithm does not compare the parameter to the points values (1-4). Instead their code attempts to compare a new answer to the attribute answer and change the number of points.

Mark for (c)(iii) = 1 out of 5

13 The candidate calls their declared procedure readData. A parameter is sent, which is not a requirement, but this achieves follow through because the candidate has called their declared function appropriately and it is not unreasonable, so they are awarded marking point 1.

14 The candidate outputs a request for the question number but the code does not read the answer in from the user. They then compare the question number to a variable, but from the code provided it is not clear what this variable is or what it stores, so they are not awarded the marks.

Mark for (c)(iv) = 1 out of 7

Mark for (c)(v) = 0 out of 2

Total mark awarded = 9 out of 31

How the candidate could have improved their answer

- (b) The candidate began to open the file but did not include the appropriate filename to demonstrate that the correct file was being opened. Even without accessing the file, the candidate could have continued with their code to access elements, or declare and instantiate objects, including closing the file.
- (c)(ii) The candidate needed a return value from the subroutine, so it needed to be a function. The subroutine correctly had a parameter passed to it, but this was then overwritten by reading an input from the user, so the candidate could not be awarded marking point 2.

- **(c)(iii)** The candidate's function attempted to read in an answer from the user, compare this to the parameter and then increment a counter for each attempt but this was not a requirement of this function. This function should have taken the number of attempts and then used this to calculate the number of points. When calculating the number of points, the candidate used the division operator instead of the required `DIV` operator for `VB.NET`.
- **(c)(iv)** The candidate output a request for the question number to be input, but they did not read in the answer from the user. The candidate needed to use the array and methods declared to access the appropriate question and answer. They attempted to output the number of points, but they needed to access the method of the object and not call an independent function.
- **(c)(v)** The candidate did not attempt this question.

Common mistakes candidates made in this question

- **(a)** Some candidates did not declare the attributes as private. This could be done in different ways depending on the programming language. When writing in Python, candidates needed to make clear the data type of the attributes. This could be done using comments or by another method, provided this was clear.
- **(b)** Some candidates did not make the data type clear in their answer, when an array needed to be declared of a given data type. In Python, a comment could have been used to identify the number of elements and the data type, or empty elements could have been declared of the correct number and type.
- **(c)(i)** Some candidates took a parameter and read this in or read in a value as input and then returned this value. This method should have returned the attribute without any additions to it.
- **(c)(ii)** Some candidates output the result instead of returning the required Boolean value. This function was a method within the class and some answers declared an independent function that did not make use of the attributes.
- **(c)(iii)** Some candidates used standard division in place of integer division. In some cases, this was then cast appropriately as an integer which performed the same function. Some candidates used modular division instead of integer division.
- **(c)(iv)** Some candidates accessed the question and answer directly from the array or from another structure declared separately, instead of using the methods to access the objects. The number of attempts was often incremented or initialised in a way that meant if the answer was correct first time, then the number of attempts was 0 and not 1, which gave the incorrect number of points.