
COMPUTER SCIENCE

9608/22

Paper 2 Written Paper

May/June 2016

MARK SCHEME

Maximum Mark: 75

Published

This mark scheme is published as an aid to teachers and candidates, to indicate the requirements of the examination. It shows the basis on which Examiners were instructed to award marks. It does not indicate the details of the discussions that took place at an Examiners' meeting before marking began, which would have considered the acceptability of alternative answers.

Mark schemes should be read in conjunction with the question paper and the Principal Examiner Report for Teachers.

Cambridge will not enter into discussions about these mark schemes.

Cambridge is publishing the mark schemes for the May/June 2016 series for most Cambridge IGCSE[®], Cambridge International A and AS Level components and some Cambridge O Level components.

| Question | Answer | | | | | Marks |
|--|-----------------------------|--|------------------|------------------|-------------------|-------|
| 1 (a) (i) | Item | Statement | Selection | Iteration | Assignment | 6 |
| | 1 | MyScore = 65 | | | ✓ | |
| | 2 | FOR IndexVal = 0 TO 99 | | ✓ | | |
| | 3 | MyArray[3] = ID(MyString, 3, 2) | | | ✓ | |
| | 4 | IF MyScore >= 70 THEN | ✓ | | | |
| | 5 | ENDWHILE | | ✓ | | |
| | 6 | ELSE Message = "Error" | ✓ | | ✓ | |
| One mark per row Additional ticks in any row cancels that row | | | | | | |
| (ii) | Item | Purpose of statement | | | | 6 |
| | 1 | Assign <u>65</u> to <u>MyScore</u> | | | | |
| | 2 | (Start of) loop with loop counter starting from zero & going to 99 / repeating 100 times | | | | |
| | 3 | Assign 2 chars from position 3/4 in MyString to MyArray element 3/4 | | | | |
| | 4 | Test if MyScore is <u>greater than or equal to</u> 70 | | | | |
| | 5 | Marks the end of WHILE / precondition loop //Return to top of loop to check condition | | | | |
| | 6 | If a condition is FALSE, variable Message is assigned the value "ERROR" | | | | |
| Exact wording not important Explanation must refer to variables or values used in code (except for row 5) | | | | | | |
| (iii) | Expression | | Result | | | 2 |
| | "D" & RIGHT(MyString, 4) | | "Dance" | | | |
| | LEFT(RIGHT(MyString, 7), 3) | | "ten" | | | |
| Must have correct case Quotation marks optional | | | | | | |

| | | | |
|---------------|---|-----------------|--------------|
| Page 3 | Mark Scheme | Syllabus | Paper |
| | Cambridge International AS/A Level – May/June 2016 | 9608 | 22 |

| Question | Answer | | | Marks |
|--|-------------------|--------------------------------|---|--------------|
| 2 (a) | Identifier | Data Type | Description | 4 |
| | AlarmState | BOOLEAN | Alarm is set to ON or OFF | |
| | SensorValue | INTEGER | Value / number from sensor / as input by user // used in calculation of Temperature | |
| | ThresholdValue | REAL / FLOAT / SINGLE / DOUBLE | Threshold value for comparison | |
| | Temperature | REAL / FLOAT / SINGLE / DOUBLE | Temperature value calculated from sensor value | |
| <p>One mark per row Data types as shown Descriptions given above are examples only</p> | | | | |

| Question | Answer | Marks |
|----------|---|-------|
| (b) | <pre> AlarmState ← FALSE ① INPUT ThresholdValue INPUT SensorValue } ② Temperature ← SensorValue * 1.135 ③ IF Temperature > ThresholdValue THEN AlarmState ← TRUE OUTPUT "Temperature Alarm" } ⑤ ELSE OUTPUT "Temperature OK" ⑥ AlarmState ← FALSE ⑦ ENDIF </pre> <p>④</p> | Max 6 |
| | <p>Mark points as circled, descriptions as below:</p> <ol style="list-style-type: none"> Setting AlarmState to FALSE (Cond. check not essential but must be correct if present) Inputting SensorVal and ThresholdVal Correct value assigned to Temperature (must be * not x) IF..THEN..ELSE..ENDIF structure with correct condition (or two separate IF clauses) Correct THEN statements as shown Correct ELSE statement as shown Setting AlarmState to FALSE within ELSE clause only if mark point 1 not given | |

| | | | |
|--------|--|----------|-------|
| Page 5 | Mark Scheme | Syllabus | Paper |
| | Cambridge International AS/A Level – May/June 2016 | 9608 | 22 |

| Question | Answer | Marks |
|----------|--|-------|
| 3 (a) | <pre> FUNCTION EncryptString (<u>Lookup : ARRAY, PlainText : STRING</u>) RETURNS STRING DECLARE <u>OldChar, NewChar</u> : CHAR DECLARE OldCharValue : <u>INTEGER</u> DECLARE OutString: STRING //first initialise the return string <u>OutString</u> ← "" //initialise the return string //loop through PlainText to produce OutString FOR n ← 1 to <u>LENGTH(PlainText)</u> //from first to last character OldChar ← <u>MID(PlainText, n, 1)</u> //get next character OldCharValue ← <u>ASC(OldChar)</u> //find the ASCII value NewChar ← <u>Lookup[OldCharValue]</u> //look up substitute character <u>OutString</u> ← <u>Outstring & NewChar</u> // concatenate to OutString ENDFOR <u>RETURN OutString</u> // <u>EncryptString</u> ← <u>OutString</u> ENDFUNCTION </pre> <p>One mark for each part-statement (shown underlined and bold)</p> | 10 |
| (b) (i) | <p>VB: <u>Dim Lookup(0 to 127 / 128) As CHAR</u></p> <p>Pascal: <u>Var Lookup: Array[0..127 / 1..128] Of CHAR</u></p> <p>Python: Lookup = ["" for i in range(128)]</p> <p>OR</p> <pre> Lookup = [] For i in range(128) : Lookup.append("") </pre> <p>Mark as follows: VB / Pascal: one mark per part-statement as underlined and bold Python: One mark for Lookup and [] One mark for range(128)</p> | 2 |

| Question | Answer | Marks |
|----------|--|-------|
| (ii) | <p>'Pseudocode' solution included here for development and clarification of mark scheme. Programming language solutions appear in the Appendix.</p> <pre> INPUT StartPos INPUT NumToChange } ① ② { FOR n ← 0 to NumToChange - 1 OUTPUT " Input new value for position " ③ INPUT NewChar ④ Lookup[StartPos + n] ← NewChar ⑤ ENDFOR OUTPUT (NumToChange & " entries changed") ⑥ </pre> <p>ALTERNATIVE:</p> <pre> INPUT StartPos INPUT NumToChange } ① n ← 0 ② { REPEAT OUTPUT " Input new value for position " ③ INPUT NewChar ④ Lookup[StartPos + n] ← NewChar ⑤ n ← n + 1 UNTIL n = NumToChange OUTPUT (NumToChange & " entries changed") ⑥ </pre> <p>Mark points as circled, descriptions as below:</p> <ol style="list-style-type: none"> Two INPUT statements Working loop (using values of n from flowchart) OUTPUT prompt (exact text not specified) INPUT NewChar Assignment of NewChar to correct array element OUTPUT final message after loop (exact text not specified but must include NumToChange or loop counter if value correct at that point) | 6 |
| 4 (a) | <ul style="list-style-type: none"> Program code is <u>easier</u> to implement / manage Modules may be given to different people to develop // given to program specialists Program code is <u>easier</u> to test / debug / maintain Encourages the re-usability of program code | Max 2 |

| Question | Answer | Marks |
|-----------|---|-------|
| (b) (i) | <p>One mark per correct annotation as shown Arrows may be drawn clockwise or anticlockwise Diamond symbol may be filled or unfilled but must be in position shown</p> | 3 |
| (ii) | <p>A (or B) – Card details / Card number / Card info B (or A) – Cost details / amount payable / product cost / total bill C – (Flag) indicator for successful payment // payment confirmation</p> <p>Data items for A and B are interchangeable</p> | 3 |
| 5 (a) (i) | <ul style="list-style-type: none"> • So that the data / information is saved after the program is run / when the computer is switched off • So the data / information can be accessed next time the program is run • So the data information can be "permanently stored" | Max 1 |

| | | | |
|--------|--|----------|-------|
| Page 8 | Mark Scheme | Syllabus | Paper |
| | Cambridge International AS/A Level – May/June 2016 | 9608 | 22 |

| Question | Answer | Marks |
|----------|---|-------|
| (ii) | <p>Problem:</p> <ul style="list-style-type: none"> • When retrieving / searching for / editing (text relating to a particular CD) • Can't tell where the artist name stops and the title begins (or any similar explanation or example) <p>Solution 1:</p> <ul style="list-style-type: none"> • Use of a separator character// or by example • Where the separator character does not occur in the original strings <p>Solution 2:</p> <ul style="list-style-type: none"> • Use a fixed number of characters for each data item • Data items are padded with e.g. <Space> character where needed <p>Solution 3:</p> <ul style="list-style-type: none"> • Convert original data items to CamelCase • ...and add a Space separator <p>Mark as follows: Two marks for description of problem Two marks for description of solution</p> | 4 |

| | | | |
|--------|--|----------|-------|
| Page 9 | Mark Scheme | Syllabus | Paper |
| | Cambridge International AS/A Level – May/June 2016 | 9608 | 22 |

| Question | Answer | Marks |
|----------|--|-------|
| (b) | <p>'Pseudocode' solution included here for development and clarification of mark scheme. Programming language solutions appear in the Appendix.</p> <pre> PROCEDURE InputData() DECLARE CDTitle : STRING DECLARE CDArtist : STRING DECLARE CDLocation : STRING DECLARE FileData : STRING OPENFILE "MyMusic" FOR WRITE OUTPUT "Input CD Title" INPUT CDTitle WHILE CDTitle <> "##" OUPUT "Input CD Artist" INPUT CDArtist OUPUT "Input CD Location" INPUT CDLocation FileData = CDTitle & ':' & CDArtist & ':' & CDLocation WRITEFILE "MyMusic.txt", FileData OUTPUT "Input CD Title" INPUT CDTitle ENDWHILE CLOSEFILE("MyMusic.txt ") ENDPROCEDURE </pre> <p>One mark for each of the following:</p> <ul style="list-style-type: none"> • Procedure heading and ending • Declaration of CDTitle, CDArtist and CDLocation • Open file for writing (Allow MyMusic or MyMusic.txt) • Working conditional loop structure including test for rogue value (including initial input of CDTitle) • Input of three data values (CDTitle, CDArtist and CDLocation) inside a loop • String concatenation of three variables inside a loop • Write three variables in single line to file inside a loop • Close file • Use of string separator <p>Solutions may repeatedly OPEN – WRITE – CLOSE within the loop. In this case the first OPEN could be in WRITE or APPEND mode with all others in APPEND.</p> | Max 8 |

| Question | Answer | Marks | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|----------|---|-------|---|--------------------|--------------------|--------------------|--------------------|---|---|--|--|--|--|---|--|---|---|-----|-----|--|--|---|---|-----|-----|---|--|---|---|-----|-----|---|--|---|---|-----|-----|---|--|---|---|-----|-----|--|--|---|---|-----|-----|--|--|---|---|-----|-----|--|---|--|--|--|--|---|
| 6 (a) | <table border="1"> <thead> <tr> <th>n</th> <th>f</th> <th>x</th> <th>y</th> <th>MID(String1, x, 1)</th> <th>MID(String2, y, 1)</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>1</td> <td></td> <td>1</td> <td>1</td> <td>'R'</td> <td>'R'</td> </tr> <tr> <td></td> <td></td> <td>2</td> <td>2</td> <td>'E'</td> <td>'A'</td> </tr> <tr> <td>2</td> <td></td> <td>2</td> <td>1</td> <td>'E'</td> <td>'R'</td> </tr> <tr> <td>3</td> <td></td> <td>3</td> <td>1</td> <td>'T'</td> <td>'R'</td> </tr> <tr> <td>4</td> <td></td> <td>4</td> <td>1</td> <td>'R'</td> <td>'R'</td> </tr> <tr> <td></td> <td></td> <td>5</td> <td>2</td> <td>'A'</td> <td>'A'</td> </tr> <tr> <td></td> <td></td> <td>6</td> <td>3</td> <td>'C'</td> <td>'C'</td> </tr> <tr> <td></td> <td>4</td> <td></td> <td></td> <td></td> <td></td> </tr> </tbody> </table> <p>One mark per correct column</p> <ul style="list-style-type: none"> • Minus one mark if anything else on first row • Column 2 – '4' must not precede '4' in column 1 (as shown by arrow) • Letters must all be in upper case • Ignore quotation symbol | n | f | x | y | MID(String1, x, 1) | MID(String2, y, 1) | 0 | 0 | | | | | 1 | | 1 | 1 | 'R' | 'R' | | | 2 | 2 | 'E' | 'A' | 2 | | 2 | 1 | 'E' | 'R' | 3 | | 3 | 1 | 'T' | 'R' | 4 | | 4 | 1 | 'R' | 'R' | | | 5 | 2 | 'A' | 'A' | | | 6 | 3 | 'C' | 'C' | | 4 | | | | | 6 |
| n | f | x | y | MID(String1, x, 1) | MID(String2, y, 1) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | | 1 | 1 | 'R' | 'R' | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | 2 | 2 | 'E' | 'A' | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 2 | | 2 | 1 | 'E' | 'R' | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 3 | | 3 | 1 | 'T' | 'R' | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 4 | | 4 | 1 | 'R' | 'R' | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | 5 | 2 | 'A' | 'A' | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | 6 | 3 | 'C' | 'C' | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | 4 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| (b) (i) | <ul style="list-style-type: none"> • to search for a string within another string / String2 within String1 • to return the position of the start of String2 within String1 // by example <p>First mark point: allow locate / find / calculate position of</p> | 2 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| (ii) | <p>Value: 0 / zero</p> <p>Meaning: String2 not found in String1</p> | 2 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| (iii) | <p>Option 1</p> <ul style="list-style-type: none"> • It is possible to “fall off” the end of String1 (or by example) while string match (for example, String1 = "Retrace", String2 = "Raced") • MID (String1, x, 1) // description of 'subscript out of range' <p>Option 2</p> <ul style="list-style-type: none"> • If either string is empty then • subscript out of range error // description <p>Option 3</p> <ul style="list-style-type: none"> • If String1 found within String 2 • An endless loop will occur | 2 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

| | | | |
|----------------|---|-----------------|--------------|
| Page 11 | Mark Scheme | Syllabus | Paper |
| | Cambridge International AS/A Level – May/June 2016 | 9608 | 22 |

Appendix - Program Code Solutions

3 (b)(ii): VB.NET

```

Console.WriteLine("Enter start position: ")
StartPos = Console.ReadLine()
Console.WriteLine("Enter number to change: ")
NumToChange = Console.ReadLine()
For n = 0 To NumToChange - 1
    Console.WriteLine("Enter new value for position: ")
    NewChar = Console.ReadLine()
    Lookup(StartPos + n) = NewChar
Next
Console.WriteLine(NumToChange & " entries changed")

```

ALTERNATIVE:

```

Console.WriteLine("Enter start position: ")
StartPos = Console.ReadLine()
Console.WriteLine("Enter number to change: ")
NumToChange = Console.ReadLine()
n = 0
Do
    Console.WriteLine("Enter new value for position: ")
    NewChar = Console.ReadLine()
    Lookup(StartPos + n) = NewChar
    n = n + 1
Loop Until n = NumToChange
Console.WriteLine(NumToChange & " entries changed")

```

3 (b)(ii): Pascal

```

write('Enter start position: ');
readln(StartPos);
write('Enter number to change: ');
readln(NumToChange);
for n := 0 to NumToChange - 1 do
begin
    write('Input new value for position: ');
    readln(NewChar);
    LookUp[Startpos + n] := NewChar;
end;
writeln(IntToStr(NumToChange) + ' entries changed');

```

| | | | |
|----------------|---|-----------------|--------------|
| Page 12 | Mark Scheme | Syllabus | Paper |
| | Cambridge International AS/A Level – May/June 2016 | 9608 | 22 |

ALTERNATIVE:

```
write('Enter start position: ');
readln(StartPos);
write('Enter number to change: ');
readln(NumToChange);
n := 0;
repeat
    write('Input new value for position: ');
    readln(NewChar);
    LookUp[Startpos + n] := NewChar;
    n := n + 1;
until (n = NumToChange);
writeln(IntToStr(NumToChange) + ' entries changed');
```

3 (b)(ii): Python

```
StartPos = int(input("Enter start position: "))
NumToChange = int(input("Enter number to change: "))
for n in range(NumToChange) :
    NewChar = input("Input new value for position: ")
    LookUp[StartPos + n - 1] = NewChar
print(str(NumToChange) + " entries changed")
```

ALTERNATIVE:

```
StartPos = int(input("Enter start position: "))
NumToChange = int(input("Enter number to change: "))
n = 0
while n < NumToChange :
    NewChar = input("Input new value for position: ")
    LookUp[StartPos + n] = NewChar
    n = n + 1
print(str(NumToChange) + " entries changed")
```

| | | | |
|---------|--|----------|-------|
| Page 13 | Mark Scheme | Syllabus | Paper |
| | Cambridge International AS/A Level – May/June 2016 | 9608 | 22 |

5 (b): VB.NET

A StreamWriter() solution:

```

Sub InputData()
    Dim CDTitle, CDArtist, CDLocation As String
    Dim FileHandle As IO.StreamWriter
    FileHandle = New IO.StreamWriter("MyMusic.txt") ("MyMusic.txt")
    Console.WriteLine("Input CD Title: ")
    CDTitle = Console.ReadLine()
    Do Until CDTitle = "###"
        Console.WriteLine("Input CD Artist: ")
        CDArtist = Console.ReadLine()
        Console.WriteLine("Input CD Location: ")
        CDLocation = Console.ReadLine()
        FileHandle.WriteLine(CDTitle & ":" & CDArtist & ":" & CDLocation)
        Console.WriteLine("Input CD Title: ")
        CDTitle = Console.ReadLine()
    Loop
    FileHandle.Close()
End Sub

```

A legacy FileOpen() solution:

```

Sub InputData()
    Dim CDTitle, CDArtist, CDLocation As String
    FileOpen(1, "MyMusic", OpenMode.Output)
    Console.WriteLine("Input CD Title: ")
    CDTitle = Console.ReadLine()
    Do Until CDTitle = "###"
        Console.WriteLine("Input CD Artist: ")
        CDArtist = Console.ReadLine()
        Console.WriteLine("Input CD Location: ")
        CDLocation = Console.ReadLine()
        Print(1, CDTitle & ":" & CDArtist & ":" & CDLocation)
        Console.WriteLine("Input CD Title: ")
        CDTitle = Console.ReadLine()
    Loop
    FileClose(1)
End Sub

```

| | | | |
|----------------|---|-----------------|--------------|
| Page 14 | Mark Scheme | Syllabus | Paper |
| | Cambridge International AS/A Level – May/June 2016 | 9608 | 22 |

5 (b): Pascal

```

procedure InputData;
var
    CDTitle, CDArtist, CDLocation : string;
    CDFile : Textfile;
begin
    assign(CDFile, 'MyMusic');
    rewrite(CDFile);
    writeln('Input CD Title: ');
    readln(CDTitle);
    while (CDTitle <> '##') do
    begin
        writeln('Input CD Artist: ');
        readln(CDArtist);
        writeln('Input CD Location: ');
        readln(CDLocation);
        writeln(CDFile, CDTitle + ':' + CDArtist + ':' + CDLocation);
        writeln('Input CD Title: ');
        readln(CDTitle);
    end;
    close(CDFile);
end;

```

5 (b): Python

```

def InputData() :
    #CDTitle String (or CDTitle = "")
    #CDArtist String (or CDArtist = "")
    #CDLocation String (or CDLocation = "")

    FileHandle = open("MyMusic", "w")
    CDTitle = input("Input CD Title: ")
    while CDTitle != "##" :
        CDArtist = input("Input CD Artist: ")
        CDLocation = input("Input CD location: ")
        FileHandle.write(CDTitle + ":" + CDArtist + ":" + CDLocation)
        CDTitle = input("Input CD Title: ")
    FileHandle.close()

```