

**A-level Computer Science (7516/7517)**

Computer Science

Year 13

Name:

|  |
| --- |
|  |

Class:

|  |
| --- |
|  |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Q1 | Q2 | Q3 | Q4 | Q5 | Q6 | Q7 | Q8 | Q9 | Q10 | Q11 | Q12 | Q13 | Total |
| /1 | /3 | /6 | /12 | /4 | /3 | /11 | /4 | /10 | /18 | /15 | /13 | /20 | /120 |

|  |  |
| --- | --- |
|  | |
| Date: | **17 January 2017** |
| Time: | **120 minutes** |
| Marks: | **120 marks** |
| Instructions: | **This is an on-screen exam.**  **Internet access is prohibited.**  **Read through the entire paper before you begin.**  **Questions must be answered in the space provided.**  **Text must be clearly legible.**  **Source code must be inserted as neat screen snippets.**  **Source code must have a white background.**  **Answers that do not conform to the instructions will be disregarded.**  **This document must be printed before you leave the exam room. You will be provided printing time in addition to your 120 minutes should you need it.** |
|  | |

**Q1.**A computer system consists of hardware and software.

1. What is meant by *hardware*?

|  |
| --- |
| Hardware is the physical components. |

(a)     What is meant by *software*?

|  |
| --- |
| Software is the programs stored on the hardware. |

**(Total 1 mark)**

**Q2.**State **three** features of well-written program code that help to make it understandable without the need to include lots of comments.

|  |
| --- |
| Documentation on an external site.  Easy to understand code that is well written with no redundancies.  Well named variables. |

**(Total 3 marks)**

**Q3.**

(a)     A program is to be written which calculates the hourly pay rate for an employee. The calculation is based on the number of complete years the employee has worked for the firm (e.g. 3 years). All employees get a basic £7.88 per hour. For each year worked, up to a maximum of 5 years only, an additional £0.65 is added to the basic hourly rate.

The algorithm for this program is as follows:

1.      Enter the surname

2.      Enter the number of years of service

3.      Calculate the employee’s pay rate

4.      Output the surname and pay rate

(i)      Complete the table showing **three** variable identifiers and their data types you would use for this problem.

|  |  |  |
| --- | --- | --- |
|  | Variable Identifier | Data Type |
|  | **BaseValue** | **Constant double** |
|  | **IncrementalValue** | **Constant double** |
|  | **YearsOfService** | **Integer** |

**(3)**

(ii)     The detail for step 3 in the algorithm is broken down into more detail as follows:

3.1    If the number of years of service value is over 5, then change the value stored to 5

3.2    Calculate the employee’s pay rate

Write pseudo-code for these two steps using the appropriate identifiers from the table.

*(Reminder: alt+27 on the number pad for ← )*

3.1.

|  |
| --- |
| If YEARSOFSERVICE >=5 (YEARSOFSERVICE ← 5) |

3.2

|  |
| --- |
| TOTALPAY = BASEVALUE + (INCREMENTALVALUE \* YEARSOFSERVICE) |

**(3)**

**(Total 6 marks)**

**Q4.**

The image below shows an 8-bit bit pattern.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 0 |

1. If the bit pattern above is an **unsigned binary integer**, what is the denary equivalent of this bit pattern?

|  |
| --- |
| 182 |

**(1)**

1. If the bit pattern above is a **two’s complement binary integer**, what is the denary equivalent of this bit pattern?

|  |
| --- |
| 01001001 + 00000001 = 01001010 = 74 |

**(2)**

1. What is the range of **denary** numbers that can be represented using **8-bit two’s complement binary integers**?

|  |
| --- |
| 0 to 255 giving a total of 256 |

**(2)**

1. If the bit pattern above is an **unsigned binary fixed point** number with 3 bits before and 5 bits after the binary point, what is the denary equivalent of this bit pattern?

|  |
| --- |
| 101.10110 = 5.10110 = 5 1/22 |

**(2)**

1. What is the **hexadecimal** equivalent of the bit pattern above?

|  |
| --- |
| E6 |

**(2)**

1. Why are bit patterns often displayed using hexadecimal instead of binary?

|  |
| --- |
| They are shorter and take up less space. |

**(1)**

1. Describe a method that can, without the use of binary addition, multiply any **unsigned binary integer** by the binary number 10 (the denary number 2).

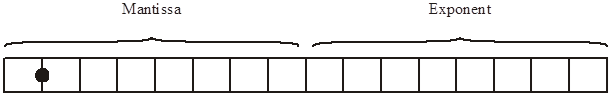
|  |
| --- |
| 111 \* 10 = (7 \* 2 = 14) = 1110 meaning that you just add a 0 to the end of th number. |

**(2)**

**(Total 12 marks)**

**Q5.**(a) By completing **Table 2**, show how the decimal value 57.0 could be stored in normalised floating point form as an 8 bit mantissa followed by an 8 bit exponent. Both mantissa and exponent are to be stored as signed values using two’s complement.

**Table 2**



Show your workings

|  |
| --- |
| **0 (positive) \* 00111001 (57) | 0 (nothing after 57, its .0)** |

**Answer**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **0** | **0** | **0** | **1** | **1** | **1** | **0** | **0** | **1** | **0** | **0** | **0** | **0** | **0** | **0** | **0** |

**(2)**

(e)     Give **two** advantages of normalised floating point format over fixed point format.

1 .

|  |
| --- |
| Smallers can be stored better. |

2 .

|  |
| --- |
| Numbers will be more accurate. |

**(2)**

**(Total 4 marks)**

**Q6.**

(a)     (i)      What is meant by a *parameter* of a procedure?

|  |
| --- |
| It is the input value that the procedure will use. For example, doubleproc(10) will give me an output of 20 as the procedure for doubleproc is to double it’s input and return an output. |

**(1)**

(ii)     What is meant by a *global variable*?

|  |
| --- |
| A variable that can be used all over the program. It is normally constant and will never change. |

**(1)**

(b)     When writing a procedure, why might a programmer prefer to use parameters rather than global variables?

|  |
| --- |
| Parameters allow for more complex tasks to be performed over a wide range of inputs, whereas global variable would need to be changed all the time, resulting in excess memory usage. |

**(1)**

**(Total 3 marks)**

**Q7.**A retail store employs ten sales staff. Staff try to persuade customers to take out a store card with the company when they make a purchase. The store keeps a record of the number of new store cards issued by its sales staff over the first six months of the year.

**Table 1**

StoreCards

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | [1] | [2] | [3] | [4] | [5] | [6] |
|  | [1] | 12 | 12 | 6 | 8 | 3 | 2 |
|  | [2] | 12 | 17 | 7 | 4 | 5 | 6 |
|  | [3] | 2 | 12 | 0 | 12 |  |  |
|  | [4] | 4 | 10 | 7 | 4 |  |  |
|  | [5] | 5 | 0 | 0 | 0 | 0 | 0 |
|  | [6] | 6 | 1 | 4 | 6 | 7 | 8 |
|  | [7] | 12 | 19 | 12 | **16** | 17 | 6 |
|  | [8] | 13 | 9 | 7 | 3 | 4 | 5 |
|  | [9] | 12 | 8 | 4 | 4 | 5 | 4 |
|  | [10] | 14 | 11 | 12 | 4 | 5 | 6 |

The data is to be stored in a 2-dimensional array with identifier StoreCards as shown in the table above The first subscript of the array represents the row number (the salesperson number), and the second subscript the column number (the month).

(a)     In the table the value 16 has been **emboldened**. Explain what this value represents.

|  |
| --- |
| Sales person 7 sold 16 store cards in the fourth month. |

**(2)**

(b)     Write a declaration statement for the array StoreCards.

|  |
| --- |
| //month #, sales person #  Dictionary<int, int> StoreCards = new Dictionary<int, int>(); |

**(2)**

(c)     Using the data given in the table above, write an assignment statement for the January sales for salesperson 8.

|  |
| --- |
| int StoreCards(1, 8) = null; |

**(2)**

(d)     Study the pseudo-code below.

Input SalesPersonNumber  
PersonTotal ← 0  
For Month ← 1 to 6 Do  
     PersonTotal← PersonTotal +   
     storeCards[SalesPersonNumber, Month]  
End For   
Print PersonTotal

Explain what this algorithm is designed to do.

|  |
| --- |
| For each month within the first 6 months, which is the maximum amount shown in the table, add up the total amount of store cards sold for everyone. |

**(2)**

(e)     A number of programs are to be written for the store card application, and the following are some of the data values which will need to be stored and/or calculated.

State what data type the programmer would use for each data item below.

(i)      Average overtime hours worked by each member of staff.

|  |
| --- |
| Double (for higher accuracy) |

**(1)**

(ii)     Whether or not the staff are willing to work on Boxing Day.

|  |
| --- |
| Boolean |

**(1)**

(iii)     The number of customer complaints made about each member of staff.

|  |
| --- |
| Integer. |

**(1)**

**(Total 11 marks)**

**Q8.**Traditionally, sound was recorded in analogue form, such as on vinyl records. For digital audio systems, the signals received from the microphone are sampled and the measurement of the amplitude can be stored as digital data. To reproduce the sound, the digital data is fed through a digital-to-analogue converter.

(a)     Give **two** factors which affect the quality of sound.

1

|  |
| --- |
| The maximum amount of hertz that the recording has. |

2

|  |
| --- |
| The quality of the playback device. |

**(2)**

(b)     What is possible when using the digital method of representing sound that could **not** be done with the sound recorded in analogue form?

|  |
| --- |
| The quality of the recording is never lowered by just playing the recording back. |

**(1)**

(c)     What is sound synthesis?

|  |
| --- |
| The generation of sound using programming such as sin wave generation. |

**(1)**

**(Total 4 marks)**

**Q9.**A *recursively-defined* procedure **ProcA** that takes two integers as parameters is defined below.

(a)     What is meant by a recursively-defined procedure?

|  |
| --- |
| Procedures that call upon themselves until a task is complete. |

**(1)**

(b)     What is the role of the stack when a recursively-defined procedure is executed?

|  |
| --- |
| The stack keeps track of where all the variables are. |

**(1)**

(c)     Dry run the procedure call **ProcA(11,1)** using the data in the array, **Items**, by completing the trace table below.

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  | Items |
|  | Procedure ProcA (Number,Entry)        If Number <> Items[Entry]        Then ProcA (Number,Entry+1)        Else Output (Entry)        EndIf  EndProc | [1] | 4 |
|  | [2] | 5 |
|  | [3] | 8 |
|  | [4] | 11 |
|  | [5] | 15 |
|  | [6] | 19 |
|  |  | [7] | 21 |
|  |  | [8] | 28 |
|  |  | [9] | 33 |

|  |  |  |  |
| --- | --- | --- | --- |
|  | Number | Entry | Output |
|  | 11 | 1 | 1 |
|  | 15 | 2 | 2 |
|  | 19 | 3 | 3 |
|  | 21 | 4 | 4 |
|  | 28 | 5 | 5 |

**(4)**

(d)     What is the purpose of this algorithm?

|  |
| --- |
| To use two streams of data and see if they ever intertwine. |

**(1)**

(e)     Give a situation where this algorithm will fail.

|  |
| --- |
| If the entry is higher than 9, the program will crash as there is no 9th entry. |

**(1)**

(f)      Suggest a modification to the algorithm that will prevent it from failing.

|  |
| --- |
| If items[entry] > 10  return |

**(1)**

(g)     With an ordered array, Items, of many more entries, what more efficient algorithm could be used to achieve your expressed purpose in part (d)?

|  |
| --- |
| A\* |

**(1)**

**(Total 10 marks)**

**Q10.**

Create a C# Console project for your new program.

The algorithm, represented the using pseudo-code below, and the variable table underneath, describe the process of using a check digit to check if a value entered by the user is a valid 13 digit International Standard Book Number (ISBN).

FOR Count ← 1 TO 13 DO  
  OUTPUT "Please enter next digit of ISBN: "  
  INPUT ISBN[Count]  
ENDFOR  
CalculatedDigit ← 0  
Count ← 1  
WHILE Count   
  CalculatedDigit ← CalculatedDigit + ISBN[Count]  
  Count ← Count + 1  
  CalculatedDigit ← CalculatedDigit + ISBN[Count] \* 3  
  Count ← Count + 1  
ENDWHILE  
WHILE CalculatedDigit >= 10 DO  
  CalculatedDigit ← CalculatedDigit – 10  
ENDWHILE  
CalculatedDigit ← 10 – CalculatedDigit  
IF CalculatedDigit = 10  
  THEN CalculatedDigit ← 0  
ENDIF  
IF CalculatedDigit = ISBN[13]  
   THEN OUTPUT "Valid ISBN"  
   ELSE OUTPUT "Invalid ISBN"  
ENDIF

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Identifier** | **Data Type** | **Purpose** |
|  | ISBN | Array[1..13] Of Integer | Stores the 13 digit ISBN entered by the user – one digit is stored in each element of the array. |
|  | Count | Integer | Used to select a specific digit in the ISBN. |
|  | CalculatedDigit | Integer | Used to store the digit calculated from the first 12 digits of the ISBN. It is also used to store the intermediate results of the calculation. |

**What you need to do**

Write a program for the algorithm above.

Test the program by showing the result of entering the digits 9, 7, 8, 0, 0, 9, 9, 4, 1, 0, 6, 7, 6 (in that order).

Test the program by showing the result of entering the digits 9, 7, 8, 1, 8, 5, 7, 0, 2, 8, 8, 9, 4 (in that order).

**Evidence that you need to provide**

1. Your PROGRAM SOURCE CODE screen capture(s).

|  |
| --- |
| // THIS MAY NOT WORK //  Question10.cs using System; //using System.\*; namespace Application {     public class Question10 {         int count;         int[] isbn = new int[13];         int caculatedDigit;                  public void Main(string[] args) {             for (int i = 0; count >= 1 && count <= 13; i++) {                 Console.Write("Please enter the next digit of ISBN: ");                 isbn[i];             }              calculatedDigit = 0;             count = 1;              while (count) {                 calculatedDigit = calculatedDigit + isbn[count];                 count++;                 calculatedDigit = calculatedDigit + isbn[count] \* 3;                 count++;             }              while (calculatedDigit >= 10) {                 calculatedDigit = calculatedDigit - 10;             }               calculatedDigit = 10 - calculatedDigit;              if (calculatedDigit = 10) {                 calculatedDigit = 0;             }              if (calculatedDigit = isbn[13]) {                 Console.WriteLine("Valid ISBN");             } else {                 Console.WriteLine("Invalid ISBN");             }                 }     } } |

**(15)**

(b)     SCREEN CAPTURE(S) for the test when the digits 9, 7, 8, 0, 0, 9, 9, 4, 1, 0, 6, 7, 6 are entered (in that order).

Your evidence must show the result of the test and, as a minimum, the last three digits entered for the test.

|  |
| --- |
|  |

**(2)**

(c)     SCREEN CAPTURE(S) for the test when the digits 9, 7, 8, 1, 8, 5, 7, 0, 2, 8, 8, 9, 4 are entered (in that order).

Your evidence must show the result of the test and, as a minimum, the last three digits entered for the test.

|  |
| --- |
|  |

**(1)**

**(Total 18 marks)**

**Q11.**Create C# Console project for your new program.

The variable table, the table given below, and the Structured English algorithm, the diagram below, describe a linear search algorithm that could be used with a simplified version of the Dice Cricket game to find out if a particular player’s name appears in the high score table.

In this simplified version only the names of the players getting a top score are stored. Their scores are **not** stored.

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Identifier** | **Data Type** | **Purpose** |
|  | Names | Array[1..4] of String | Stores the names of the players who have one of the top scores |
|  | PlayerName | String | Stores the name of the player being looked for |
|  | Max | Integer | Stores the size of the array |
|  | Current | Integer | Indicates which element of the array Names is currently being examined |
|  | Found | Boolean | Stores True if the player’s name has been found in the array,  False otherwise |

Names[1] ← ‘Ben’  
Names[2] ← ‘Thor’  
Names[3] ← ‘Zoe’  
Names[4] ← ‘Kate’  
Max ← 4  
Current ← 1  
Found ← False  
OUTPUT ‘What player are you looking for?’  
INPUT PlayerName  
WHILE (Found = False) AND (Current <= Max)  
  IF Names[Current] = PlayerName  
    THEN Found ← True  
    ELSE Current ← Current + 1  
  ENDIF  
ENDWHILE  
IF Found = True  
  THEN OUTPUT ‘Yes, they have a top score’  
  ELSE OUTPUT ‘No, they do not have a top score’  
ENDIF

**What you need to do**

•    Write a program for the above algorithm.

•    Test the program by searching for a player named ‘Thor’.

•    Test the program by searching for a player named ‘Imran’.

**Evidence that you need to provide.**

1. Your PROGRAM SOURCE CODE.

|  |
| --- |
|  |

**(11)**

1. SCREEN CAPTURE(S) for the test searching for ‘Thor’.

|  |
| --- |
|  |

**(2)**

1. SCREEN CAPTURE(S) for the test searching for ‘Imran’.

|  |
| --- |
|  |

**(2)**

**(Total 15 marks)**

**Q12.**

**Figure 1** contains the pseudo-code for a program to output a sequence according to the ‘Fizz Buzz’ counting game.

**Figure 1**

OUTPUT "How far to count?"  
INPUT HowFar  
WHILE HowFar < 1  
  OUTPUT "Not a valid number, please try again."  
  INPUT HowFar  
ENDWHILE

FOR MyLoop ← 1 TO HowFar  
  IF MyLoop MOD 3 = 0 AND MyLoop MOD 5 = 0  
  THEN  
      OUTPUT "FizzBuzz"  
    ELSE  
      IF MyLoop MOD 3 = 0  
        THEN  
          OUTPUT "Fizz"  
        ELSE  
          IF MyLoop MOD 5 = 0  
            THEN  
              OUTPUT "Buzz"  
            ELSE  
              OUTPUT MyLoop  
          ENDIF  
      ENDIF  
  ENDIF  
ENDFOR

**What you need to do:**

Write a program that implements the pseudo-code as shown in **Figure 1**.

Test the program by showing the result of entering a value of 18 when prompted by the program.

Test the program by showing the result of entering a value of -1 when prompted by the program.

**Evidence that you need to provide**

1. Your PROGRAM SOURCE CODE for the pseudo-code in **Figure 1**.

|  |
| --- |
|  |

**(8)**

1. SCREEN CAPTURE(S) for the tests conducted when a value of 18 is entered by the user and when a value of -1 is entered by the user.

|  |
| --- |
|  |

**(1)**

The main part of the program uses a FOR repetition structure.

(c)     Explain why a FOR repetition structure was chosen instead of a WHILE repetition structure.

|  |
| --- |
|  |

**(1)**

(d)     Even though a check has been performed to make sure that the variable HowFar is greater than 1 there could be inputs that might cause the program to terminate unexpectedly (crash).

Provide an example of an input that might cause the program to terminate and describe a method that could be used to prevent this.

|  |
| --- |
|  |

**(3)**

**(Total 13 marks)**

**Q13.**

Mr Wy Lee Coyote, a senior manager (non I.T.) at A.C.M.E. Corp., an international bank based in London, has recently heard about Big Data. He followed up by asking human resources (HR) how many Big Data specialists they currently employ. HR said they do not currently employ any, but they do have an apprentice programmer that might know something about Big Data in the I.T. department – you.

To be first into the *next big thing* and look good in front of the board for his next bonus review, Mr Coyote has decided that A.C.M.E. Corp. should employ a large team of Big Data specialists so that they can begin processing the banking data using this new fantastic technology that everyone is talking about.

He is far too busy to research Big Data, but being experienced in corporate climbing, he feels he should cover his bases and has instructed you to provide him with the motivation brief for his next meeting (you write it, he reads it and gets the credit).

As an apprentice, you are currently earning just over minimum wage. Mr Coyote earned more than your annual salary before the end of the first week of the year. This is your chance to be noticed. If you do well, there could be a significant promotion in it for you as his technology adviser.

You do not wish to be fired.

He is not interested in your opinion.

**What you need to do:**

You need to write a brief for Mr Coyote that provides him with:

* a clear understanding of Big Data, and
* a discussion about how it applies to A.C.M.E. Corp., and
* most importantly, sufficient information for him to make the right move.

*The brief should be in the form of a well-structured essay.*

*The quality of written communication of your answer will also be assessed.*

|  |
| --- |
|  |

**(Total 20 marks)**