

B10.

Consider the following problem statement:

The ATM cash machines of ABC Bank contain biometric security features besides conventional PIN control when a card is submitted.

- The customers of ABC Bank have two options to authenticate access to withdraw cash from an ATM:
 - either PIN or Biometric.
- Customers of other banks can also withdraw cash from an ABC ATM; however, they can only authenticate by PIN.
- A customer's card is retained by the ATM after three consecutive failed attempts.
- The withdrawn amount must not be more than the account balance.

Draw a flowchart showing the process of cash withdrawal from an ABC ATM. Assume that cash withdrawals are the only transactions allowed.

(12 marks)

B11.

Compare and contrast the following pairs of related terms:

a) White box testing vs black box testing.

(4 marks)

b) Unit testing vs functional testing.

(4 marks)

c) Scripted testing vs exploratory testing.

(4 marks)

B12.

Compare and contrast the following pairs of related terms: You should include a clear explanation of the purpose of **each**.

a) Compiler vs Interpreter.

(4 marks)

b) Linker vs Loader.

(4 marks)

c) Parser vs Lexical analyser.

(4 marks)

END OF EXAMINATION

BCS THE CHARTERED INSTITUTE FOR IT

**BCS HIGHER EDUCATION QUALIFICATIONS
BCS Level 4 Certificate in IT**

SOFTWARE DEVELOPMENT

Wednesday 17th April 2024 – Afternoon

Time: TWO hours

Section A and Section B each carry 50% of the marks.

You are advised to spend about 1 hour on Section A (30 minutes per question)
and 1 hour on Section B (12 minutes per question).

Answer any Section A questions you attempt in Answer Book A

Answer any Section B questions you attempt in Answer Book B

The marks given in brackets are **indicative** of the weight given to each part of the question.

Calculators are **NOT** allowed in this examination.

Section A

Answer two questions (out of four). Each question carries 30 marks.

A1.

- a) Write program code (using a language of your choice and without using functions) to implement the following algorithm to calculate pay owed to an employee:
- Get Hours Worked this week.
 - Check the number of hours is valid (more than 0 and not greater than 80).
 - Calculate Wages (a basic wage of £10.00 an hour. If an employee works over 40 hours a week, they receive £15.00 an hour for every hour of overtime they work).
 - Print Wages and Save to File.
- (10 marks)**
- b) Represent the algorithm as a flowchart.
- (5 marks)**
- c) Write code to show how **each** of the four steps in the algorithm above could be implemented as a separate function. You should also provide sample code to call these functions to calculate the wage or to display a suitable error message.
- (10 marks)**
- d) What are the general advantages of breaking program code into separate functions?
- (5 marks)**

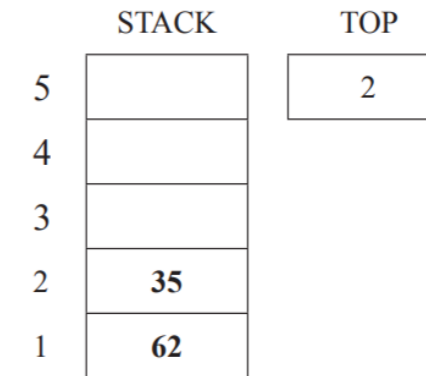
A2.

- a) Consider an array of positive and negative numbers. Using this array, write code in a language of your choice to achieve the following:
- i. Output the entire contents of the array using a 'for loop'.
 - ii. Double each number in the array. Save the resulting array called doubleNums.
 - iii. Use a filter to generate a new array containing the positive numbers in your original array.
 - iv. Instead of using a 'hard-coded' array of numbers, use a 'for loop' to repeatedly ask the user for a number. Each time the user answers with a number, store the answer in the next element of the array.
- (16 marks)**
- b) What is the difference between a stack and a queue? Give examples of situations where **each** of these might be used.
- (7 marks)**
- c) Explain how a binary search could be used to find the position of a specific item in a sorted array.
- (7 marks)**

B9.

- a) Consider Fig B9 below, an example of a stack, where top indicates the current position of the top of the stack.

Fig B9 Stack



Redraw Fig B9 to show the result of the following operations:

- i. Push(21)
 - ii. Push(10)
 - iii. Pop.
- (2 marks)**
- b) Describe **each** of the following data structures, including how each data structure manipulates data (specifically insertion and deletion). Provide examples with diagrams to assist your answer.
- i. Array
 - ii. Linked List.
- (8 marks)**
- c) Give **ONE** reason why a linked list is preferred to an array in some applications.
- (2 marks)**

[Turn Over]

B6.

- a) List and describe **FOUR** advantages of an object-oriented approach to programming. **(8 marks)**
- b) Describe the main characteristics of the functional programming paradigm. **(4 marks)**

B7.

There are many file formats that structure data in different ways. The file extension (for example *.jpg; *.pdf; *.html; *.csv; *.txt) determines the file format.

- a) Why is it important to know the file format before accessing data from such a file? **(2 marks)**
- b) Describe the file format that **each** of the **FIVE** file extensions listed above represents. **(10 marks)**

B8.

A popular type of user interface is based on a web form where the user fills in certain details that are required and submits the details to the web server for processing.

- a) Describe an application of your choosing that would use a web form. **(3 marks)**
- b) List **FOUR** types of GUI objects that could be used on a web form. Outline the overall function of **each** control. **(4 marks)**
- c) Draft out a design of a web form user interface to your chosen application that uses **FOUR** types of GUI objects. Justify **each** type of GUI object you used. **(5 marks)**

A3.

- a) Using a programming language of your choice, write sample code to ask the user a question then store their answer in a file. **(6 marks)**
- b) Explain the difference between sequential and random-access files and how **each** of these is supported in a programming language of your choice. **(6 marks)**
- c) Explain how comma separated value (CSV) files can be used to help share data between applications. Your explanation should include examples of situations where these are often used. **(6 marks)**
- d) How can abstract data types be used to make programs easier to understand? **(6 marks)**
- e) Explain the difference between an abstract data type and an abstract class as used in object-oriented programming. **(6 marks)**

[Turn Over]

A4.

- a) Roman numerals are represented by seven different symbols: I, V, X, L, C, D and M. These have the following meanings in denary:

Roman	Denary	Roman	Denary	Roman	Denary
I	1	V	5	X	10
L	50	C	100	D	500
M	1000				

For example, 2 is written as II in Roman numeral, just two one's added together. 12 is written as XII, which is simply X + II. The number 27 is written as XXVII, which is XX + V + II.

Roman numerals are usually written largest to smallest from left to right. For example:

MDCIII means 1000+500+100+3 = 1603

However, the numeral for four is not IIII. Instead, the number four is written as IV. Because the one is before the five, we subtract it making four. The same principle applies to the number nine, which is written as IX. There are six instances where subtraction is used:

I can be placed before V (5) and X (10) to make 4 and 9.

X can be placed before L (50) and C (100) to make 40 and 90.

C can be placed before D (500) and M (1000) to make 400 and 900.

- i. Given a roman numeral, explain how you would design a program to convert it to a denary integer. You do not need to write actual code but clearly present an algorithm that would solve this problem.

(8 marks)

- ii. Given an integer in denary, explain how you would design a program to convert it to a roman numeral. Again, you do not need to write actual code but clearly present an algorithm that would solve this problem.

(7 marks)

- b) A string is a variable that holds a sequence of one or more alphanumeric characters. These characters can be letters, numbers or symbols. It is usually possible to manipulate a string to provide information or to alter the contents of a string. Most programming languages include string functions to help with this.

For a programming language of your choice, explain how **each** of the following string functions is supported. You should provide an illustrative example of how **each** function could be used:

- Length
- Character position
- Substring
- Concatenation
- String conversion.

(15 marks)

Section B

Answer five questions (out of eight). Each question carries 12 marks.

B5.

The mean absolute deviation (MAD) measures the spread or dispersion of a data set around the mean.

It is calculated using the following formula:

$$\text{Mean Absolute Deviation} = \frac{1}{n} \sum_{i=1}^n |x_i - \text{mean}(X)|$$

Where **n** is the number of values in a dataset and **x_i** is the **ith** element in a list.

The above formula can be expressed in expanded form as follows:

$$\text{MAD} = (\text{abs}(x_1 - \text{mean}) + \text{abs}(x_2 - \text{mean}) + \text{abs}(x_3 - \text{mean}) \dots \text{abs}(x_n - \text{mean})) / n$$

Where the function **abs()** returns the absolute value of its argument

- a) Write out the steps needed to calculate the MAD. Show the result of applying **each** step to the data set {2, 6, 13, 7}.

(4 marks)

- b) Assume that a program called main() is provided containing the following pseudocode:

```
main()
Declare float arr[] = { 34, 21, 56, 76, 45, 11}
Declare int n = sizeof(arr)
Call fnDeviation(arr[], n)
```

- i. Write pseudocode for a function called fnMean(arr[], n) that returns the mean of n values contained in an array named arr[].

(4 marks)

- ii. Write pseudocode for a function called fnDeviation(arr[], n) that returns the Mean Absolute Deviation of the n values contained in an array named arr[].

Hint: The fnMean() needs to be incorporated in fnDeviation().

(4 marks)

[Turn Over]