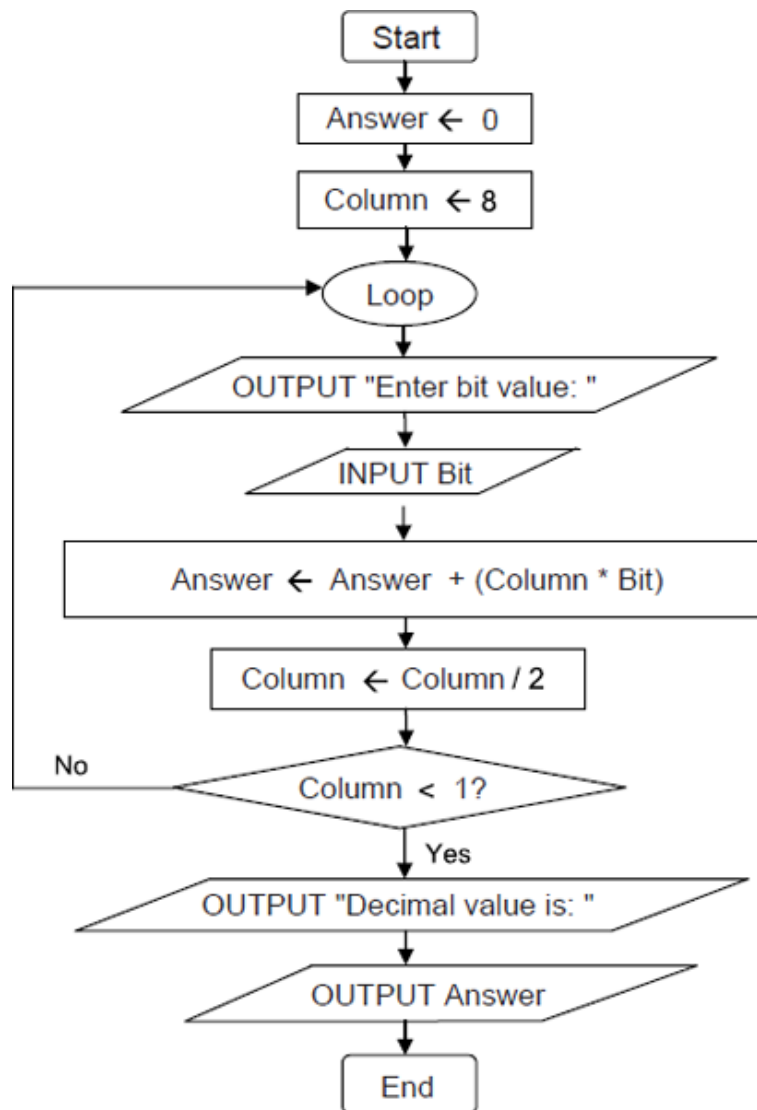


Q1.

Create a folder/directory for your new program.

The algorithm, represented as a flowchart below, and the variable table, describe the converting of a 4-bit binary value into denary.



Identifier	Data type	Purpose
Column	Integer	Stores the place value (column heading)
Answer	Integer	Stores the denary value equivalent to the bit pattern entered by the user
Bit	Integer	Stores a 0 or 1 entered by the user

What you need to do

Write a program for the above algorithm.

Test the program by showing the result of entering the values 1, 1, 0, 1 (in that order).

Save the program in your new folder/directory.

Evidence that you need to provide

(a) Your PROGRAM SOURCE CODE. (11)

(b) SCREEN CAPTURE(S) for the test described above. (3)

(c) What is the largest denary number that could be output by the algorithm represented by the flowchart in the diagram above?
..... (1)

(d) The algorithm represented by the flowchart above can convert sixteen different bit patterns into denary.

If the symbol

Column ← 8

 is changed to

Column ← 16

 how many **more** bit patterns could be converted into denary?

..... (1)

(e) When developing a new system the stages of the systems development life cycle could be followed.

At which stage of the systems development life cycle would the flowchart above have been created?

..... (1)

(f) At which stage of the systems development life cycle would the algorithm represented by the flowchart above be automated using a programming language?

..... (1)
(Total 18 marks)

Q2.

Create a folder/directory for your new program.

The variable table, the table below, and the Structured English algorithm below, describe a simplified version of a noughts and crosses match. A match consists of a user-specified number of games. In this simplified version, the two players complete each game on paper and then enter information about the result of each game into a program that totals the number of games won by each player. Assume that all games have a winner – there are no drawn games.

Identifier	Data Type	Purpose
NoOfGamesInMatch	Integer	Stores the number of games in the match (specified by user)
NoOfGamesPlayed	Integer	Stores the number of games played so far
PlayerOneScore	Integer	Stores the number of games won by Player One
PlayerTwoScore	Integer	Stores the number of games won by Player Two
PlayerOneWinsGame	Char	Stores a 'Y' if Player One won the game and 'N' otherwise

```
PlayerOneScore " 0
PlayerTwoScore " 0
OUTPUT "How many games?"
INPUT NoOfGamesInMatch
FOR NoOfGamesPlayed " 1 TO NoOfGamesInMatch Do
    OUTPUT "Did Player One win the game (enter Y or N)?"
    INPUT PlayerOneWinsGame
    IF PlayerOneWinsGame = 'Y'
        THEN PlayerOneScore " PlayerOneScore + 1
        ELSE PlayerTwoScore " PlayerTwoScore + 1
    ENDIF
ENDFOR
OUTPUT PlayerOneScore
OUTPUT PlayerTwoScore
```

What you need to do

Write a program for the above algorithm.

Test the program by showing the results of a match consisting of three games where Player One wins the first game and Player Two wins the second and third games.

Save the program in your new folder/directory.

Evidence that you need to provide

(a) Your PROGRAM SOURCE CODE.

(9)

(b) SCREEN CAPTURE(S) for the test described above.

(4)

(Total 13 marks)

Q3.

Create a folder/directory for your new program.

The variable table, **Table 1**, and the Structured English algorithm describe a simplified version of the **Guess the Word / Phrase Game**.

Table 1

Identifier	Data Type	Purpose
NewWord	String	Stores the setter's word to be guessed
UserWordGuess	String	Stores a word that is the user's guess

```
OUTPUT "The new word?"
INPUT NewWord
OUTPUT "Your guess?"
INPUT UserWordGuess
IF UserWordGuess IS EQUAL TO NewWord
    THEN OUTPUT "CORRECT"
    ELSE OUTPUT "INCORRECT"
ENDIF
```

What you need to do

Write a program for the above algorithm in the programming language of your choice.

Test the program as follows.

Test 1: Input of the new word EAGLE followed by a correct guess.

Test 2: Input of the new word BEAR followed by an incorrect guess.

Save the program in your new folder / directory.

Evidence that you need to provide

(a) SCREEN CAPTURES for the following tests:

(i) Test 1

(3)

(ii) Test 2

(3)

(b) Your PROGRAM SOURCE CODE.

(7)

(Total 13 marks)