0 1 Develop an algorithm using either pseudo-code or a flowchart that allows a taxi company to calculate how much a taxi fare should be.

The algorithm should:

- prompt the user to enter the journey distance in kilometres
  - the distance entered must be greater than zero
  - the user should be made to re-enter the distance until the distance entered is valid
- prompt the user to enter the number of passengers (no validation is required)
- calculate the taxi fare by
  - charging £2 for every passenger regardless of the distance
  - charging a further £1.50 for every kilometre regardless of how many passengers there are
- output the final taxi fare.

[8 marks]

Turn over for the next question

0 2 Run length encoding (RLE) is a form of compression that creates frequency/data pairs to describe the original data.

For example, an RLE of the bit pattern `00000011101111` could be `6 0 3 1 1 0 4 1` because there are six `0`s followed by three `1`s followed by one `0` and finally four `1`s.

The algorithm in **Figure 7** is designed to output an RLE for a bit pattern that has been entered by the user.

Five parts of the code labelled **L1**, **L2**, **L3**, **L4** and **L5** are missing.

- Note that indexing starts at zero.

Figure 7

```
pattern ← L1
i ← L2
count ← 1
WHILE i < LEN(pattern)-1
   IF pattern[i] L3 pattern[i+1] THEN
      count ← count + 1
   ELSE
      L4
      OUTPUT pattern[i]
      count ← 1
   ENDIF
   L5
ENDWHILE
OUTPUT count
OUTPUT pattern[i]
```

0 2 . 1 Shade **one** lozenge to show what code should be written at point **L1** of the algorithm.

[1 mark]

A `OUTPUT` ⬭

B `'RLE'` ⬭

C `True` ⬭

D `USERINPUT` ⬭

0 2 . 2 Shade **one** lozenge to show what value should be written at point **L2** of the algorithm.

[1 mark]

A -1 ⬭

B 0 ⬭

C 1 ⬭

D 2 ⬭

0 2 . 3 Shade **one** lozenge to show what operator should be written at point **L3** of the algorithm.

[1 mark]

A = ⬭

B ≤ ⬭

C < ⬭

D ≠ ⬭

0 2 . 4 Shade **one** lozenge to show what code should be written at point **L4** of the algorithm.

[1 mark]

A `count` ⬭

B `count ← count - 1` ⬭

C `count ← USERINPUT` ⬭

D `OUTPUT count` ⬭

0 2 . 5 Shade **one** lozenge to show what code should be written at point **L5** of the algorithm.

[1 mark]

A `i ← i * 2` ⬭

B `i ← i + 1` ⬭

C `i ← i + 2` ⬭

D `i ← i DIV 2` ⬭

0 2 . 6 State a run length encoding of the series of characters `ttjjeeess`

[2 marks]

0 2 . 7 A developer implements the algorithm shown in **Figure 7** and tests their code to check that it is working correctly. The developer tests it only with the input bit pattern that consists of six zeros and it correctly outputs `6 0`.

Using example test data, state **three** further tests that the developer could use to improve the testing of their code.

[3 marks]

0 3 A developer creates the algorithm shown in **Figure 8** to provide support for users of a new brand of computer monitor (display).

- Line numbers are included but are not part of the algorithm.

Figure 8

```
1   OUTPUT 'Can you turn it on?'
2   ans ← USERINPUT
3   IF ans = 'no' THEN
4      OUTPUT 'Is it plugged in?'
5      ans ← USERINPUT
6      IF ans = 'yes' THEN
7         OUTPUT 'Contact supplier'
8      ELSE
9         OUTPUT 'Plug it in and start again'
10     ENDIF
11  ELSE
12     OUTPUT 'Is it connected to the computer?'
13     ans ← USERINPUT
14     IF ans = 'yes' THEN
15        OUTPUT 'Contact supplier'
16     ELSE
17        OUTPUT 'Connect it to the computer'
18     ENDIF
19  ENDIF
```

0 3 . 1 Shade **one** lozenge to show which programming technique is used on line 3 of the algorithm in **Figure 8**.

[1 mark]

A Assignment ⬭

B Iteration ⬭

C Selection ⬭

0 3 . 2 Shade **one** lozenge to show the data type of the variable `ans` in the algorithm in **Figure 8**.

[1 mark]

A Date ⬭

B Integer ⬭

C Real ⬭

D String ⬭

0 3 . 3 Regardless of what the user inputs, the same number of `OUTPUT` instructions will always execute in the algorithm shown in **Figure 8**.

State how many `OUTPUT` instructions will execute whenever the algorithm is run.

[1 mark]

0 3 . 4 The phrase `'Contact supplier'` appears twice in the algorithm in **Figure 8**.

State the **two** possible sequences of user input that would result in `'Contact supplier'` being output.

[2 marks]

Sequence 1:

Sequence 2:

0 3 . 5 Another developer looks at the algorithm shown in **Figure 8** and makes the following statement.

> "At the moment if the user enters 'y' or 'n' they will sometimes get unexpected results. This problem could have been avoided."

Explain why this problem has occurred and describe what would happen if a user entered 'y' or 'n' instead of 'yes' or 'no'.

You may include references to line numbers in the algorithm where appropriate. You do **not** need to include any additional code in your answer.

[3 marks]

0 4 An application allows only two users to log in. Their usernames are stated in **Table 1** along with their passwords.

Table 1

username	password
`gower`	`9Fdg3`
`tuff`	`888rG`

Develop an algorithm, using either pseudo-code **or** a flowchart, that authenticates the user. The algorithm should:

- get the user to enter their username and password
- check that the combination of username and password is correct and, if so, output the string `'access granted'`
- get the user to keep re-entering their username and password until the combination is correct.

[6 marks]

0 5 Develop an algorithm, using either pseudo-code **or** a flowchart, that helps an ice cream seller in a hot country calculate how many ice creams they are likely to sell on a particular day. Your algorithm should:

- get the user to enter whether it is the weekend or a weekday
- get the user to enter the temperature forecast in degrees Celsius (they should enter a number between 20 and 45 inclusive; if the number falls outside of this range then they should be made to re-enter another number until they enter a valid temperature)
- calculate the number of ice creams that are likely to be sold using the following information:
  - 100 ice creams are likely to be sold if the temperature is between 20 and 30 degrees inclusive,
  - 150 ice creams are likely to be sold if the temperature is between 31 and 38 degrees inclusive,
  - and 120 ice creams are likely to be sold if the temperature is higher than 38 degrees
- double the estimate if it is a weekend
- output the estimated number of ice creams that are likely to be sold.

[9 marks]

0 6 Write a C# program to check if an email address has been entered correctly by a user.

Your program must:

- get the user to input an email address
- get the user to input the email address a second time
- output the message `Match` **and** output the email address if the email addresses entered are the same
- output the message `Do not match` if the email addresses entered are not the same.

You **should** use meaningful variable name(s) and C# syntax in your answer.

The answer grid below contains vertical lines to help you indent your code.

[5 marks]

0 7 Write a C# program that calculates the value of a bonus payment for an employee based on how many items they have sold and the number of years they have been employed.

The program should:

- get the user to input the number of items sold
- get the user to input the number of years employed
- output the value of the bonus payment:
  - if the years of employment is less than or equal to 2 **and** the number of items sold is greater than 100, then the bonus will be the number of items sold multiplied by 2
  - if the years of employment is greater than 2, then the bonus will be the number of items sold multiplied by 10
  - otherwise, the bonus is 0

You **should** use meaningful variable name(s) and C# syntax in your answer.

The answer grid below contains vertical lines to help you indent your code.

[7 marks]

0 8 A program is being developed in C# to simulate a card game.

Throughout the game each player always has 100 cards. Each card displays a number.

Players take it in turns to swap one of their cards with another random card from a set of cards until a player has a run of five numbers in sequence within their 100 cards.

0 8 . 1 **Figure 14** shows part of the program that will get a player to enter the position of a card to swap.

Figure 14

```
Console.Write("Enter card position: ");
int position = Convert.ToInt32(Console.ReadLine());
```

Extend the program in **Figure 14**. Your answer must be written in C#.

The program should keep getting the user to enter the card position until they enter a card position that is between 1 and 100 inclusive.

You **should** use meaningful variable name(s) and C# syntax in your answer.

The answer grid below contains vertical lines to help you indent your code.

[4 marks]

0 8 . 2 There are 500 cards within the game in total. Each card is numbered from 1 to 250 and each number appears twice in the whole set of cards.

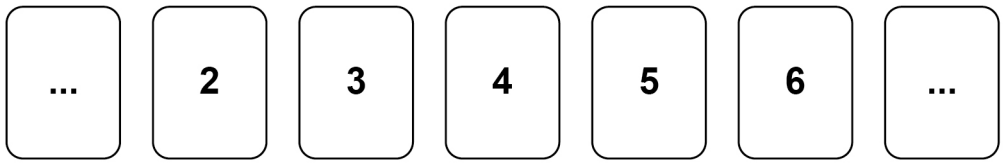
The player's 100 cards are always stored in numerical order.

When a player has a valid run of five cards within their 100 cards they have won the game.

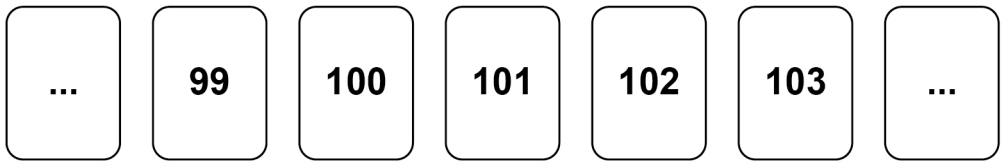
A valid run:
- consists of five cards
- can start from any position in the player's 100 cards
- the second card's value is one more than the first card's value, the third card's value is one more than the second card's value, the fourth card's value is one more than the third card's value, and the fifth card's value is one more than the fourth card's value.

Below are examples of valid runs which means a player has won.

Valid run example 1

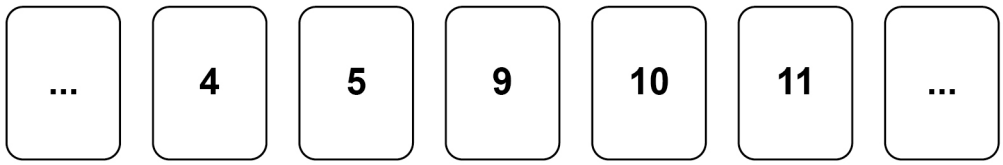


Valid run example 2

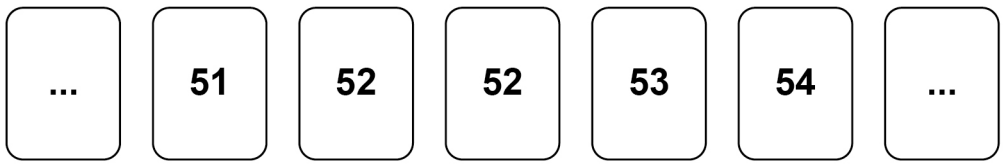


Below are examples of invalid runs.

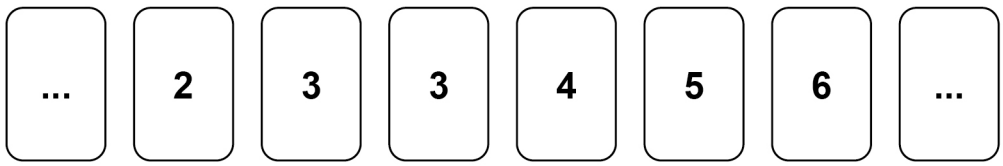
Invalid run example 1



Invalid run example 2



Invalid run example 3



Write a C# program to check if a player has a valid run of five cards within their 100 cards.

When writing your program you should assume:

- there is an array called `cards` that contains the values of the player's 100 cards
- `cards[0]` will contain the value of the first card and `cards[99]` will contain the value of the last card
- the values in `cards` are already stored in numerical order
- there is a Boolean variable called `gameWon` that has a value of `False`.

Your program should set `gameWon` to `True` if there is a valid run.

You **should** use meaningful variable name(s) and C# syntax in your answer.

The answer grid below contains vertical lines to help you indent your code.

[6 marks]

0 9 **Figure 2** shows an algorithm that uses integer division which has been represented using pseudo-code.

- Line numbers are included but are not part of the algorithm.

Figure 2

```
1   again ← True
2   WHILE again = True
3      a ← USERINPUT
4      IF a > 0 THEN
5         counter ← 0
6         WHILE a > 0
7            a ← a DIV 3
8            counter ← counter + 1
9         ENDWHILE
10     ELSE
11        again ← False
12     ENDIF
13     OUTPUT a
14  ENDWHILE
```

Integer division is the number of times one integer divides into another, with the remainder ignored.

For example:

- `14 DIV 5` evaluates to 2
- `25 DIV 3` evaluates to 8

0 9 . 1 Where is iteration **first** used in the algorithm in **Figure 2**?

Shade **one** lozenge.

[1 mark]

A Line number 2 ⬭

B Line number 4 ⬭

C Line number 6 ⬭

D Line number 11 ⬭

0 9 . 2 In the algorithm in **Figure 2**, what will be output when the user input is `10`?

Shade **one** lozenge.

[1 mark]

A `0` ⬭

B `1` ⬭

C `2` ⬭

D `4` ⬭

0 9 . 3 In the algorithm in **Figure 2**, what is the largest possible value of the variable `counter` when the user input is `36`?

Shade **one** lozenge.

[1 mark]

A `0` ⬭

B `2` ⬭

C `4` ⬭

D `5` ⬭

1 0 **Figure 5** shows an algorithm represented using pseudo-code.

The algorithm is for a simple authentication routine.

The pseudo-code uses a subroutine `getPassword` to check a username:

- If the username exists, the subroutine returns the password stored for that user.
- If the username does not exist, the subroutine returns an empty string.

Parts of the algorithm are missing and have been replaced with the labels L1 to L4.

Figure 5

```
login ← False
REPEAT
   username ← ''
   WHILE username = ''
      OUTPUT 'Enter username: '
      username ← L1
   ENDWHILE
   password ← ''
   WHILE password = ''
      OUTPUT 'Enter password: '
      password ← USERINPUT
   ENDWHILE
   storedPassword ← getPassword( L2 )
   IF storedPassword =  L3  THEN
      OUTPUT ' L4 '
   ELSE
      IF password = storedPassword THEN
         login ← True
      ELSE
         OUTPUT 'Try again.'
      ENDIF
   ENDIF
UNTIL login = True
OUTPUT 'You are now logged in.'
```

Figure 6

-1	OUTPUT	0
username	True	SUBROUTINE
1	User not found	''
USERINPUT	password	Wrong password

State the items from **Figure 6** that should be written in place of the labels in the algorithm in **Figure 5**.

You will not need to use all the items in **Figure 6**.

[4 marks]

L1 _____

L2 _____

L3 _____

L4 _____

Turn over for the next question

1 1 A theme park charges £15 per person for a daily ticket. If there are six or more people in a group, the group is given a £5 discount.

Write a C# program to calculate the total charge for a group of people visiting the theme park.

The program must:
- get the user to enter the number of people in a group
- calculate the total charge by:
  - charging £15 per person
  - reducing the total charge by £5 if there are six or more people
- output the total charge.

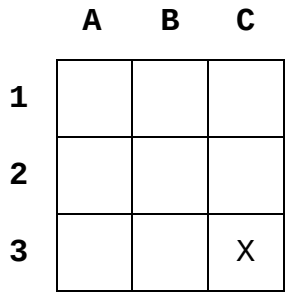
You **should** use meaningful variable name(s) and C# syntax in your answer.

The answer grid below contains vertical lines to help you indent your code.

[6 marks]

1 2 A programmer is writing a game. The game uses a 3 x 3 grid containing nine squares.

Figure 14



In the game, a square on the grid is referred to by a letter and a number.
For example, square **C3** in **Figure 14** contains an X.

Figure 15 shows part of a C# program that checks the grid reference entered by a player.
The grid reference is valid if:

- there are exactly two characters
- the first character entered is A, B or C
- the second character entered is 1, 2 or 3.

Figure 15

```
bool check = false;
while (check == false) {
   string square = "";
   while (square.Length != 2) {
      Console.Write("Enter grid reference (eg C2): ");
      square = Console.ReadLine();
      square = square.ToUpper();
   }
```

The C# function `ToUpper()` converts letters into uppercase, eg `b1` would be converted to `B1`

Extend the program from **Figure 15** so it completes the other checks needed to make sure a valid grid reference is entered.

Your extended program must:

- use the variable `check`
- repeat the following steps until a valid grid reference is entered:
  - get the user to enter a grid reference
  - output an appropriate message if the grid reference entered is not valid.

You **should** use meaningful variable name(s) and C# syntax in your answer.

The answer grid contains vertical lines to help you indent your code.

[6 marks]

```
bool check = false;
while (check == false) {
    string square = "";
    while (square.Length != 2) {
        Console.Write("Enter grid reference (eg C2): ");
        square = Console.ReadLine();
        square = square.ToUpper();
    }
}
```

1 3 A group of people have a meal in a restaurant. Instead of one person paying for the whole meal, each person will pay for what they eat.

Write a C# program that asks each person in the group how much they are paying towards the meal and works out when the bill is fully paid. Each person can pay a different amount.

The program should:

- get the user to enter the total amount of the bill
- get a person to enter how much they are paying towards the bill
- subtract the amount entered from the bill:
  - if the amount left to pay is more than `0`, output how much is left to pay and repeat until the amount left to pay is `0` or less
  - if the amount left to pay is `0`, then output the message `Bill paid`
  - if the amount left to pay is less than `0`, then output the message `Tip is` and the difference between the amount left to pay and `0`

You **should** use meaningful variable name(s) and C# syntax in your answer.

The answer grid below contains vertical lines to help you indent your code.

[8 marks]

1 4 Question **14** is about a dice game played against a computer.

The aim of the game is to get as close to a score of 21 as you can, without going over 21. If your score goes over 21 then you lose.

The player's score starts at 0.

For each turn:

- two dice (each numbered from 1 to 6) are rolled
- the total of the two dice rolls is added to the player's score
- the value of each dice and the player's new total score is output
- if the current score is less than 21, the player is asked if they would like to roll the dice again: if the player says yes, they get another turn; otherwise, the game ends.

At the end of the game, the program should work as follows:

- if the final score is 21, output a message to say the player has won
- if the final score is greater than 21, output a message to say the player has lost
- if the final score is less than 21, the program generates a random number between 15 and 21 inclusive:
  - if this random number is greater than the player's final score, output a message to say the player has lost
  - otherwise, output a message to say the player has won.

Figure 17 shows the output of a program that plays this dice game.

Figure 17

```
Roll 1: 1
Roll 2: 4
Current score: 5
Would you like to roll again? yes

Roll 1: 1
Roll 2: 6
Current score: 12
Would you like to roll again? yes

Roll 1: 1
Roll 2: 2
Current score: 15
Would you like to roll again? yes

Roll 1: 6
Roll 2: 1
Current score: 22
You lost!
```

Write a C# program to simulate this game.

The first line has been written for you in the answer grid.

The dice rolls are carried out by the program generating random numbers between 1 and 6. You will need to use the C# function `r.Next(a, b)` which generates a random integer in the range `a` to `b` starting at `a` but finishing one before `b`.

You **should** use meaningful variable name(s) and C# syntax in your answer.

The answer grid below contains vertical lines to help you indent your code.

[11 marks]

```
Random r = new Random();
```


1 5 Write a C# program that allows a taxi company to calculate how much a taxi fare should be.

The program should:

- allow the user to enter the journey distance in kilometres (no validation is required)
- allow the user to enter the number of passengers (no validation is required)
- calculate the taxi fare by
  - charging £2 for every passenger regardless of the distance
  - charging a further £1.50 for every kilometre regardless of how many passengers there are
- output the final taxi fare.

You **should** use meaningful variable name(s), correct syntax and indentation in your answer.

The answer grid below contains vertical lines to help you indent your code accurately.

[7 marks]

1 6 Write a C# program that inputs a password and checks if it is correct.

Your program should work as follows:

- input a password and store it in a suitable variable
- if the password entered is equal to `secret` display the message `Welcome`
- if the password entered is not equal to `secret` display the message `Not welcome`.

You **should** use meaningful variable name(s), correct syntax and indentation in your answer.

The answer grid below contains vertical lines to help you indent your code accurately.

[5 marks]

1 7 Write a C# program that inputs a character and checks to see if it is lowercase or not.

Your program should work as follows:

- gets the user to enter a character and store it in a suitable variable
- determines if the entered character is a lowercase character
- outputs `LOWER` if the user has entered a lowercase character
- outputs `NOT LOWER` if the user has entered any other character.

You **should** use meaningful variable name(s), correct syntax and indentation in your answer.

The answer grid below contains vertical lines to help you indent your code accurately.

[7 marks]

1 8 Write a C# program that calculates an estimate of the braking distance in metres for a new model of go-kart that is travelling between `10` and `50` kilometres per hour (kph).

Your program should:

- keep asking the user to enter a speed for the go-kart until they enter a speed that is between `10` and `50` (inclusive)
- calculate the braking distance in metres by dividing the speed by 5
- ask the user if the ground is wet (expect the user to enter `yes` if it is)
- if the ground is wet, multiply the braking distance by 1.5
- output the final calculated braking distance.

You **should** use meaningful variable name(s), correct syntax and indentation in your answer.

The answer grid below contains vertical lines to help you indent you code accurately.

[8 marks]

1 9 A university is writing a program to calculate a student's total mark for three essays.

If any essays are handed in late, the total mark is reduced.

Write a C# program to calculate the total mark.

You should assume there are three integer variables called `e1`, `e2` and `e3` which have already been given values to represent the marks of the three essays.

The program should:

- get the user to enter the number of essays handed in late and store the number in a variable
- calculate the total mark for the three essays
  - if only one essay is handed in late, the total mark is reduced by 10
  - if more than one essay is handed in late, the total mark should be halved
  - the total mark should **not** be less than 0
- output the total mark.

You **should** use meaningful variable name(s) and C# syntax in your answer.

The answer grid below contains vertical lines to help you indent your code**.**

[7 marks]

Turn over for the next question

2 0 A shop owner wants to create stock codes for each type of sweet they sell.

Figure 5 shows some of the sweets.

Figure 5

sweetID	sweetName	brand
S1	WINE GUMS	MAYNARDS
S2	COLA CUBES	BERRYMANS
S3	STARBURST	WRIGLEY

A stock code is made up of the:
- `sweetID`
- first letter and the second letter in `sweetName`
- first letter of the `brand`

For example:
- the stock code for `WINE GUMS` would be `S1WIM`
- the stock code for `STARBURST` would be `S3STW`

Write a C# program to create the stock code for a sweet.

The program should:
- get the user to enter the `sweetID`, `sweetName` and `brand`
- create the stock code
- assign the stock code to a variable called `code`

You **should** use meaningful variable name(s) and C# syntax in your answer.

The answer grid below contains vertical lines to help you indent your code.

[4 marks]

Turn over for the next question

2 1 . 1 Which of the following best describes a **data structure**?

Shade **one** lozenge.

[1 mark]

A A number with a fractional part ⬭

B A value such as a whole number ⬭

C All of the data used and stored within a program ⬭

D An organised collection of values ⬭

2 1 . 2 **Figure 7** shows an **incomplete** algorithm, represented using pseudo-code.

The algorithm is used to store and manage books using records.

The algorithm should do the following:

- create a record definition called `Book` with the fields `bookName`, `author` and `price`
- create a variable for each book using the record definition.

Complete **Figure 7** by filling in the gaps using the items in **Table 2**.

- You may need to use some of the items in **Table 2** more than once.
- You will **not** need to use all the items in **Table 2**.

[3 marks]

Table 2

1	2	author
B1	B2	Book
bookName	i	Real
OUTPUT	String	Boolean

Figure 7

```
RECORD _____________

   bookName : String

   _____________ : String

   price : _____________

ENDRECORD

B1 ← Book("The Book Thief", "M Zusak", 9.99)

B2 ← _____________ ("Divergent", "V Roth", 6.55)
```

2 1 . 3 Write an algorithm using pseudo-code to display the name of the most expensive book.

The algorithm should:

- compare the price of `B1` and the price of `B2`
- output the book name of the most expensive book
- output `Neither` if the books are the same price.

The algorithm should work for any values stored in `B1` and `B2`

[3 marks]

Turn over for the next question

2 2 A program is to be written to authenticate a username and password entered by the user.

Figure 9 shows the only two pairs of valid usernames and passwords.

Figure 9

Username	Password
Yusuf5	33kk
Mary80	af5r

Write a C# program to authenticate a username and password.

The program should:
- get the user to enter a username
- get the user to enter a password
- display the message `Access denied` if the username and password pair entered is not valid
- display the message `Access granted` if the username and password pair entered is valid
- repeat until a valid username and password pair is entered.

You **should** use meaningful variable name(s) and C# syntax in your answer.

The answer grid below contains vertical lines to help you indent your code accurately.

[7 marks]

2 3 A programmer is writing a game.

The game uses a row of cells represented as an array. **Figure 20** shows an example.

Figure 20

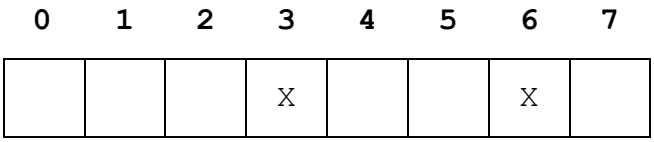


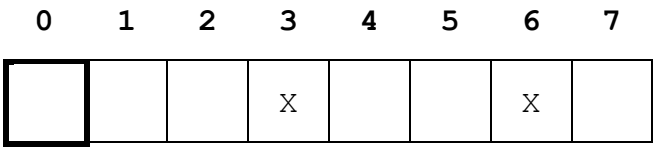
Figure 21 describes how the game is to be played.

Figure 21

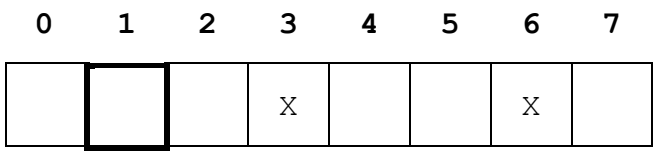
- The player starts at position `0` in a row of cells.
- The aim of the game is for the player to reach the end of the row.
- At each turn the player must enter either `1` or `2`
  - if the player enters `1`, the player's position increases by 1
  - if the player enters `2`, the player's position increases by 2
- If the player's position goes beyond the end of the row or contains an `X`:
  - the message `Bad move` is displayed
  - the player goes back to position `0`
- These steps are repeated until the player reaches the end of the row.
- If the player reaches the end of the row the game is finished.

For example, using the array in **Figure 20**:

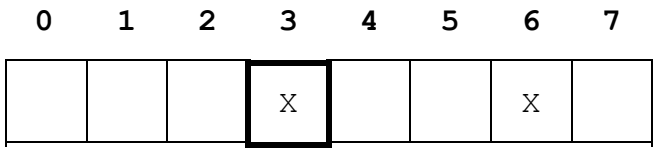
- the player starts in position `0`



- if the player enters a `1`, then they move to position `1`

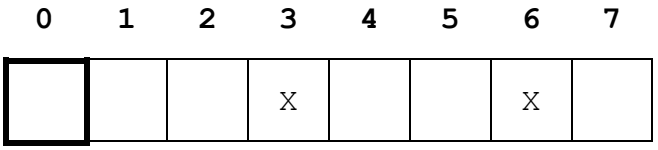


- if the player then enters a `2`, `Bad Move` is displayed as position `3` contains an `X`

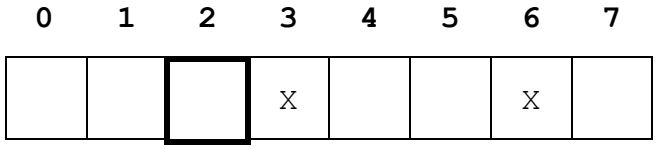



```
Bad move
```

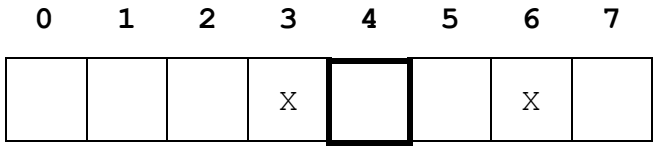
- the player then goes back to position `0`



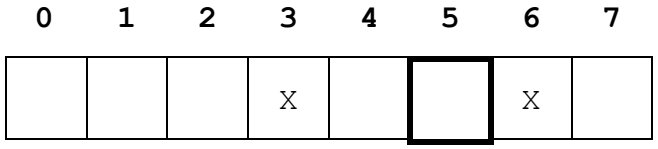
- if the player then enters a `2`, they move to position `2`



- if the player then enters a `2`, they move to position `4`



- if the player then enters a `1`, they move to position `5`



- if the player then enters a `2`, the game finishes.

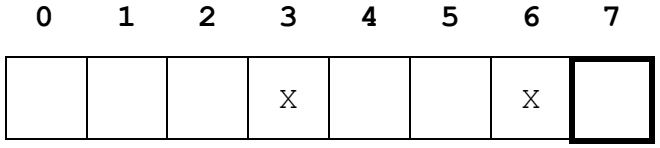


Figure 22 shows part of a C# program that will be used for the game.

Figure 22

```
int pos = 0;
int lastPos = row.Length - 1;
while (pos < lastPos)
{
```

`pos` is a variable that contains the player's current position.

Extend the program from **Figure 22** so that the game works as described in **Figure 21**, on page 37.

When writing your program you should assume:

- there is an array called `row`
- the number of `X` characters in `row` can vary
- the position of the `X` characters in `row` can vary
- the `X` characters have already been added to the array called `row`
- the `row` array can be of any length.

You **should** use meaningful variable name(s) and C# syntax in your answer.

The answer grid below contains vertical lines to help you indent your code.

[8 marks]

```
int pos = 0;
int lastPos = row.Length - 1;
while (pos < lastPos)
{
```

}				