



# Computational Thinking

A set of exercises to support learners preparing for GCSE Computer Science exams.



**arm** School Program

## Task 1

```
loopCounter = const(9)
for x = 0 to loopCounter
    repeat
        value = input("Enter a number between 1 and 10
                       inclusive")
    until value > 0 and value < 11
    print(calculate(value))
next x
```

- a) Identify **two** variables used in the program
- b) Identify **one** constant used in the program
- c) Identify the lines where there is a sequence
- d) Identify **two** lines where there is an iteration statement
- e) Identify the line where there is a function call
- f) Change the program so it loops 20 times instead of 10

## Task 2

```
function calculateValues(num, value, word)
    if int(value) < num then
        return True
    elseif str(value) != word and word.length() < 10 then
        return True
    else
        return False
    endif
endfunction
```

- a) How many parameters does the function take?
- b) What data type is the value returned?
- c) Identify the lines where there is a selection statement?
- d) Identify the Boolean comparison used in the selection statement?
- e) Why is this a function and not a procedure?

### Task 3

```
function unknown(theData, data1)
    first = 0
    last = theData.length()-1
    while first <= last
        mid = (first + last) // 2
        if theData[mid] == data1 then
            return mid
        elseif theData[mid] < data1 then
            first = mid + 1
        else
            last = mid - 1
        endif
    endwhile
    return -1
endfunction
```

a) Complete the trace table for the algorithm when the following data is sent as parameters:

theData:

1	5	8	10	15
---	---	---	----	----

data1: 20

theData					data1	first	last	mid	return
1	5	8	10	15	2				
						0	4	2	
						3		3	
						4		4	
									-1

b) Why is this a function and not a procedure.

c) Identify **three** variables in the program

d) If theData stores what is in part (a), what will theData[1] return?

e) What algorithm does this perform.

## Task 4

FILMS					
FilmID	FilmName	Genre	Age	YearRelease	Rating
012MX	The Matrix	Sci-fi	15	1999	8
952ST	Star Trek Into Darkness	Sci-fi	12	2016	9
512SM	There's Something About Mary	Comedy	15	1998	5
195JJ	Jumanji: Welcome to the Jungle	Adventure	12	2017	6
215MI	Mission: Impossible – Fallout	Adventure	12	2018	3

a) What will the following SQL program return?

```
SELECT FilmName
FROM FILMS
WHERE Rating >= 5
```

b) What will the following SQL program return?

```
SELECT FilmID, FilmName
FROM FILMS
WHERE GENRE = "Sci-fi" AND YearRelease >= 2000
```

c) Highlight the error in this SQL program.

```
SELECT FilmName FilmID
FROM FILMS
WHERE Age > 12
```

d) The program should display the genre and rating of all films that are below age 15. Complete the code.

```
SELECT Genre, Rating
FROM FILMS
WHERE AGE < 15
```

e) The program should output the names of all films, along with the rating, that were release in either 2000, 2005, 2010 and/or 2015. Put the code in order.

```
SELECT FilmName, Rating
FROM FILMS
WHERE YearRelease = 2000 OR YearRelease = 2005 OR YearRelease = 2010 OR
YearRelease = 2015
```

## Task 5

```
01  mainMeal = 25.99
02  dessert = 7.50
03  drinks = 2.50
04  mainPeople = input("How many people require a main meal")
05  repeat //number of deserts cannot be more than the number of 06
//people for main meals. Loop until this is true
07      dessertPeople = input("How many require a dessert")
08  until dessertPeople <= mainPeople
09  total = ((mainMeal + drinks) * mainPeople) + (dessertPeople * dessert) //
calculate total cost
10  print("The total cost is " + str(total))
```

- a) Identify **one** line where there is a comment.
- b) Identify **one** line where there is an iteration comparison.
- c) Identify the line where there is concatenation.
- d) The program needs to be changed to include a starter for 5.50. Which of these would allow this to be added to the program.
- e) The program needs changing so there is a 10% tip added. How does the program need to be changed to allow for the tip to be added to the price before it is output.

## Context driven tasks

A system has been developed to automatically generate usernames for new students entering school into Year 7. The username is calculated by taking the last two digits of the year of entry followed by the first four digits of a student's surname.

### Question 1

Using the information above what would be the username for Amy Jones entering school in 2019?

### Question 2

The code to generate the username is:

```
username=(yearOfEntry(0,3)+surname(0,4)).
```

What is the name of the operation which the '+' symbol performs on two strings?

### Question 3

Once a username has been defined the student sets their password when they first log in. Their password is stored as userPassword.

The code below should ask the user to enter their password. What command is missing?

```
userPassword=_____("Please enter your password")
```

### Question 4

If the user enters their password they should be prompted to try again. The code below contains a bug as it just displays "Incorrect password" in an infinite loop.

```
while enteredPassword != userPassword:  
    print ("You have entered the incorrect password")
```

How could this infinite loop be resolved?

Use a count controlled loop and set the count to once only.

User an IF statement instead of a While statement.

Add an extra line of code to ask for an input after the print statement.

### Question 5

Once the user has successfully set up their account it is necessary to save their details in an external file. Fill in the blanks so that the username and password is saved.

```
userDetails = openWrite("accounts.txt")  
userDetails.writeLine(username,userpassword)  
userDetails.close()
```



## Zoo Animals

A zoo has a database of animals. An extract from the database table 'tblanimals' is detailed below.

Animal	Type	Quantity
Elephant	Mammal	6
Giraffe	Mammal	9
Lion	Mammal	9
Leopard	Mammal	4
Ostrich	Bird	24
African Grey	Bird	2
Buzzard	Bird	4

SQL is used to query the database.

### Question 1

What would the following SQL statement return?

```
SELECT * FROM 'tblanimals' WHERE quantity > 9;
```

### Question 2

SQL commands sometimes include a 'wildcard'. Describe what a wildcard is.

```
SELECT % FROM tblanimals WHERE Type = 'Mammal';
```

### Question 3

Write a SQL statement which will return only the Animal field where the type is Bird.

### Question 4

Write a SQL statement which will return all animals and the quantity where the animal starts with the letter l.

```
SELECT Animal AND Quantity FROM tblanimals WHERE Animal LIKE 'l';
```

### Question 5

What will the following SQL statement return?

```
SELECT Animal FROM tblanimals WHERE Quantity > 5 AND NOT Type = 'Mammal';
```

## Football league

An amateur football league in Trowbridge requires a system to calculate how many league points they achieve. If one team scores more goals than their opposition they will win the game and receive three points. If both teams score the same number of points they will draw the game and receive one point each.

### Question 1

Using the information above, write out the league table positions using the results below:

Trowbridge 1 – Chippenham 3

Melksham 2 – Bath 1

Bradford 2 – Bradley Road 5

Warminster 0 – Warmley 0

### Question 2

Identify the most appropriate data type for the homeTeamName, awayTeamName, homeTeamGoals and awayTeamGoals.

Choose from string, character, float or integer.

### Question 3

Identify the most appropriate data structure to store the league table.

String. 1D array. 2D array. 3D array.

### Question 4

A statement is required to determine whether a team should be awarded 3, 1 or 0 points. What is the name of this type of statement?

Sequence      Selection      Iteration

## Test Results

Students have sat four tests. Each test is worth 25% of the overall total. Results are detailed in the table below and are all out of 100.

	Test Number			
Student	0	1	2	3
Bob	75	65	42	73
Helen	62	65	67	79
Vic	13	17	19	17

The data is stored in a zero indexed 2D array named testResults.

```
testResults=[["Bob", 75, 65, 42, 73], ["Helen", 62, 65, 67, 79], ["Vic", 13, 17, 19, 17]]
```

### Question 1

What would the following code output?

```
print(testResults[0])
```

### Question 2

What would the following code output?

```
print(testResults[0][0])
```

### Question 3

Arrange the code below so that the user can enter the name of the student and the prints out their test scores.

```
name=input("Enter the student name")
if name=="Bob" then
student=0
elseif name=="Helen" then
student = 1
elseif name=="Vic" then
student = 2
endif
for x = 0 to 3
print (testResults[student][x])
```

#### Question 4

A function has been designed to calculate the grade that a student achieves. It takes in the average result and returns the grade. If the student achieves 80% or above they are awarded an A, 60% to 79% a B, 40% to 59% a C or below 40% an Unclassified.

Identify the function identifier and parameters from the code below.

```
function gradeCalculator(average)
```

#### Question 5

Arrange the code below to return the grade.

```
function gradeCalculator (average)
switch average
case 80:
return="A"
case 60:
return = "B"
case 40:
return = "C"
case default:
return = "Unclassified"
end switch
end function
```

## Ten pin bowling

Violet has produced a table to record the scores from a game of ten pin bowling. There are five players and ten rounds in a game. She is storing the data in a 2 dimensional array named `bowlingScore`.

	Player			
Round	0	1	2	3
0	3	5	9	13
1	5	8	4	1
2	20	9	13	14
3	2	3	2	3
4	6	7	7	6
5	11	15	4	3
6	15	14	9	11
7	20	17	14	17
8	1	2	17	4
9	0	19	4	2
10	11	7	16	9

### Question 1

State the output if Violet runs the code:

```
print(bowlingScore[2,0])
```

### Question 2

Write the code to output the score Player 1 achieved in round 2.

```
print(bowlingScore[1,2])
```

### Question 3

State the output if Violet runs the code:

```
print(bowlingScore[0,0]+ bowlingScore[1,0]+ bowlingScore[2,0]+ bowlingScore[0,3])
```

### Question 4

Arrange the code below to produce a program which outputs the total score for player 0.

```
score = 0
for x = 0 to 9
score = score+bowlingScore[0,x]
next x
print(score)
```

### Question 5

Arrange the code below to produce a function which returns the total number of points scored by all players in the game.

```
function pointsScored
score = 0
for x = 0 to 3
for y = 0 to 9
score = score+bowling score[x,y]
return score
```

## Arm Theatre

Arm Theatres are opening a new show in their theatre in Cambridge. They are developing an online booking system, so people can select a performance, and then see which seats are available and book as many seats as they need to.

The current performance dates are:

Date	Time
18 <sup>th</sup> June	Evening
19 <sup>th</sup> June	Matinee
19 <sup>th</sup> June	Evening
21 <sup>st</sup> June	Evening
22 <sup>nd</sup> June	Evening

These are the seats in the stalls.

1	2	3	4	5	6	7	8	9	10	11	12	13	H	14	15	16	17	18	19	20	21	22	23	24	25
1	2	3	4	5	6	7	8	9	10	11	12	13	G	14	15	16	17	18	19	20	21	22	23	24	
1	2	3	4	5	6	7	8	9	10	11	12	13	F	14	15	16	17	18	19	20	21	22	23		
	1	2	3	4	5	6	7	8	9	10	11		E	14	15	16	17	18	19	20	21	22			
		1	2	3	4	5	6	7	8	9			D	14	15	16	17	18	19	20	21				
			1	2	3	4	5	6	7	8	9		C	14	15	16	17	18	19	20	21				
				1	2	3	4	5	6	7	8	9	B	14	15	16	17	18	19	20	21				
					1	2	3	4	5	6	7	8	A	14	15	16	17	18	19	20	21				

### Question 1

What is the most appropriate data type for each piece of data that will be stored in the program?

## Questions 2

This part of the program asks the user to select a performance.

Complete the gaps in the program indicated by red question marks (?)

```
//performances are stored in a 2D array. Element 0 stores the date, element 1
stores the time.

print("Enter the date from ")

for x = 0 to ? //loop through all 5 performances
    print("Enter " & x & " for:") //output what to enter
    //output the performance date and time for each performance
    print(performanceDates[x][.....] & " " & performanceDates[x][1])
next ?

//take the user's input
? = input()

while performance ? or performance != 1 or performance != 2 or performance != 3
or performance != 4
    ?("Invalid, try again")
    ? = input()
endwhile

print("You selected " & performanceDates[?][0] & " " ?
performanceDates[performance][?])
```



### Question 3

This procedure should search the array that stores the seats and display all the seat numbers for each row, along with whether they are available or not. Those that are available (the array element is True) as "A" and those unavailable (the array element is False) as "N". Highlight the errors in the program.

```
procedure display(seatArray)
    for row == 1 to 7
        if row = 0 then
            print("Row A")
        elseif row == 1 then
            print("Row B")
        elseif row == 2 then
            print("Row C")
        elseif row == 3 then
            print("Row D")
        elseif row == 4 then
            print("Row E")
        elseif row == 5 then
            print("Row F")
        else
            print("Row I")
        endif
    for seat = 0 to 23
        print("Seat number " & seat " ")
        if seatArra[row][seat] == Yes then
            print("A")
        else
            print("N")
        endif
    next seat
next row
endprocedure
```

#### Question 4

Select the function that takes the seat selected as a parameter along with the relevant seat array, and then books the seat and returns true if the seat was free, and returns false if it was already booked

##### Function 1

```
function book(seat, seatArray)
    if seatArray[seat] == "True" then
        book = False
    else
        seatArray[seat] = "True"
        book = True
    endif
endfunction
```

##### Function 2

```
function book(seatArray)
    if seatArray[seat] == "True" then
        book = False
    else
        seatArray[seat] = "True"
        book = True
    endif
endfunction
```

##### Function 3

```
function book(seat, seatArray)
    if seatArray[seat] == "True" then
        book = True
    else
        seatArray[seat] = "True"
        book = False
    endif
endfunction
```

### Question 5

Which of the following best describes how abstraction been applied to the seat layout?

- Removed details of theatre, seats etc and just represented them as boxes
- Showed all of the seats
- Split the rows into individual seats
- Split the problem into subproblems that are easily solvable

## Take Two

Xander is creating a computerised version of the card game 'Take Two'. There is a 52-deck of cards. A card has a suit (hearts, diamonds, spades or clubs) and a number (Ace, 1, 2, 3, 4, 5, 6, 7, 8, 9, Jack, Queen, King). The 52 cards are shuffled and put in a random order at the start of the game.

The game has up to three players. Each are given 7 cards at the start, this is the player's 'hand'. The remainder of the cards are face down in 'the deck'. Player 1 goes first.

The first card in the deck is turned over. Player 1 has to play either a) a card of the same suit, b) a card of the same number but a different suit. If at any point a player cannot play a valid card, they pick up the next card from the deck. It is then player 2's turn.

There are different rules depending on which cards have been played. If a player puts down:

- a '2' e.g. 2 of Hearts, then the next player has to take 2 cards from the deck and they miss their turn.
- a 'Queen' e.g. Queen of spades, then they get another go.
- a 'King' e.g. King of Diamonds, then the next person misses a go.
- an 'Ace' e.g. Ace of clubs, then they can change the suit. Any ace can be played at any time.

The aim of the game is to get rid of all your cards.

### Question 1

What is the most appropriate data structure to store the deck of cards, both the suit and number are stored as string data types?

### Question 2

What happens if a player plays a Queen, and then can't play another card?

### Question 3

The following code starts the game. It shuffles the deck using the function `shuffle()` that takes an array as a parameter and returns it in a random order. It asks how many players there are and deals each player 7 cards. Complete the comments to describe what the code is doing.

```
repeat
    numPlayers = input("How many players")
until numPlayers > 0 and < 4 //
theDeck = shuffle(theDeck) //
topDeck = 0 //
for cards = 0 to 7 //
    for players = 1 to numPlayers //
        //
        if players == 1 then
            player1[cards][0] = theDeck[cards][topDeck]
            player1[cards][1] = theDeck[cards][topDeck]
        elseif players == 2 then
            player2[cards][0] = theDeck[cards][topDeck]
            player2[cards][1] = theDeck[cards][topDeck]
        else
            player3[cards][0] = theDeck[cards][topDeck]
            player3[cards][1] = theDeck[cards][topDeck]
        endif
        //
        theDeck[topDeck][0] = "null"
        theDeck[topDeck][1] = "null"
        topDeck = topDeck + 1 //
    next players
next cards
```

#### Question 4

This function takes two cards as parameters (the previous card and the card played) and returns True if it is a valid move, and False if not. Complete the procedure where the red questions marks appear (?).

```
function checkValid(lastCard, cardPlayed)
    suit = lastCard[0]
    number = lastCard[1]
    if suit == cardPlayed[0] then
        return ?
    elseif number == cardPlayed[1] then
        return ?
    elseif number == "Ace" then
        return ?
    else
        return ?
    endif
endfunction
```

### Question 5

A procedure sorts a player's hand into ascending numerical order, the player's hand is sent as a parameter and returned when sorted. The function `convert()` is used to take a card number as a parameter and return its integer value e.g. Ace would return 0, 1 would return 1, King would return 13.

Complete the procedure where the red questions marks appear (?).

```
procedure sortHand(hand)
  for count = 1 to hand.length()-1
    ? = hand[count]
    temp2 = count - 1
    while temp2 >= 0 and convert(hand[temp2][1]) > convert(temp[1])
      hand[temp2 + 1] = ?
      temp2 = ?
    endwhile
    hand[temp2 + 1] = ?
  next count
endprocedure
```

## Board Game

Cora is creating a computerised board game. There is a path (in grey) that the players must follow. The players start in the bottom left hand corner, in the square 'S', and need to get to square 'F'.

There are two different sets of cards that players may need to pick from. Treat cards give players a boost i.e. they get to move forward a set amount. Back cards make players move backwards a set number of spaces. A player picks up a card when they land on a space with the corresponding letter on e.g. 'T' means pick a treat card. After the player has moved from a treat or back card, they do not need to pick up another card if the square they land on requires it.

F	B	B	B			B		T		B			B	
													B	
B					B	T	B				B		B	
B														
B							B						B	
B					B			B	B			T		B
		B				T		B			B		B	
													T	
B					B			B	B	B			B	
B			B				B	T			B		B	
													B	
S			B	T			B					B	B	

There are two players, and they take it in turns to roll two 6-sided dice. If they roll a double, i.e. two 2s, then they move and then roll again without picking up the card they land on first time; for example they roll two 2s, move forward 4 spaces and land on a treat card space, they do not pick up the card, they roll again and move according to the numbers rolled, if this space has a card then it must be picked up.

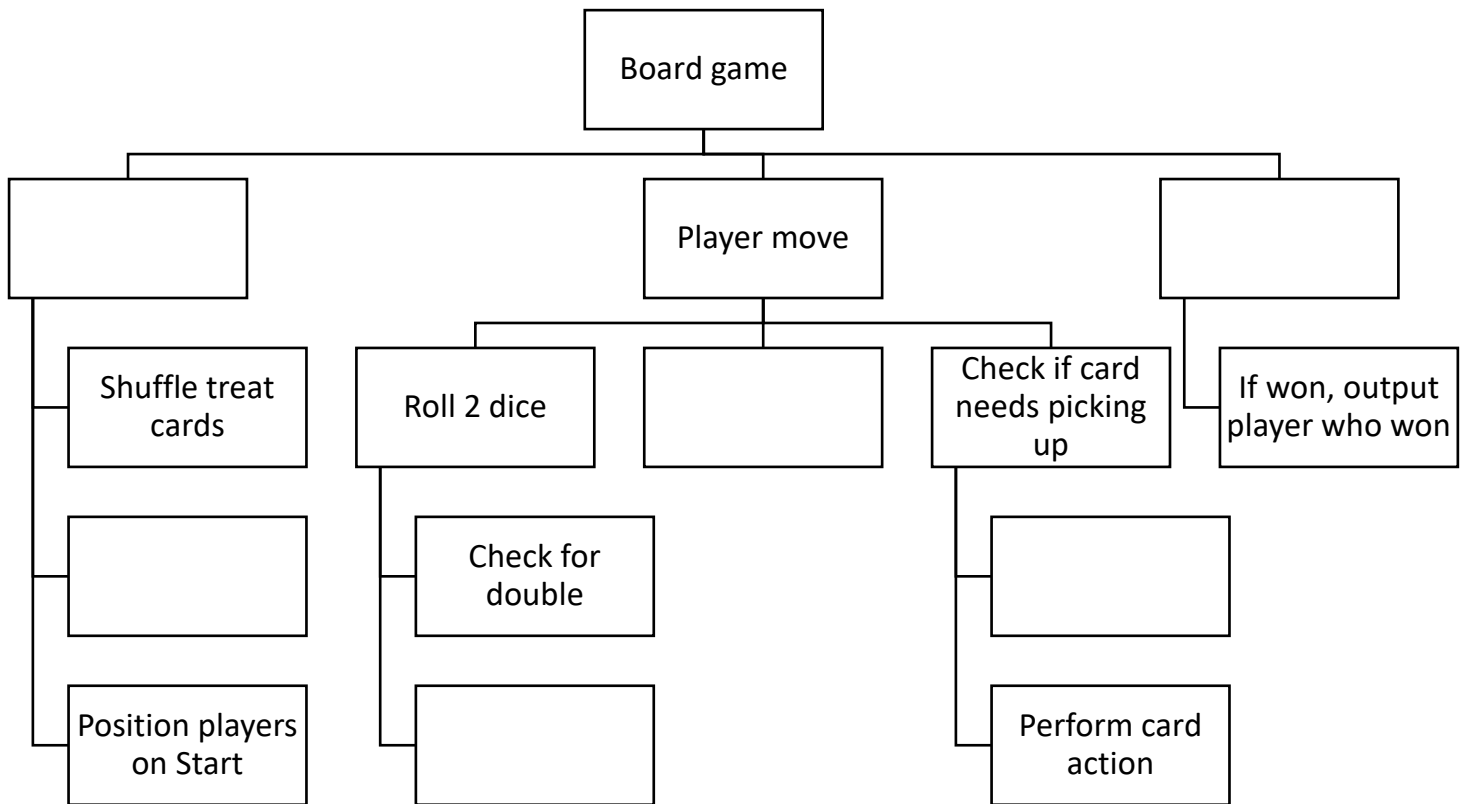
Players do not have to land exactly on F to finish, for example if they are four spaces away and they roll 11, they will still get to Finish. The first player to get to the square F wins the game.

The program uses the function `roll()` to generate and return a random number between 1 and 6 inclusive.



### Question 1

The program is split into subprograms and the following structure diagram is created. Fill in the missing boxes.



## Question 2

This function rolls the dice for a player, checks if they rolled a double and calculates the number of spaces the player should move in total. This value is returned.

Highlight the **three** errors in the code.

```
function move()  
    dice1 = roll()  
    dice = roll()  
    total = dice1 + dice2  
    print("You rolled a " + str(dice1) + " and a " + str(dice2))  
    while dice1 == dice2  
        print("That's a double. Roll again")  
        dice1 = roll()  
        dice2 = roll()  
        print("You rolled a " + str(dice1) + " and a " + str(dice2))  
        total = total + dice1 + dice2  
    endwhile  
    return neTotal  
endfunction
```

### Question 3

Match the function call to the function `move`.

```
function checkWon(theBoard[], playerPosition, spaces)
    playerPosition = playerPosition + spaces
    if theBoard[playerPosition] == "F" then
        return -9999
    else
        return playerPosition
    endif
endfunction
```

a)

```
player1Roll = move()
player1Position = checkWon(theBoard[], player1Position, player1Roll)
```

b)

```
player1Roll = move()
player1Position = checkWon(theBoard[], player1Roll, player1Position)
```

c)

```
player1Roll = move()
player1Position = checkWon(player1Roll, player1Position)
```

d)

```
player1Roll = move()
player1Position = checkWon(player1Position, player1Roll)
```

## Question 4

The function `card` takes the player's new position, checks if there is a card, if there are outputs the contents of the next card and the new position, before returning the new position. The deck of treat and back cards are stored as global variables.

```
function card(theBoard[], playerPosition)
    if theBoard[playerPosition] == "T" then
        print("Treat!  Move forward ")
        newMove = treatCards[treatTopCard]
        print(str(newMove) & " spaces")
        treatTopCard = treatTopCard + 1
        playerPosition = playerPosition + newMove
    elseif theBoard[playerPosition] == "B" then
        print("Back!  Move back ")
        newMove = backCards[backTopCard]
        print(str(newMove) & " spaces")
        backTopCard = backTopCard + 1
        playerPosition = playerPosition - newMove
    endif
    return (playerPosition)
endfunction
```

### Question 5

Identify the problem with using global variables to store the treat and back cards.

- a) The memory they take up is used throughout the program
- b) They can only be accessed where they are declared
- c) It doesn't allow you to change the data in them
- d) It will be overwritten every time you refer to them

## Encryption

Encryption is the jumbling up of messages so that they can be read, but not understood, if they are intercepted and someone does not know the algorithm for jumbling them up.

A message is being encrypted by the ASCII value of each letter of the message being increased. The first character is increased by 1, the second by 2, the third by 3 etc., until 10 is reached. The 10th character is increased by 10, then the 11th is increased by 1 again. This repeats to the end of the message.

For example, the message This is me! would become Uj!w%oz(vo"

Character	T	h	i	s		i	s		m	e	!
ASCII value	84	104	105	115	32	105	115	32	109	101	33
New ASCII value	85	106	108	119	37	111	122	40	118	111	34
New character	U	j	l	w	%	o	z	(	v	o	"

The encryption function should a message as a parameter and return the encrypted message.

```

01  function encrypt(message)
02      messageLength = message.length()
03      addition = 1
04      newMessage = ""
05      for count = 0 to messageLength-1
06          ASCIIValue = ASC(message.substring(count,1))
07          ASCIIValue = ASCIIValue + addition
08          newMessage = newMessage + CHR(ASCIIValue)
09          if addition = 11 then
10              addition = 1
11          else
12              addition = addition + 1
13          endif
14      next count
15  endfunction

```

## Question 1

What is the putpouse of line 4?

## Question 2

The function encrypt does not currently work. Add in the missing command in the appropriate place.

```
function encrypt(message)

    messageLength = message.length()

    addition = 1

    newMessage = ""

    for count = 0 to messageLength-1

        ASCIIValue = ASC(message.substring(count,1))

        ASCIIValue = ASCIIValue + addition

        newMessage = newMessage + CHR(ASCIIValue)

        if addition = 11 then

            addition = 1

        else

            addition = addition + 1

        endif

    next count

endfunction
```

### Question 3

Which function call correctly sends a message to be encrypted and outputs the encrypted message. Explain why.

```
print(encrypt("Hello World"))
```

```
encrypt = "Hello World"  
print(encrypt)
```

```
encrypt("Hello World")  
print(encrypt())
```

```
message = encrypt("Hello World")  
print(encrypt)
```



## Question 4

Add comments to the code below explaining what is happening where indicated (//).

```
function encrypt(message)
    //
    messageLength = message.length()
    addition = 1
    newMessage = ""
    //
    for count = 0 to messageLength-1
        //
        ASCIIValue = ASC(message.substring(count,1))
        //
        ASCIIValue = ASCIIValue + addition
        //
        newMessage = newMessage + CHR(ASCIIValue)
        //
        if addition = 11 then
            addition = 1
        else
            //
            addition = addition + 1
        endif
    next count
    //
    return newMessage
endfunction
```

## Question 5

The function decrypt takes an encrypted message and decrypts it. Complete the program code where indicated by a red question mark (?).

```
function decrypt(?)  
    messageLength = encryptedMessage.length()  
    addition = 1  
    newMessage = ""  
    for count = 0 to messageLength-1  
        ASCIIValue = ASC(encryptedMessage.substring(count,1))  
        ASCIIValue = ASCIIValue ?  
        newMessage = newMessage + CHR(?)  
        if addition = ? then  
            addition = 1  
        else  
            addition = addition + 1  
        endif  
    next count  
    return ?  
endfunction
```

## Mastermind

Mastermind is a game where a player (the setter) chooses five coloured dots and puts them in a set order. The second player (the guesser) has to work out which dots have been chosen and in what order they are position.

The guesser has to select 5 dots in a specific order. Then the setter then tells them a) how many dots are the right colour in the right place, and b) how many dots are the right colour but in the wrong place.

For example.

The setter selects:

Red	Green	Blue	Black	Green
-----	-------	------	-------	-------

The guesser guesses:

Blue	Green	Blue	Green	Blue
------	-------	------	-------	------

They will be told they have 2 correct colours in the correct place (green in position 2, blue in 3), and a correct colour in the wrong place (green in 4).

The game is being created as a computer game, but the computer will act as the setter. The player has to work out the combination the computer has selected.

The function guesser allows the player to choose their selection and it returns this to the main program as an array.

```
function guesser()
    print("Choose colours from blue, green, black, red, white, yellow")
    dot1 = input("What colour is the dot in position 1")
    dot2 = input("What colour is the dot in position 2")
    dot3 = input("What colour is the dot in position 3")
    dot4 = input("What colour is the dot in position 4")
    dot5 = input("What colour is the dot in position 5")
    guess = [dot1, dot2, dot3, dot4, dot5]
    guesser = guess
endfunction
```

## Question 1

What is the most appropriate data type to use to store the colour of the dot?

## Question 2

Which line number is where the data is returned from the function?

```
01 function guesser()  
02   print("Choose colours from blue, green, black, red, white, yellow")  
03   dot1 = input("What colour is the dot in position 1")  
04   dot2 = input("What colour is the dot in position 2")  
05   dot3 = input("What colour is the dot in position 3")  
06   dot4 = input("What colour is the dot in position 4")  
07   dot5 = input("What colour is the dot in position 5")  
08   guess = [dot1, dot2, dot3, dot4, dot5]  
09   guesser = guess  
10 endfunction
```

## Question 3

The function `getCorrect` takes the answer and guesser selections as parameters and returns the number of correct dots. Complete the algorithm where indicated by the red question mark (?).

```
function getCorrect(computerGrid[], playerGrid[])  
    correct = ?  
    for count = 0 to 4  
        if computerGrid[count] == playerGrid[count] then  
            correct = correct + ?  
        endif  
    next ?  
    return ?  
endfunction
```

## Question 4

The function `wrongPlace` compares the answer with the guess, and returns the number of colours that are correct but in the wrong place. Complete the algorithm where indicated by the red question mark (?) from the list below:

- 0
- 4
- Wrong
- Null
- !=
- if

```
function wrongPlace(computerGrid[], playerGrid[])
    correctWrongPlace = ?
    wrongColour = []
    for count = 0 to 4
        ? computerGrid[count] ? playerGrid[count] then
            wrongColour.append[playerGrid[count]]
        else
            computerGrid[count] = ?
        endif
    next count

    for ? = 0 to wrongColour.length-1
        for count = 0 to ?
            if computerGrid[count] == wrongColour[wrong] then
                correctWrongPlace = correctWrongPlace + 1
                computerGrid[count] = Null
            endif
        next count
    next wrong
endfunction
```

## Question 5

The main program code is shown below. Highlight all the function identifiers that are called in this part of the program.

```
print("Welcome to Mastermind")

computerBoard = computerSelect()

won = False

playerGuess = [Null, Null, Null, Null, Null]

attempts = 0

while won == False
    playerGuess = guesser()
    attempts = attempts + 1
    correct = getCorrect(computerGrid[], playerGrid[])
    print("You got " & correct " dots correct in the correct place")
    if correct = 5 then
        print("Whoo! You win. It took you " & attempts & " guesses")
        won = True
    else
        wrong = wrongPlace(computerGrid[], playerGrid[])
        print("You got " & wrong " colours correct but in the wrong place")
    endif
endwhile
```

## Complete the code

For the following questions, complete the code where indicated by the red question marks (?) or highlight the code as required by the question.

### Question 1

The following program should input 3 numbers and output the smallest. Complete the program code.

```
num1 = input("Enter number 1")
num2 = input("Enter number 2")
num3 = input("Enter number 3")
if num1 < num2 ? num1 < num3 then
    print(?)
elseif num2 < num1 ? num2 < num3 then
    print(?)
else
    print(?)
endif
```

### Question 2

The following program should perform an insertion sort on the array `theData`. Complete the program code.

```
procedure insertionSort(theData)
    for count = 2 to theData.length()
        temp = theData(?)
        temp2 = count - 1
        while temp2 >= 0 and theData[temp2] > temp
            theData[temp2+1] = theData[temp2]
            temp2 = ? - 1
        endwhile
        theData[temp2 + 1] = ?
    next count
endprocedure
```

### Question 3

The following program should input 2 numbers, add them together and output the total. Complete the program code.

```
? = input("Enter first number")
? = input("Enter second number")
? = value1 + value2
print(total)
```

### Question 4

The following SQL script should return the fields firstName, lastName and dateOfBirth for all people with the title "Miss". Complete the SQL script.

```
Select ?, ?, ?  
? PEOPLE  
WHERE Title = "?"
```

### Question 5

The following function should take two numbers as parameters and return the first number to the power of the second number, divided by the first number. Complete the program code.

```
function calculate(?, ?)  
    return (first ? second) / ?  
endfunction
```

### Question 6

The program should perform a bubble sort. Highlight the logic error below.

```
length = theData.length  
sorted = True  
while sorted == False  
    sorted = True  
    for x = 0 to length - 1  
        if theData(x) > theData(x+1)  
            temp = theData(x)  
            theData(x) = theData(x+1)  
            theData(x+1) = temp  
            sorted = True  
        endif  
    next x  
endwhile
```

### Question 7

The program should output 22 numbers. Highlight the **one** logic error.

```
for x = 0 to 20  
    print x  
next x
```



## Question 8

The program should output:

- The result of two numbers added together
- The result of the first number subtracted from the second number
- The result of the first number divided by the second – only displaying the integer division

Highlight the **two** logic errors.

```
x = 10
y = 20
print(x + y)
print(x - y)
print(x MOD y)
```

## Question 9

Highlight the concatenation symbol.

```
firstName = input("Enter first name")
secondName = input("Enter second name")
result = firstName.substring(0,3) & secondName.substring(0,3)
```

## Question 10

Highlight the iteration statement.

```
count = 0
while count <= 10
    print(count ** count)
endwhile
```