1)

(a) Describe an algorithm to insert one data item into a queue data structure.

……

……

……

……

……

……

……

…… [4]

(ii) Demonstrate an insertion sort to place the following numbers into **descending** numerical order.

12 7 4 5 26

……

……

……

……

……

……

……

…… [4]

(iii) State **one disadvantage** of an insertion sort compared with a quick sort.

……

…… [1]

3) The layout for a 2-player game is shown below.

START	1	2	3	4	5	6	7
15	14	13	12	11	10	9	8
16	17	18	19	20	21	22	23
31	30	29	28	27	26	25	24
32	33	34	35	36	37	38	39
47	46	45	44	43	42	41	40
48	49	50	51	52	53	54	55
END	62	61	60	59	58	57	56

The game is played by rolling two 6-sided dice and moving that number of spaces. Both players start on the START space. If a player lands on a space occupied by the other player, they move to the next available space.

The board is to be stored as a 2-dimensional array.

(b) Each time a player moves, a series of obstacles are to be added to the board.

On their turn, each player rolls two dice. The smaller number from the two dice is taken, and that many obstacles will appear on the board in random locations.

For example, if a 3 and 6 are rolled, then 3 obstacles will appear.

A recursive function is written in pseudocode to perform this task.

```
01 function generateObstacle(diceNumber)
02     if diceNumber == 0 then
03         return true
04     else
05         x = randomNumber(0, 7)
06         y = randomNumber(0, 7)
07         board(x, y) = new obstacle()
08         generateObstacle(diceNumber-1)
09     endif
10 endfunction
```

The code `new obstacle()` generates an instance of the object obstacle.

(i) Explain the purpose of the code in line 01 in the algorithm.

.....

.....

.....

..... **[2]**

(ii) Identify the line of code where recursion occurs.

..... **[1]**

(v) If a position on the board is not occupied, its value is set to a blank string ("").

The current algorithm does not check if the random space generated is currently occupied.

Write a subroutine that takes the generated position of the board, checks if it is free and returns `true` if free, or `false` if occupied.

.....

.....

.....

.....

.....

.....

.....

.....

..... [3]

4) Explain the stages of a merge sort, to sort these numbers into ascending order, using the data set shown below:

34 12 76 15 13 27 11 59 60 3

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... [4]

5) a. Explain one possible advantage of using a merge rather than an insertion sort to sort the data. **[2]**

……

……

……

……

b. Explain one disadvantage of using a merge rather than a quicksort to sort the data. **[2]**

……

……

……

……

6) A company releases an Internet connected fridge. Users can email messages to the fridge and it puts them on its display.

When the fridge receives a message it takes the string and stores it in a queue called `words`.

For example `REMEMBER TO TAKE CHARLIE TO THE DENTIST THIS AFTERNOON` becomes a queue:

```
words=["REMEMBER","TO","TAKE","CHARLIE","TO","THE","DENTIST",
"THIS","AFTERNOON"]
```

`words.remove()` then returns the next item in the queue
for example `temp=words.remove()` assigns `temp` the value `"REMEMBER"` and leaves `words` as `["TO","TAKE","CHARLIE","TO","THE","DENTIST","THIS","AFTERNOON"]`

The display has four lines; each can show a maximum of 20 characters including spaces.

If a word can't fit on a line a new line is started.

Examples

R	E	M	E	M	B	E	R		T	O		T	A	K	E				
C	H	A	R	L	I	E		T	O		T	H	E						
D	E	N	T	I	S	T		T	H	I	S								
A	F	T	E	R	N	O	O	N											

G	E	T		S	O	M	E		M	O	R	E							
C	H	O	C	O	L	A	T	E		P	L	E	A	S	E				

The contents of the display are stored in a 2D array of characters called `display`.

The procedure `updateDisplay` receives the queue `words` which holds the message and writes the message to the display.

Write the procedure `updateDisplay`. Credit will be given for the readability of your code.

You can assume:

- Messages contain no punctuation.
- All messages will fit on the display.
- The previous message is removed before the procedure is run.

```
global array display[20,4]
...
...
...
procedure updateDisplay(words)

    ..................................................................
    ..................................................................
    ..................................................................
    ..................................................................
    ..................................................................
    ..................................................................
    ..................................................................
    ..................................................................
    ..................................................................
    ..................................................................
    ..................................................................
    ..................................................................
    ..................................................................
    ..................................................................
    ..................................................................
    ..................................................................
    ..................................................................

endprocedure
```

[7]

7) Players are given 10 random letters and asked to find the largest word they can make from those letters. Each letter can only be used once. The length of the word determines the number of points awarded. e.g. a word with 6 letters would mean 6 points are awarded.

The function validateAnswer() takes in the randomLetters as an array of letters and the player's answer as a string. It then checks if the word the player has entered only contains letters from the 10 random letters with each letter being used only once. (At this stage the program doesn't check if the answer provided is an actual word.)

It then returns a score, out of 10, for a valid word or 0 for an invalid word.

Example:

If the random letters are OPXCMURETN

The word COMPUTER returns 8

Whereas

The word POST returns 0 (there is no S in the random letters).

The word RETURN returns 0 (there is only one R in the random letters).

(b) Complete the function `validateAnswer`

```
function validateAnswer(answer, randomLetters[])

    ..........................................................

    ..........................................................

    ..........................................................

    ..........................................................

    ..........................................................

    ..........................................................

    ..........................................................

    ..........................................................

    ..........................................................

    ..........................................................

    ..........................................................

    ..........................................................

    ..........................................................

    ..........................................................

    ..........................................................

endFunction
```

[6]

(c) Code is to be added to check if the word is an actual English word. All English words are stored in a binary search tree.

Give **one** advantage of storing the words in a binary search tree over an array.

..... **[1]**

8) A list of integers is to be sorted into numeric order. The programmer must decide whether to use a bubble or quick sort in order to do the task.
Compare the use of the two sorts above, including detail about complexity and any assumptions that you have made about the data set itself. **[8]**

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....