12 **(a)** A stack has been implemented using pseudocode to store a maximum of 100 string items using the global variables in the following table:

Identifier	Data type	Description	Initialisation value
`Base`	`INTEGER`	pointer for the bottom of the stack	`0`
`Top`	`INTEGER`	pointer for the top of the stack	`-1`
`StackArray`	`STRING`	1D array to implement the stack	`[0:99]`
`Max`	`INTEGER`	maximum number of items in the stack	`100`

The value of `Top` is incremented each time a data item is added to the stack and decremented each time a data item is removed. If the stack is full, an appropriate error message is output.

(i) Complete the **pseudocode** for the procedure to add a data item onto the stack.

```
PROCEDURE Push(.....)
   IF Top < Max - 1 THEN

      Top ← .....

      ..... ← NewData
   ELSE

      OUTPUT .....
   ENDIF
ENDPROCEDURE
```

[4]

(ii) Write **pseudocode** to input a new data item and add it to the stack using `Push()`.

.....

.....

.....

..... [2]

(b) Explain the reasons why a stack is used when a recursive algorithm is executed.

.....

.....

.....

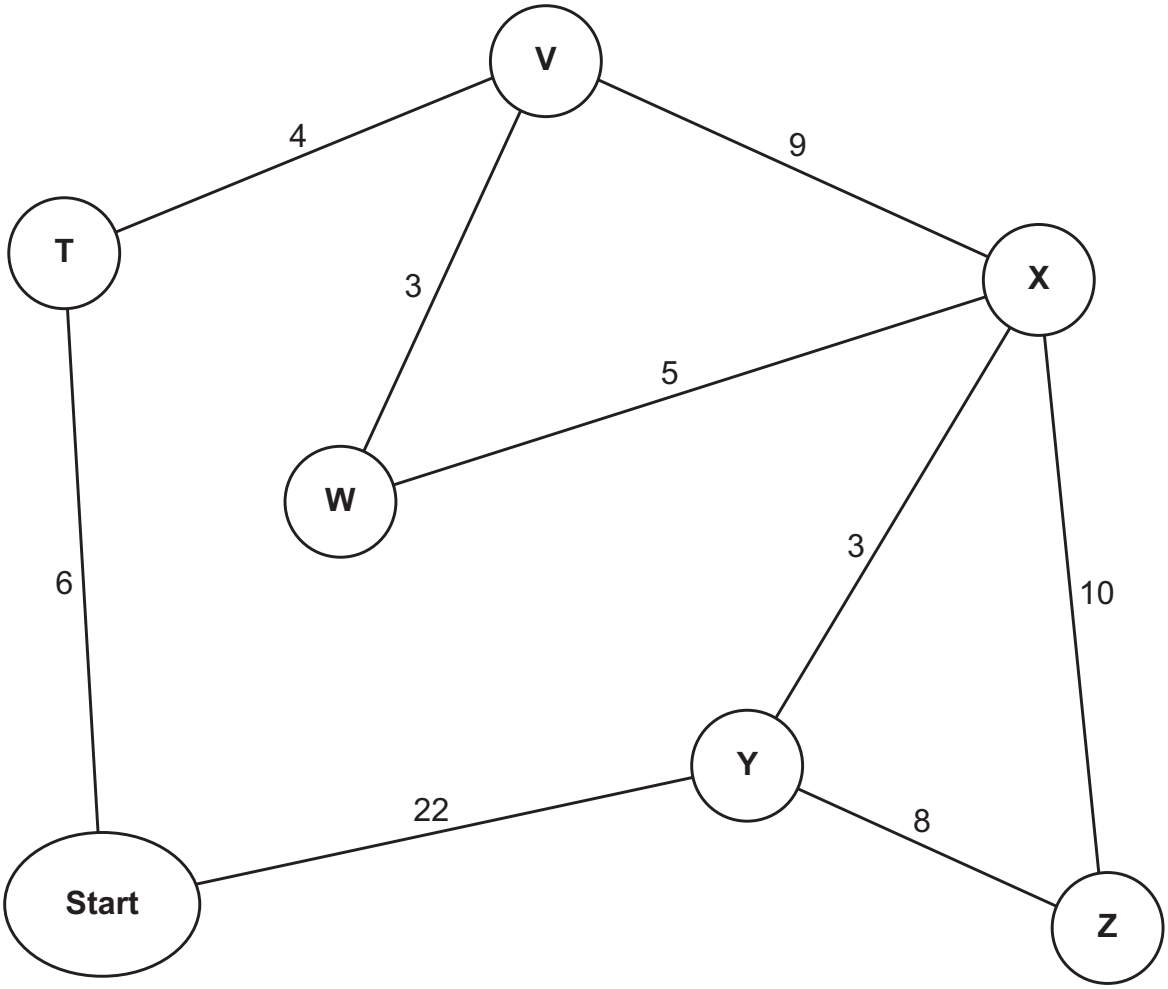
.....

.....

..... [3]

8 Calculate the shortest distance between the **Start** node and each of the nodes in the graph using Dijkstra's algorithm.

Show your working on the graph **or** in the working space. Write your answers in the table provided.



Working .....

.....

.....

.....

.....

.....

Answers:

T	V	W	X	Y	Z

[5]

7 A program is being developed to implement a customer loyalty scheme for a coffee shop.

Each customer has a unique customer ID starting at `10001` with this value increasing by one each time a new customer joins the loyalty scheme.

For example, the third customer who joins the loyalty scheme is given the customer ID `10003`

The loyalty scheme is limited to 1000 customers.

A customer is awarded a loyalty point every time they buy a coffee.

The programmer has decided to use a global 2D array `Loyalty` of type `INTEGER`. The array `Loyalty` is made up of 1000 rows and 2 columns. Each row relates to one customer; column 1 contains the unique customer ID and column 2 contains the number of customer loyalty points.

Rows in the array `Loyalty` that are not currently being used have the value of Column 1 set to `99999`

The array is sorted in ascending order by customer ID.

The programmer has defined a program module:

Module	Description
`FindCustomer()`	• called with parameter of type `INTEGER` representing a customer ID • searches the `Loyalty` array for this customer ID • the search will stop as soon as the customer ID is found • the search should efficiently deal with the situation when the customer ID is **not** stored in the `Loyalty` array • if the customer ID is found, return an integer value representing the loyalty points, otherwise return `-1`

(a) Write efficient pseudocode for module `FindCustomer()`

[8]

(b) A customer can claim a free coffee for every 11 loyalty points.

The programmer has defined a second program module:

Module	Description
`PointsReport()`	• output the customer ID for each customer who has 11 or more loyalty points • output the average loyalty points for all customers in the `Loyalty` array along with an appropriate message

Write efficient pseudocode for module `PointsReport()`

Assume the array contains the data for at least **one** customer.

.....

.....

.....

.....

.....

.....

.....

..... [7]

(c) The programmer decides to amend the customer ID; it will be stored as a `STRING` instead of an `INTEGER`. This means that the 2D array `Loyalty` can no longer be used.

(i) Explain why the 2D array `Loyalty` can no longer be used.

.....

..... [1]

(ii) Explain how a 1D array could be used to store both the loyalty points and the amended customer ID.

.....

.....

.....

..... [2]