Question A (40 Points) – I.T. Time!

While an IDE is not required to write this answer, you may use OneCompiler if you like.

Your company was recently affected by the disastrous Crowdstrike update. The CEO has asked you to calculate how long it will take for each of your department's computers and servers to be fixed. The estimated times are below:

	Work Computer	**Server**
Standard	15 minutes	19 minutes
With Bitlocker	39 minutes	57 minutes

Your program should repeatedly prompt the user to enter the department's information (how many computers and servers need to be fixed) or to quit. When they quit, your program should calculate the subtotals for each type, then display the final time to the user.

Example Output

(Note that your program must meet all requirements to earn full marks – this is just an example of possible output)

```
[Post-Crowdstrike Plan]

Department 1

How many computers need to be fixed? 5

How many of those computers had Bitlocker enabled? 2

How many servers need to be fixed? 8

How many of those servers had Bitlocker enabled? 6

Is there another department? Y

Department 2
```

How many computers need to be fixed? **3**

How many of those computers had Bitlocker enabled? **0**

How many servers need to be fixed? **5**

How many of those servers had Bitlocker enabled? **2**

Is there another department? **N**

Across all departments, there are 8 computers and 13 servers. Of those, 2 computers and 8 servers had Bitlocker enabled.

The 6 computers without Bitlocker will take 90 minutes to fix.

The 2 computers with Bitlocker will take 78 minutes to fix

The 5 servers without Bitlocker will take 95 minutes to fix.

The 8 servers with Bitlocker will take 456 minutes to fix

Assuming we fix them one at a time, it will take 719 minutes to repair all devices.

Question B (35 Points): Sorting with Strings!

While an IDE is not required to write this answer, you may use OneCompiler if you like.

In this course, we learned how to sort a list of numbers. For this question, you will sort a list of Strings!

Ask the user how many names they want to enter, then prompt for those names and store them in a String list. Sort the list by string length from longest to shortest. You will then print the sorted list. To accomplish this, you will need to create and use at least two different functions:

- **sortByString():** This should take in a String list and sort it based on the length of the string in each list index.

  Remember that there are built-in String functions that will tell you how long the string is.

- **print_list**(): This should print out the contents of the list.

NOTE: You *must* implement the list sorting and printing code yourself – do not use any built-in methods to do this. You will receive no credit if you do not implement the sorting and printing code yourself. You must also use these methods rather than implementing the functionality in the main method.

Sample Output

[String List Sorting]
How many names do you want to sort? **6**
Please enter Name 1: **Sally**
Please enter Name 2: **Estanislao**
Please enter Name 3: **Omar**
Please enter Name 4: **Li**
Please enter Name 5: **Dymtrus**
Please enter Name 6: **Yamamoto**

Now sorting...

Sorted!
Estanislao, Yamamoto, Dymtrus, Sally, Omar, Li,

Question C (25 Points) – **Programming Languages**

While an IDE is not required to write this answer, you may use OneCompiler if you like.

In the last week of the lab, you learned that there are other programming languages besides Python. For this final question, you're going to store and display information about some of those languages in a program. You will use **objects** to complete this assignment – you will not receive points if you do not use objects or hardcode the responses.

You will start by creating a **Programming** class. It should have the following properties with the appropriate data types.

Variable Name	**Possible Values**
name	Valid String
uses_braces	True, False
syntax	"ELSE IF", "ELIF", or "elseif"

You should have a constructor that takes in those parameters and assigns them in the object. It should also have default values set to initialize the name to "Python", uses_braces to False, and syntax to "elif". If the user passes in an invalid value for syntax, it should default to ELSE.

Finally, you should create a languageInfo() method that prints out information about the language. It should take a Programming object as a parameter, and compare the uses_braces and syntax variables. This information should also be included in the print-out. Look at the example output to see how it should be formatted.

You will then write a driver class. In it, create a Programming object that uses the default values. Create a second Programming object that uses custom values (you may use whatever programming language you want). Then call, the languageInfo() function from the first Programming object and pass the second Programming object as input.

Example Output (if your Programming object is Java)

```
[Programming Languages]
```

```
Python is a programming language.

Unlike Java, it does not use curly braces.

Unlike Java, its conditional statements are IF, ELIF, ELSE

(Java uses ELSE IF)
```

Example Output (if your Programming object is Bash)

```
[Programming Languages]

Python is a programming language.

Like Bash, it does not use curly braces.

Like Bash, its conditional statements are IF, ELIF, ELSE
```