Section C

Question 1.
Open the program called Question1.py from your device.
Enter your name in the space provided on Line 2.

This is a simple calculator program that can add and subtract two numbers. When this program is run it prompts the user to select addition or subtraction.

The user enters the letter 'a' if they wish to add the numbers or enters the letter 's' if they wish to subtract.

```
1  # Question 1(a)
2  # Name:
3
4  num1 = 9
5  num2 = 5
6
7  choice=str(input('Do you want to (a)dd or (s)ubtract?: '))
8
9  if choice == 'a':
10     print(num1+num2)
11 elif choice =='s':
12     print(num1-num2)
```

Question 1 starting code

Modify the program to do the following:

(i) Add a comment at the start of the program that states 'This calculator can only add and subtract'.

(ii) The user should be prompted to enter their name when the program runs. A **suitable variable** should be used to **store the name**:

```
Please enter your name: Martin
```

(iii) The program should **output the following to the screen**, including the **user's name**:

```
Welcome to the addition and subtraction Martin
```

(iv) The program currently adds or subtracts the two numbers stored in the variables **num1** and **num2**.

Modify the program so that the **user is asked** to **enter the numbers** that will be added or subtracted:

```
Please enter the first number for the calculation: 10
Please enter the second number for the calculation: 8
```

(v) Currently the program only displays the answer. If the user enters 10 and 8 as the variables, when addition is selected, the output is: 18

Modify the program so that it **outputs the equation and the answer**.
When the program is run the output may look as follows:

```
Do you want to (a)dd or (s)ubtract?: a
10 + 8 = 18
```

(vi) The program currently only works if the user enters a lowercase 'a' for addition or a lowercase 's' for subtraction.

Modify the program so that it will **<u>also</u>** work if the user enters an 'A' for addition or an 'S' for subtraction.

```
Do you want to (a)dd or (s)ubtract?: A
10 + 8 = 18
```

(vii) If the user enters an invalid option (anything other than 'a', 'A', 's' or 'S') the program terminates without warning.

Modify the program so that if the user enters an invalid option the program will output a message stating**: Invalid option**

```
Do you want to (a)dd or (s)ubtract?: p
Invalid Option
```

(viii) Currently if the user gives an invalid input the program outputs an 'Invalid option' message.

Modify the program so that it will **continue to prompt** the user to enter a valid option.

When the program is run, the output may look as follows:

```
Do you want to (a)dd or (s)ubtract?: z
z
Invaid option
Do you want to (a)dd or (s)ubtract?: a
10 + 8 = 18
```

Question 2
Open the program called Question2.py from your device.

Enter your name in the space provided on line 2.
This file contains an empty list called **squared_numbers.**

A squared number is a number that is multiplied by itself, for example the first squared number is 1 (1*1), the second squared number is 4 (2*2) etc.

```
1  # Question 2
2  # Name:
3
4  squared_numbers=[]
5
```

Question 2 starting code

Modify the program to do the following:

(i) Calculate the first 20 squared numbers and place them in the list **squared_numbers.** You must use a **loop** and the **append()** method.

(ii) Output the list of **squared_numbers** to the screen.

(iii) Find the **mean and the median** of the list of numbers and output the mean and median to the user. You may use whatever method you like to find the mean and the median of the list.

When the program is run it may look something like below.

```
[1, 4, 9, 16, 25, 36, 49, 64, 81, 100, 121, 144, 169, 196, 225, 256, 289, 324, 361, 400]
The mean is: 143.5
The median is: 110.5
```

Question 3
Open the program called Question3.py from your device.

Enter your name in the space provided on line 2.

A friend has asked you to help them to create a program that will solve quadratic equations quickly and display the roots to the user. However, your friend is unsure what to do next with their code.

```
1  # Question 3
2  # Name:
3
4  import math
5
6  print("Quadratic functions take the form : (Ax^2) + Bx + C=0")
7
8  a=float(input("a ="))
9  b=float(input("b ="))
10 c=float(input("c ="))
```

Question 3 starting code

You can find the two roots of any quadratic equation using the following two formulas:

$$\text{Root 1} = \frac{-b + \sqrt{b^2 - 4ac}}{2a} \qquad \text{Root 2} = \frac{-b - \sqrt{b^2 - 4ac}}{2a}$$

Modify the program to do the following:

(a) Using the given formulas calculate values for **Root 1 and Root 2.** You can find the square root of a value by using the code **math.sqrt( )** method.

If you use the values **a**= 1, **b**= -5, **c**= -6, your roots should be 6 and -1.

```
Quadratic functions take the form : (Ax^2) + Bx + C=0

a =1
b =-5
c =-6


Root 1 = 6.0
Root 2 = -1.0
```

(b) Modify your code so that if the user inputs a value of '0' for the 'a' variable, they get an output telling them their equation is not a quadratic.

```
Quadratic functions take the form : (Ax^2) + Bx + C=0

a =0
b =-5
c =-6
If a=0 then your equation is not a quadratic
```

(c) Using the rules shown below in Fig 1, modify the code so the program will tell the user what kind of roots the equation has.

1. If $(b^2 - 4ac) > 0 \Rightarrow$ two different (distinct) real roots
2. If $(b^2 - 4ac) = 0 \Rightarrow$ two equal real roots
3. If $(b^2 - 4ac) < 0 \Rightarrow$ two imaginary roots

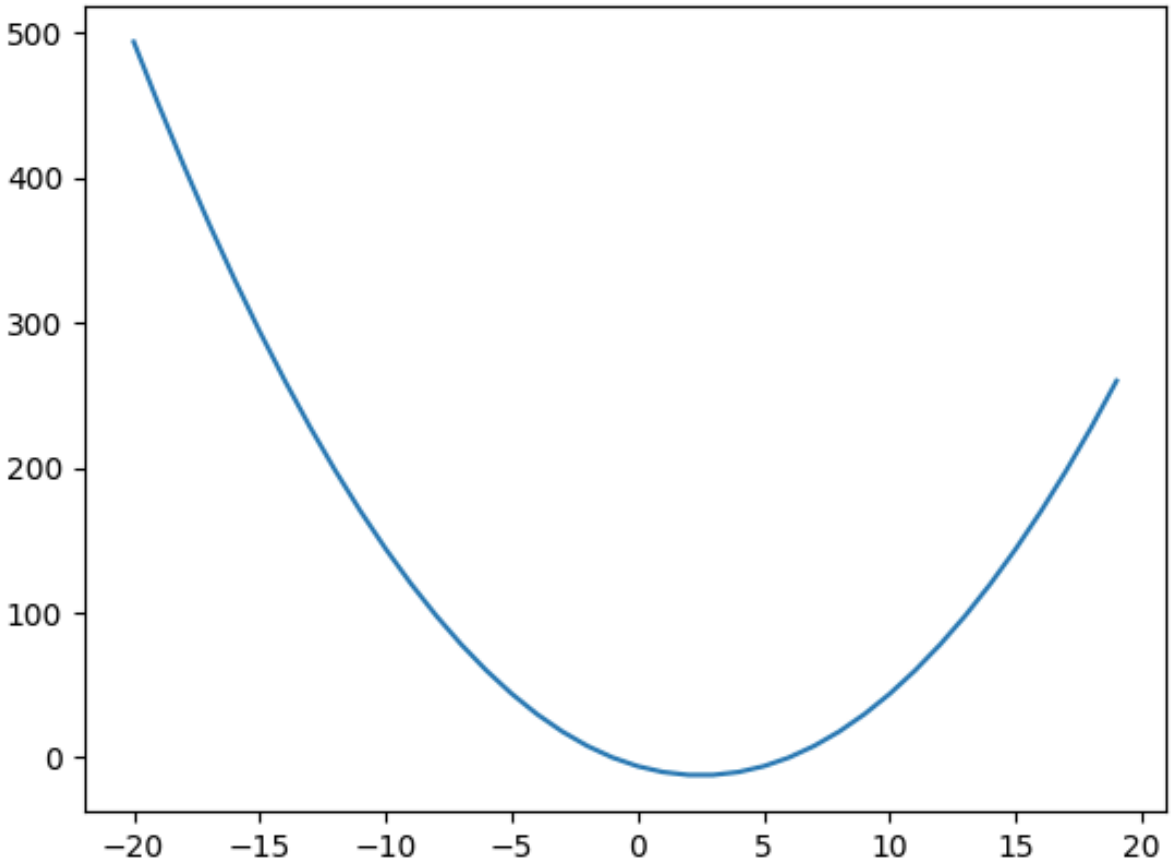
Fig 1

```
Quadratic functions take the form : (Ax^2) + Bx + C=0

a =1
b =-5
c =-6
6.0,-1.0 are two different real roots
```

(d) Using the following pseudocode modify the program so that the program outputs the graph of a given equation.

```
Import matplotlib.pyplot as plt
Create a list to store x co-ordinates for graph
Create a list to store y co-ordinates for graph
For x values in the range -20 to 20
        y=a*(x**2) + (b*x) + c
        append x value to x values list
        append y value to y values list
fig= plt.figure()
axes=fig.add_subplot(111)
axes.plot(xValues,yValues)
plt.show()
```



Output for a=1 b=-5 c=-6