

Question 15

Open the file labelled **Question 15_starter**, and place your name where It indicates at the start of the program.

```
# Question 15
# Name

print('Program to solve linear and quadratic function problems\n') # Q1

print ('Linear fucntions take the form of f(x) = ax + b')
print ('Quadratic functions take the form of f(x) = cx^2 + dx + e \n')

quad_points=[]
lin_points=[]

# Coefficients and x values
a = 2
b = -3
x = 1

c = 1
d = -2
e = -3

print(f'The linear function {a}x + {b}, has an output of {(a*x)+b} for an input x value of {x}')
```

Starter Code

Both quadratic and linear functions play roles in many disciplines in the real world, from science to banking. In some processes linear functions may intercept quadratic functions indicating where they would have similar outputs for a given input value. Linear functions may intercept quadratic functions at just one co-ordinate or at two co-ordinates.

A linear function is a function in the form $y = Ax + B$, where A and B are constants, x is an input variable and y is the corresponding output. For example,

$$\text{Linear Function: } y = 4(x) + 3$$

Input (x value)	Function (4(x) + 3)	Output (y value)
1	4(1) + 3	7
Co-ordinate (x , y). \rightarrow (1, 7)		

A quadratic function is a function in the form $y = Cx^2 + Dx + E$, where C, D and E are constants, x is an input variable and y is its corresponding output. For example,

$$\text{Quadratic Function: } y = 2(x)^2 + 3(x) + 1$$

Input (x value)	Function (2(x)^2 + 3(x) + 1)	Output (y value)
1	2(1)^2 + 3(1) + 1	8
Co-ordinate (x , y). \rightarrow (1, 8)		

(a) Modify the code so a message is displayed at the start of the program stating 'Program to solve linear and quadratic function problems'. Your output should look similar to Fig 1

```
Program to solve linear and quadratic function problems

Linear fucntions take the form of  $f(x) = ax + b$ 
Quadratic functions take the form of  $f(x) = cx^2 + dx + e$ 

The linear function  $2x + -3$ , has an output of  $-1$  for an input  $x$  value of  $1$ 
```

Fig 1

(b) Currently the x variable is hard coded to '1'. However, when plotting functions we must find the y value for a range of different x values.

Using a **for loop**, modify the program to output to the user the y values for the linear function for all the x values in the range $-1 < x < 4$. The linear function is $2x + b$. Your output should look similar to Fig 2.

```
Program to solve linear and quadratic function problems

Linear fucntions take the form of  $f(x) = ax + b$ 
Quadratic functions take the form of  $f(x) = cx^2 + dx + e$ 

The linear equation  $2x + -3$  in the range  $-1 < x < 4$  has the following (x,y) co-ordinates
(x,y)-->(-1, -5)
(x,y)-->(0, -3)
(x,y)-->(1, -1)
(x,y)-->(2, 1)
(x,y)-->(3, 3)
(x,y)-->(4, 5)
```

Fig 2

(c) Using a **for loop**, modify the program to output to the user the y values for the quadratic function for all the x values in the range $-1 < x < 4$. The quadratic function is $x^2 - 2x - 3$. Your output should look similar to Fig 3.

```
Program to solve linear and quadratic function problems

Linear fucntions take the form of  $f(x) = ax + b$ 
Quadratic functions take the form of  $f(x) = cx^2 + dx + e$ 

The linear equation  $2x + -3$  in the range  $-1 < x < 4$  has the following (x,y) co-ordinates
(x,y)-->(-1, -5)
(x,y)-->(0, -3)
(x,y)-->(1, -1)
(x,y)-->(2, 1)
(x,y)-->(3, 3)
(x,y)-->(4, 5)

The quadratic equation  $1x^2 + -2x + -3$  in the range  $-1 < x < 4$  has the following (x,y) co-ordinates
(x,y)-->(-1, 0)
(x,y)-->(0, -3)
(x,y)-->(1, -4)
(x,y)-->(2, -3)
(x,y)-->(3, 0)
(x,y)-->(4, 5)
```

Fig 3

(d) It is important to find the point of intersection between a linear and quadratic function.

To do this you need to identify (x, y) co-ordinates from both functions that are identical. Modify your program to output to the user the points of interception between the given linear and quadratic functions. Your output should look similar to Fig 4.

```
Program to solve linear and quadratic function problems

Linear fucntions take the form of  $f(x) = ax + b$ 
Quadratic functions take the form of  $f(x) = cx^2 + dx + e$ 

The linear equation  $2x + -3$  in the range  $-1 < x < 4$  has the following (x,y) co-ordinates
(x,y)-->(-1, -5)
(x,y)-->(0, -3)
(x,y)-->(1, -1)
(x,y)-->(2, 1)
(x,y)-->(3, 3)
(x,y)-->(4, 5)

The quadratic equation  $1x^2 + -2x + -3$  in the range  $-1 < x < 4$  has the following (x,y) co-ordinates
(x,y)-->(-1, 0)
(x,y)-->(0, -3)
(x,y)-->(1, -4)
(x,y)-->(2, -3)
(x,y)-->(3, 0)
(x,y)-->(4, 5)

The functions intercept at point [0, -3]
The functions intercept at point [4, 5]
```

Fig 4

Question 16

Open the file labelled **Question 16_starter**, and place your name where it indicates at the start of the program.

The genetic code of each living organism is a long sequence of simple molecules called nucleotides or bases. Although many nucleotides exist in nature, only 4 nucleotides, labeled A, C, G, and T, have been found in DNA. They are abbreviations of Adenine, Cytosine, Guanine, and Thymine. The NCBI (National Center for Biotechnology Information) has recently started maintaining a data hub for genetic sequences related to the virus causing COVID-19 and a part of that genetic code is stored in the variable *genome* in your program. Since most data come in files and streams, a data scientist must be able to effectively work with them. Python provides many facilities to make this easy.

```
# Q16 (b)
# Name:

# Sample of Covid19 genome
genome = (''ATTAAAGTTTATACCTTCCCAGGTAACAAACCAACCACTTTTCGATCTCTTGTAGATCTGTTCTCTAAACGAACCTTTAGTAA
AAATCTGTGTGGCTGTCACTCGGCTGCATGCTTAGTGCACTCACGCAGTATAATTAATAACTAATTACTGTCGTTGACAG
GACACGAGTAACCTCGTCTATCTTCTGCAGGCTGCTTACGGTTTCGTCCGTGTTGCAGCCGATCATCAGCACATCTAGGTT
TCGTCCGGGTGTGACCGAAAGGTAAGATGGAGAGCCTTGTCCCTGGTTTCAACGAGAAAAACACAGTCCAACCTCAGTTTG
CCTGTTTTACAGGTTTCGCGACGTGCTCGTACGTGGCTTTGGAGACTCCGTGGAGGAGGTCTTATCAGAGGCACGTCAACA
TCTTAAAGATGGCACTTGTGGCTTAGTAGAAGTTGAAAAAGGCGTTTTGCCTCAACTTGAACAGCCCTATGTGTTTATCA
AACGTTTCGGATGCTCGAAGTGCACCTCATGGTTCATGTTATGGTTGAGCTGGTAGCAGAACTCGAAGGCATTTCAGTACGGT
CGTAGTGGTGAGACACTTGGTGTCTTGTCCCTCATGTGGGCGAAATACCAAGTGGCTTACCGCAAGGTTCTTCTTCGTAA
GAACGGTAATAAAGGAGCTGGTGGCCATAGTTACGGCGCCGATCTAAAGTCATTGACTTAGGCGACGAGCTTGGCACTG
ATCCTTATGAAGATTTTCAAGAAACTGGAACACTAAACATAGCAGTGGTGTACCCGTGAACCTCATGCGTGAGCTTAAC
GGAGGGGCATACACTCGCTATGTGATAACAACCTTCTGTGGCCCTGATGGCTACCTCTTGAAGTGCATTAAAGACCTTCT
AGCACGTGCTGGTAAAGCTTCATGCACCTTGTCCGAACAACTGGACTTTATTGACACTAAGAGGGGTGATAGTCTGCC
GTGAACATGAGCATGAAATTGCTTGGTACACGGAACGTTCTGCTGGCAATGGCGGTGATGCTGCTCTTGTCTTGTGCTG
CTTGACAGATTGAACAGCTTGAGAGCAAAATGTCTGGTAAAGGCCAACAACAAGGCCAACTGTCACTAAGAAATC
TGCTGCTGAGGCTTCTAAGAAGCCTCGGCAAAACGTAAGTCCACTAAAGCATACAATGTAACACAAGCTTTCGGCAGAC
GTGGTCCAGAACAAACCAAGGAAATTTGGGGACCAGGAACATCAGACAAGGAAGTATTACAAACATTGGCCGCAA
ATTGCACAATTTGCCCCAGCGCTTCAGCGTTCTTCGGAATGTGCGCATGGCAAGTCAACCTTCGGGAACGTG
GTTGACCTACACAGGTGCCATCAAATGGATGACAAAGATCCAAATTTCAAAGATCAAGTCATTTTGTGAATAAGCATA
TTGACGCATACAAAACATTCCCACCAACAGAGCTAAAAAGGACAAAAAGAGGCTGATGAACTCAAGCCTTACCG
CAGAGACAGAAGAAACAGCAAACTGTGACTCTTCTTCTGCTGCAGATTGGATGATTCTCCAAACAATTGCAACAATC
CATGAGCAGTGCTGACTCAACTCAGGCCTAAACTCATGCAGACCACACAAGGCAGATGGGCTATATAACGTTTTTCGCTT
TTCCGTTTACGATATAGTCTACTCTTGTGCAATGAATTCCTGTAACATAGCACAAGTAGATGTAGTTAACTTT
AATCTACATAGCAATCTTTAATCAGTGTGTAACATTAGGGAGGACTTGAAAGAGCCACCACATTTTCACCGAGGCCACG
CGGAGTACGATCGAGTGACAGTGAACATGCTAGGGAGAGCTGCCTATATGGAAGAGCCCTAATGTGTAAATTAATTT
TAGTAGTGCTATCCCATGTGATTTTATAGCTTCTTAGGAGAATGACAAAAAATTTTTTTTTTTTTTTTTTTTTT')

genome = genome.replace('\n','') # Removes the new line special character from the genome string.

# ORF7 RNA strains
RNA1 = 'auugcacauuuugccccagcgcuucagcgauuucggaugucgcauuggauggaugacacuccuuggggaacgug'
RNA2 = 'gaacgguuaauaaggagcugguggccauaguuaagcgccgaucuaaagucuuuagcguaggcgagcugugcacug'
RNA3 = 'gacacgaguuaucugcuuauuucugcaggcugcuuacgguuucgucgugugagcgcaucaucagcacauuagguu'
```

Starter Code

(i) The Covid-19 genome sequence can contain between 27-32 thousand nucleotides. Modify the program to tell the user the length of this part of the genome sequence. Your output should look similar to Fig 5.

```
- -
This part of the genome sequence contains 2000 nucleotides
```

Fig 5

(ii) When new viruses try to escape a cell, the cell can snare them with proteins called tetherin. Some research suggests that a protein called ORF7 cuts down an infected cell's supply of tetherin, allowing more of the viruses to escape.

Researchers have also found that the protein ORF7 can trigger infected cells to commit suicide - which contributes to the damage Covid-19 causes to the lungs. When describing the genome sequence of ORF7, the nucleotide T (Thymine) is often replaced by U (Uracil).

RNA1, *RNA2*, *RNA3*, contain the genetic code needed to make three different proteins that are part of ORF7. Modify the program so these genome sequences can be used later in the program.

- (i) Change 'u' nucleotide letter to 'T'
- (ii) Change each letter in the sequence from lower case to upper case.
- (iii) Output to the user each modified sequence.

Your output should look similar to Fig 6.

```
This part of the genome sequence contains 2000 nucleotides
```

```
The modified RNA sequences are:
```

```
ATTGCACAATTTGCCCCAGCGCTTCAGCGTTCTTCGGAATGTCGCGCATTGGCATGGAAGTCACACCTTCGGGAACGTG
GAACGGTAATAAAGGAGCTGGTGGCCATAGTTACGGCGCCGATCTAAAGTCATTTGACTTAGGCGACGAGCTTGGCACTG
GACACGAGTAACTCGTCTATCTTCTGCAGGCTGCTTACGGTTTCGTCCGTGTTGGAGCCGATCATCAGCACATCTAGGTT
```

Fig 6

(iii) Compare the each of the modified *RNA1*, *RNA2* and *RNA3* genome sequences to the Covid genome sequence and output to the user if they are found in the Covid genome sequence using an appropriate output. Your output should look similar to Fig 7.

```
This part of the genome sequence contains 2000 nucleotides
```

```
The modified RNA sequences are:
```

```
ATTGCACAATTTGCCCCAGCGCTTCAGCGTTCTTCGGAATGTCGCGCATTGGCATGGAAGTCACACCTTCGGGAACGTG
GAACGGTAATAAAGGAGCTGGTGGCCATAGTTACGGCGCCGATCTAAAGTCATTTGACTTAGGCGACGAGCTTGGCACTG
GACACGAGTAACTCGTCTATCTTCTGCAGGCTGCTTACGGTTTCGTCCGTGTTGGAGCCGATCATCAGCACATCTAGGTT
```

```
Strand ATTGCACAATTTGCCCCAGCGCTTCAGCGTTCTTCGGAATGTCGCGCATTGGCATGGAAGTCACACCTTCGGGAACGTG is in the Covid genome sequence
Strand GAACGGTAATAAAGGAGCTGGTGGCCATAGTTACGGCGCCGATCTAAAGTCATTTGACTTAGGCGACGAGCTTGGCACTG is in the Covid genome sequence
Strand GACACGAGTAACTCGTCTATCTTCTGCAGGCTGCTTACGGTTTCGTCCGTGTTGGAGCCGATCATCAGCACATCTAGGTT is not in the Covid genome sequence
```

Fig 7

(iv) The frequency of a base or a nucleotide in a genetic code is the number of times it occurs divided by the length of the code. The varying frequency of different nucleotides, called the nucleotide bias varies between organisms and is known to have biological implications. For example, comparing nucleotide bias of Covid strains allows scientists identify if the strains are the same or different.

Nucleotide bias =	Number of times base occurs
	Length of genetic sequence

Modify the program to calculate the nucleotide bias of this strain and output to the user the nucleotide bias values for each nucleotide in the genome sequence. Your output should look similar to Fig 8.

```
This part of the genome sequence contains 2000 nucleotides

The modified RNA sequences are:
  ATTGCACAATTTGCCCCAGCGCTTCAGCGTTCTTCGGAATGTCGCGCATTGGCATGGAAGTCACACCTTCGGGAACGTG
  GAACGGTAATAAAGGAGCTGGTGGCCATAGTTACGGCGCCGATCTAAAGTCATTGACTTAGGCGACGAGCTTGGCACTG
  GACACGAGTAACTCGTCTATCTTCTGCAGGCTGCTTACGGTTTCGTCCGTGTTGGAGCCGATCATCAGCACATCTAGGTT

Strand ATTGCACAATTTGCCCCAGCGCTTCAGCGTTCTTCGGAATGTCGCGCATTGGCATGGAAGTCACACCTTCGGGAACGTG is in the Covid genome sequence
Strand GAACGGTAATAAAGGAGCTGGTGGCCATAGTTACGGCGCCGATCTAAAGTCATTGACTTAGGCGACGAGCTTGGCACTG is in the Covid genome sequence
Strand GACACGAGTAACTCGTCTATCTTCTGCAGGCTGCTTACGGTTTCGTCCGTGTTGGAGCCGATCATCAGCACATCTAGGTT is not in the Covid genome sequence

The nucleotide bias of 'A' is 0.298
The nucleotide bias of 'C' is 0.2205
The nucleotide bias of 'G' is 0.227
The nucleotide bias of 'T' is 0.2545
```

Fig 8.

Question 17

Open the file labelled **Question 17_starter**, and place your name where It indicates at the start of the program.

```
# Question 17:
# Name:

Celsius = [0,10,20,30,40,50,60,70,80,90,100,110,120,130,140,150,160,170,180,190,200,210,220,230,240,250]
|
```

Starter code

Scientists tend to work with the metric system and often have to convert between imperial values and metric values. One of the most common conversions is to convert between degrees Fahrenheit and degrees Celsius. Celsius values can be converted to the Fahrenheit system by the formula shown below. Where F is degrees Fahrenheit and C is degrees Celsius.

$$F = \frac{9}{5}C + 32.$$

- (i) Use a while loop to convert all the Celsius values in the list *Celsius* to Fahrenheit and add the Fahrenheit values to their own separate list.

Output to the user both lists in the format shown in Fig 9.

F	C
32.0	0
50.0	10
68.0	20
86.0	30
104.0	40
122.0	50
140.0	60
158.0	70
176.0	80
194.0	90
212.0	100
230.0	110
248.0	120
266.0	130
284.0	140
302.0	150
320.0	160
338.0	170
356.0	180
374.0	190
392.0	200
410.0	210
428.0	220
446.0	230
464.0	240
482.0	250

Fig 9

(ii) In positive temperatures there is no temperatures whose Fahrenheit and Celsius values are the same.

Modify the program using either a For loop or a While loop to investigate if any negative temperatures have the same Celsius and Fahrenheit temperatures. Your program should stop running and output to the user the first temperature that is the same. Your output should look similar to Fig 10.

```
    F  C
-58.0 -50
-49.0 -45
-40.0 -40
-40 degrees Celsius is the same as -40.0 degrees Fahrenheit
```

Fig 10