Aayush Shah
D1 - 19BCE245
25 February 2021

Practical 1

B. Develop a python program to make a simple calculator using a conditional loop.

```
1. while 1:
2.    print("\nSelect Operation : \n0. Exit\n1. Add\n2.
   Subtraction \n3. Multiply\n4. Divide")
3.    choice = int(input("Select operations from 0,1,2,3,4 :
   "))
4.    if(choice == 0):
5.         print("Thank you")
6.         break
7.
8.    elif(choice == 1):
9.         n1 = int(input("Enter first number : "))
10.        n2 = int(input("Enter second number : "))
11.        ans = n1+n2
12.        print(n1 , " + " , n2 , " = " , ans)
13.
14.   elif(choice == 2):
15.        n1 = int(input("Enter first number : "))
16.        n2 = int(input("Enter second number : "))
17.        ans = n1-n2
18.        print(n1 , " - " , n2 , " = " , ans)
19.
20.   elif(choice == 3):
21.        n1 = int(input("Enter first number : "))
22.        n2 = int(input("Enter second number : "))
23.        ans = n1*n2
24.        print(n1 , " * " , n2 , " = " , ans)
25.
26.   elif(choice==4):
27.        n1 = int(input("Enter first number : "))
28.        n2 = int(input("Enter second number : "))
```

```
29.         ans = n1/n2
30.         print(n1 , " / " , n2 , " = " , ans)
31.
32.   else:
33.         print("Invalid choice")
```

OUTPUT :

```
Select Operation :
0. Exit
1. Add
2. Subtraction
3. Multiply
4. Divide
Select operations from 0,1,2,3,4 : 1
Enter first number : 5
Enter second number : 3
5  +  3  =  8

Select Operation :
0. Exit
1. Add
2. Subtraction
3. Multiply
4. Divide
Select operations from 0,1,2,3,4 : 3
Enter first number : 4
Enter second number : 5
4  *  5  =  20

Select Operation :
0. Exit
1. Add
2. Subtraction
3. Multiply
4. Divide
Select operations from 0,1,2,3,4 : 0
Thank you
```

Run Succeeded | Time 60 ms | Peak Memory 7.4M | Symbol | Tabs: 4 | Line 10, Column 23

C. Ask user to input different type of data (int, float, string) and convert them in other formats.

```
1. intType = int(input("Enter int type data : "))
2. print("Int Type\t\tFloat Type\t\tString Type\t")
3. print(intType,"\t\t\t", float(intType), "\t\t\t",
   str(intType))
4.
5. floatType = float(input("Enter float type data : "))
6. print("Int Type\tFloat Type\tString Type\t")
7. print(int(floatType),"\t\t\t", floatType, "\t\t\t",
   str(floatType))
8.
9. stringType = input("Enter String type data : ")
10.print("Int Type\t\tFloat Type\t\tString Type\t")
11.print(int(stringType),"\t\t\t", float(stringType),
   "\t\t\t", stringType)
```

OUTPUT :

```
Enter int type data : 123
Int Type        Float Type      String Type
123              123.0           123
Enter float type data : 123.123
Int Type    Float Type  String Type
123              123.123             123.123
Enter String type data : 123
Int Type        Float Type      String Type
123              123.0           123

Run Succeeded | Time 71 ms | Peak Memory 7.2M | Symbol | Tabs: 4 | Line 1, Column 1
```

D. Write a function areaTriangle that takes the lengths of three sides of the triangle as input parameters and returns the area of the triangle as an output. Also, assert that the sum of the length of any two sides is higher than the third side. Write a main function that accepts as command-line arguments and computes the area of a triangle using the function areaTriangle.

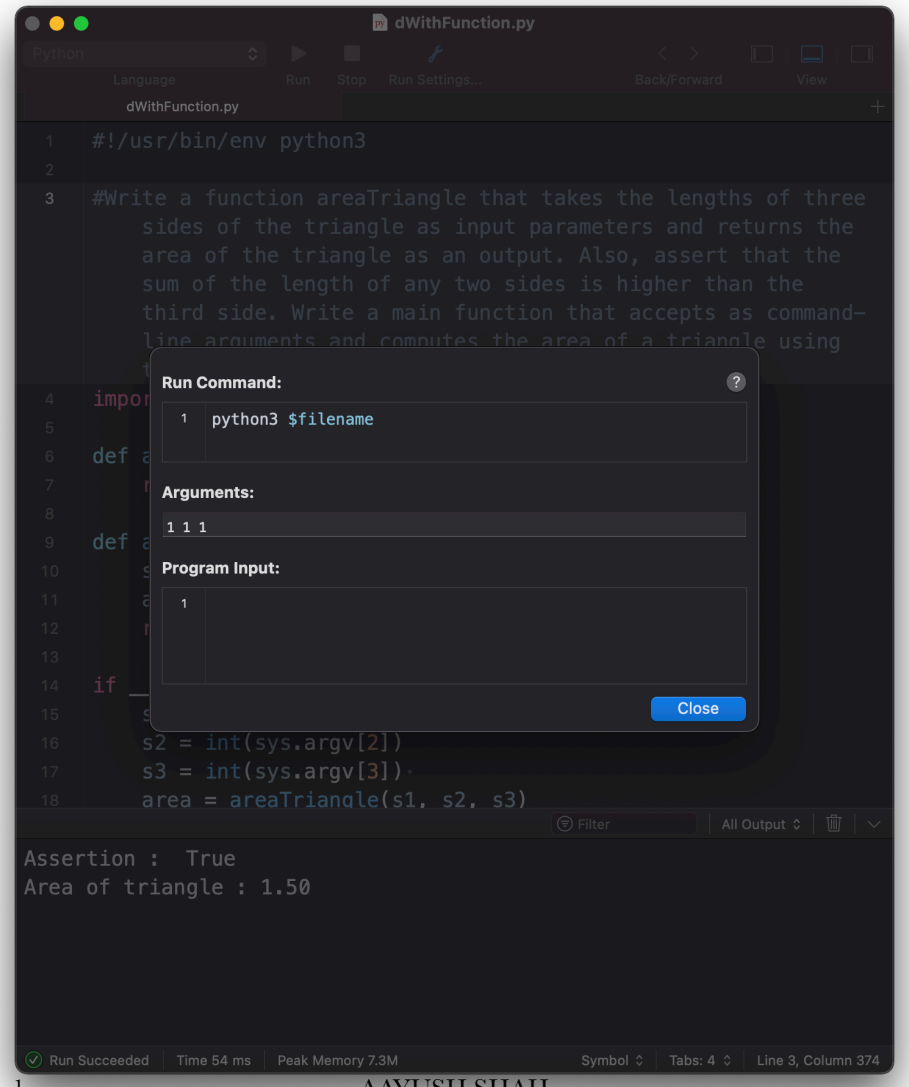
```python
import sys

def assertion(s1,s2,s3):
    return s1+s2>s3 or s1+s3>s2 or s2+s3>s1

def areaTriangle(s1,s2,s3):
    s = (s2 + s2 + s3) / 2
    area = (s*(s-s1)*(s-s2)*(s-s3)) ** 0.5
    return s

if __name__ == "__main__":
    s1 = int(sys.argv[1])
    s2 = int(sys.argv[2])
    s3 = int(sys.argv[3])
    area = areaTriangle(s1, s2, s3)
    print('Assertion : ' , assertion(s1, s2, s3))
    print('Area of triangle : %0.2f' %area)
```

OUTPUT :



E. Write a function that takes two numbers as input parameters and returns True or False depending on whether they are co- primes. Two numbers are said to be co-prime if they do not have any common divisor other than one.

```
#!/usr/bin/env python3

def findGCD(number1,number2):
    if(number2==0):
        return number1
    return findGCD(number2, number1%number2)
def isCoPrime(number1,number2):
    if(findGCD(number1, number2)==1):
        return True
    return False

#Driver code
number1 = int(input("Enter first number : "))
number2 = int(input("Enter second number : "))
if(isCoPrime(number1, number2)):
    print("The number ",number1," and ",number2," are CO-PRIME.")
else:
    print("The number ",number1," and ",number2," are NOT CO-PRIME.")

# example : 5 and 6 -> CO-PRIME
# example : 8 and 16 -> NOT CO-PRIME
```

OUTPUT :

```
Enter first number : 5
Enter second number : 6
The number  5  and  6  are CO-PRIME.
```

Run Succeeded | Time 80 ms | Peak Memory 7.3M | isCoPrime | Tabs: 4 | Line 21, Column 37

```
Enter first number : 8
Enter second number : 16
The number  8  and  16  are NOT CO-PRIME.
```

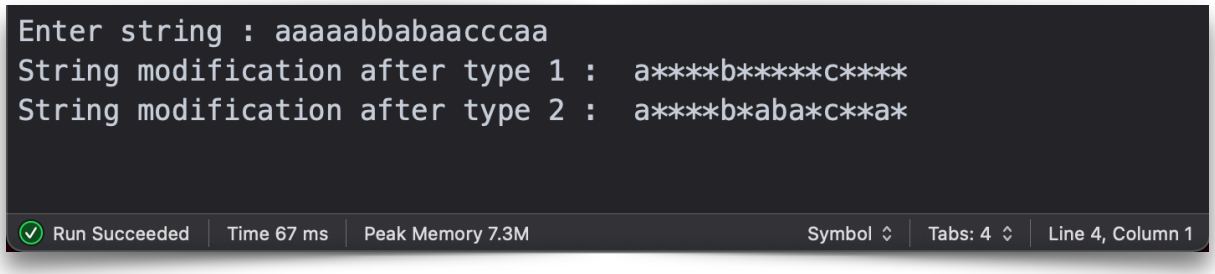
Run Succeeded | Time 74 ms | Peak Memory 7.3M | isCoPrime | Tabs: 4 | Line 21, Column 37

F. Write a function that takes a string as a parameter and returns a string with every successive repetitive character replaced with a star(*). For Example, 'balloon' is returned as 'bal*o*n'.

```
1. def modifyStringType1(original_string):
2.    '''
3.    If any repeated character is found in the string then it
   will be replaced by '*'
4.    '''
5.    alphabets = dict()
6.    new_string = original_string[0]
7.    for index in range(0,26):
8.         alphabets[chr(index+97)] = False
9.    alphabets[original_string[0]] = True
10.   for index in range(1,len(original_string)):
11.        if(alphabets[original_string[index]]==False):
12.             new_string += original_string[index]
13.             alphabets[original_string[index]] = True
14.        else:
15.             new_string += '*'
16.   return new_string
17.
18.def modifyStringType2(original_string):
19.   '''
20.   If any repeated consecutive character is found in the
   string then it will be replaced by '*'
21.   '''
22.   new_string = original_string[0];
23.   for index in range(1,len(original_string)):
24.
   if(original_string[index]==original_string[index-1]):
25.             new_string += '*'
26.        else:
27.             new_string += original_string[index]
28.   return new_string
29.
30.#Driver code
31.if __name__ == "__main__":
32.   original_string = str(input("Enter string : "))
```

```
33.  print("String modification after type 1 :
   ",modifyStringType1(original_string.lower()))
34.  print("String modification after type 2 :
   “,modifyStringType2(original_string.lower()))
```

OUTPUT :



G. Write a function that takes a number as n input parameter and returns the corresponding text in words; for example, on input 452, the function should return ‘Four Five Two’. Use a dictionary for mapping to digits to their string representation.

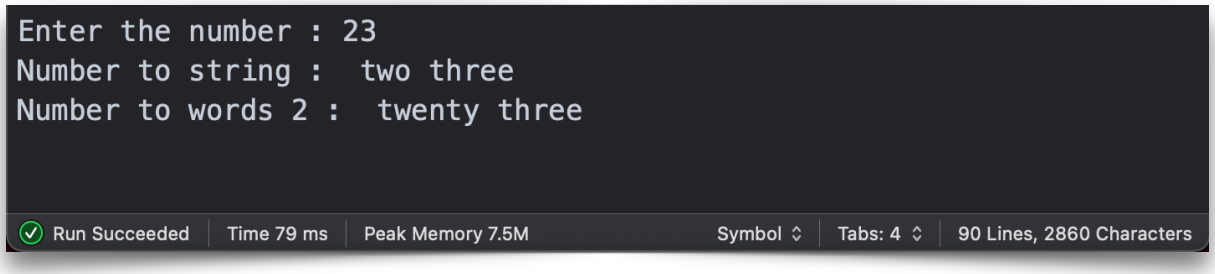
```
1. #!/usr/bin/env python3
2.
3. #Write a function that takes a number as n input parameter
   and returns the corresponding text in words; for example, on
   input 452, the function should return ‘Four Five Two’. Use a
   dictionary for mapping to digits to their string
   representation. [+ number to words]
4.
5. #from num2words import num2words
6.
7. """
8. the below function will only work after installing
   num2words.
9. We can easily install num2words using pip.
10.pip install num2words
11."""
12.
13.#def numberToWords(number):
```

```
14.# words_of_number = num2words(number)
15.# words_of_number = words_of_number.replace('-', ' ')
16.# return words_of_number
17.
18.def numberToWords2(number):
19.  lengthOfString = len(number)
20.
21.  digit = {'0': 'zero', '1': 'one', '2': 'two', '3':
   'three', '4': 'four', '5': 'five', '6': 'six', '7': 'seven',
22.            '8': 'eight', '9': 'nine'}
23.# place = {4: 'thousand', 3: 'hundred', 2: 'two', 1: ''}
24.  tens = {'0': '', '1': 'one', '2': 'twenty', '3':
   'thirty', '4': 'forty', '5': 'fifty', '6': 'sixty', '7':
   'seventy',
25.            '8': 'eighty', '9': 'ninety'}
26.  one = {'0': 'ten', '1': 'eleven', '2': 'twelve', '3':
   'thirteen', '4': 'fourteen', '5': 'fifteen', '6': 'sixteen',
27.            '7': 'seventeen', '8': 'eighteen', '9':
   'nineteen'}
28.
29.  if(lengthOfString==1):
30.       return digit[number]
31.  elif(lengthOfString==2):
32.       if(int(number)<20):
33.            return one[number[1]]
34.       else:
35.            if(int(number)%10==0):
36.                 return tens[number[0]]
37.            else:
38.                 return tens[number[0]] + " " +
   digit[number[1]]
39.  elif(lengthOfString==3):
40.       string1 = digit[number[0]] + " Hundred "
41.       if(int(number)<20):
42.            return string1 + " "  + one[number[1]]
43.       else:
44.            if(int(number)%10==0):
45.                 return string1 + " "  + tens[number[1]]
46.            else:
47.                 return string1 + " "  + tens[number[1]] +
   digit[number[2]]
48.  elif(lengthOfString==4):
49.       if(int(number)%1000):
```

```
50.             if(int(number)%100):
51.                 return digit[number[0]] + " Thousand "
52.             return digit[number[0]] + "Thousand"
53.
54.         string1 = digit[number[0]] + " Thousand " +
  digit[number[1]] + " Hundred "
55.         if(int(number)<20):
56.             return string1 + " "  + one[number[2]]
57.         else:
58.             if(int(number)%10==0):
59.                 return string1 + tens[number[2]]
60.             else:
61.                 return string1 + tens[number[2]] + " "  +
  digit[number[3]]
62.
63.
64.def numberToString(number):
65.   string_of_number = str()
66.   temp_number = number
67.   while(temp_number!=0):
68.         temp_rem = temp_number%10
69.         string_of_number += ""
70.         switcher  =  {
71.             0 : "zero",
72.             1 : "one",
73.             2 : "two",
74.             3 : "three",
75.             4 : "four",
76.             5 : "five",
77.             6 : "six",
78.             7 : "seven",
79.             8 : "eight",
80.             9 : "nine",
81.         }
82.         string_of_number = switcher.get(temp_rem) + " " +
  string_of_number
83.         temp_number=temp_number // 10
84.   return string_of_number
85.
86.if __name__ == "__main__":
87.   number = int(input("Enter the number : "))
88.   print("Number to string : ",numberToString(number))
89.# print("Number to words : ",numberToWords(number))
```

```
90.  print("Number to words 2 : ",numberToWords2(str(number)))
```

OUTPUT :



H. Write a recursive function that takes x value as an input parameter and print x-digit strictly in increasing number. [i.e. x = 6 than output 67891011]

```
1. #!/usr/bin/env python3
2. def printInc(x,count):
3.    if(count==0):
4.         return
5.    print(x, end="")
6.    printInc(x+1,count-1)
7.
8. #Driver code
9. if __name__ == "__main__":
10.   x = int(input("Enter number : "))
11.   printInc(x, x)
```

OUTPUT :