

Question 1

1. Write a program that prompts the user to enter the coefficients a , b and c from the quadratic equation $ax^2 + bx + c = 0$ and calculates the two real roots of the equation.
You may assume that the user enters values of a , b and c that lead to the equation having two distinct real roots, namely, the user will enter values such that $b^2 > 4ac$, and a will not equal 0. (1.5)

```
from math import *

def quadratic(a,b,c):
    assert ((a>0) and ((b*b-(4*a*c))>0)), "No real Roots "
    return [ ((b+sqrt(b*b - 4*a*c) )/2*a ) , ((b-sqrt(b*b - 4*a*c) )/2*a) ]

def inp():
    a=float(input("Enter a : "))
    b=float(input("Enter b : "))
    c=float(input("Enter c : "))

    print(quadratic(a,b,c)[0], " and ",quadratic(a,b,c)[1])

inp()

6.0 and 4.0
```

#Question 2

Write a program that prompts the user for the radius of a circle and calculates the area and perimeter of that circle and prints it out.

```
radius=float(input("Enter Radius : "))
area=3.14159*(radius*radius)
perimeter=6.28318*radius
print("Perimeter= ",perimeter," units \nArea= ",area," sq units")

Perimeter= 31.4159 units
Area= 78.53975 sq units
```

#Question 3

Write a program to use for loop to print the following reverse number pattern (1.5)
5 4 3 2 1

```
# 4 3 2 1
# 3 2 1
# 2 1
# 1
```

```
for x in range(5,0,-1):
    print()
    for y in range(x,0,-1):print(y,end=" ")
```

```
5 4 3 2 1
4 3 2 1
3 2 1
2 1
1
```

#Question 4

Write a program to print the following start pattern using the for loop (1.5)

```
# *
# * *
# * * *
# * * * *
# * * * * *
# * * * * *
# * * * *
# * * *
# * *
# *
```

```
for x in range(5):
    for y in range(x):
        print("*",end=" ")
    print()
for x in range(5,0,-1):
    for y in range(x,0,-1):
        print("*",end=" ")
    print()
```

```
*
* *
* * *
* * * *
* * * * *
* * * *
* * *
* *
*
```

```
# Question 5
# Explain all the functions which is applied on ( 2 + 2 = 4)
# a. List
# b. String
```

List

append() Adds an element at the end of the list

clear() Removes all the elements from the list

copy() Returns a copy of the list

count() Returns the number of elements with the specified value

extend() Add the elements of a list (or any iterable), to the end of the current list

index() Returns the index of the first element with the specified value

insert() Adds an element at the specified position

pop() Removes the element at the specified position

remove() Removes the first item with the specified value

reverse() Reverses the order of the list

sort() Sorts the list

STRING

capitalize() Converts the first character to upper case

casefold() Converts string into lower case

center() Returns a centered string

count() Returns the number of times a specified value occurs in a string

encode() Returns an encoded version of the string

endswith() Returns true if the string ends with the specified value

expandtabs() Sets the tab size of the string

find() Searches the string for a specified value and returns the position of where it was found

format() Formats specified values in a string

format_map() Formats specified values in a string

`index()` Searches the string for a specified value and returns the position of where it was found

`isalnum()` Returns True if all characters in the string are alphanumeric

`isalpha()` Returns True if all characters in the string are in the alphabet

`isascii()` Returns True if all characters in the string are ascii characters

`isdecimal()` Returns True if all characters in the string are decimals

`isdigit()` Returns True if all characters in the string are digits

`isidentifier()` Returns True if the string is an identifier

`islower()` Returns True if all characters in the string are lower case

`isnumeric()` Returns True if all characters in the string are numeric

`isprintable()` Returns True if all characters in the string are printable

`isspace()` Returns True if all characters in the string are whitespaces

`istitle()` Returns True if the string follows the rules of a title

`isupper()` Returns True if all characters in the string are upper case

`join()` Converts the elements of an iterable into a string

`ljust()` Returns a left justified version of the string

`lower()` Converts a string into lower case

`lstrip()` Returns a left trim version of the string

`maketrans()` Returns a translation table to be used in translations

`partition()` Returns a tuple where the string is parted into three parts

`replace()` Returns a string where a specified value is replaced with a specified value

`rfind()` Searches the string for a specified value and returns the last position of where it was found

`rindex()` Searches the string for a specified value and returns the last position of where it was found

`rjust()` Returns a right justified version of the string

`rpartition()` Returns a tuple where the string is parted into three parts

`rsplit()` Splits the string at the specified separator, and returns a list

`rstrip()` Returns a right trim version of the string

`split()` Splits the string at the specified separator, and returns a list

`splitlines()` Splits the string at line breaks and returns a list

`startswith()` Returns true if the string starts with the specified value

`strip()` Returns a trimmed version of the string

`swapcase()` Swaps cases, lower case becomes upper case and vice versa

`title()` Converts the first character of each word to upper case

`translate()` Returns a translated string

`upper()` Converts a string into upper case

`zfill()` Fills the string with a specified number of 0 values at the beginning