

Q1. WAP to find the roots of a quadratic equation.

```
import math

a = int(input("a = "))
b = int(input("b = "))
c = int(input("c = "))

discriminant = b ** 2 - 4 * a * c

if discriminant > 0:
    discriminantSqrt = math.sqrt(discriminant)
    firstRoot = (-b + discriminantSqrt) / (2 * a)
    secondRoot = (-b - discriminantSqrt) / (2 * a)
    print(firstRoot, secondRoot)
elif discriminant == 0:
    discriminantSqrt = math.sqrt(discriminant)
    root = (-b + discriminantSqrt) / (2 * a)
    print(root)
else:
    print("Quadratic Equation has complex roots")
```

Q2. WAP to accept a number 'n' to compute the following:

- a. Check if 'n' is prime
- b. Generate all prime numbers till 'n'
- c. Generate first 'n' prime numbers
- d. Calculate the sum of first 'n' natural numbers

```
n = int(input("Enter n = "))

#1. Check if 'n' is prime
if n == 2:
    print("'n' is prime.")
elif n%2 == 0:
    print("'n' is not prime")
else:
    divisor = 3
    isPrime = True
    while (divisor <= n/2):
        if (n % divisor == 0):
            isPrime = False
            break
        divisor+=2
    if (isPrime):
        print("'n' is prime.")
    else:
        print("'n' is not prime.")

#2. Generate all prime numbers till 'n'
print("Prime numbers till 'n'")
print(2)
number = 2
while(number <= n):
    number+=1
    if number%2==0:
        continue
    else:
        divisor = 3
        isPrime = True
        while (divisor <= number/2):
            if (number % divisor == 0):
                isPrime = False
                break
            divisor+=2
        if(isPrime):
```

```
print(number)
```

#3. Generate first 'n' prime numbers

```
print("First 'n' prime numbers")
print(2)
number = 2
count = 1
while (count < n):
    number+=1
    if (number%2 == 0):
        continue
    else:
        divisor = 3
        isPrime = True
        while (divisor <= number/2):
            if (number % divisor == 0):
                isPrime = False
                break
            divisor+=2
        if(isPrime):
            print(number)
            count+=1
```

#4. Calculate sum of first 'n' natural numbers

```
print("Sum of first 'n' natural numbers")
number = 1
sum = 0
for i in range(n):
    sum+=number
    number+=1
print(sum)
```

Q3. WAP to create a pyramid of the character '\*' and a reverse pyramid

```

    *

  ***

****

*****

*****

****

***

*
```

```
n = int(input("Enter the number of stars in Pyramid(odd) : "))

spaces = " " * int(n/2)
counter = 1
# Upper pyramid
for i in range(1, n+1, 2):
    print(spaces + ("*" * i))
    spaces = " " * (int(n/2) - counter)
    counter += 1

# Reverse pyramid
counter = 1
for i in range(n-2, 0, -2):
    spaces = " " * counter
    print(spaces + ("*" * i))
    counter += 1
```

Q4. WAP that accepts a character and performs the following:

- a. Print whether the character is a letter or numeric digit or a special character
- b. If the character is a letter, print whether the letter is uppercase or lowercase
- c. If the character is a numeric digit, prints its name in text(e.g., if input is 9, output is NINE)

```
characterInput = input("Enter a Character : ")
isInputLetter = False
isInputDigit = False

if characterInput.isalpha():
    isInputLetter = True
    print(characterInput + " is a Letter.")
elif characterInput.isnumeric():
    isInputDigit = True
    print(characterInput + " is a Numeric Digit.")
else:
    print(characterInput + " is a Special Character.")

if isInputLetter:
    if characterInput.isupper():
        print("Entered character is in Upper Case")
    else:
        print("Entered character is in Lower Case")

digitToNameMapping = {0: "Zero", 1: "One", 2: "Two", 3: "Three", 4: "Four", 5:
"Five", 6: "Six", 7: "Seven", 8: "Eight",
9: "Nine"}

if isInputDigit:
    print(characterInput + " in words is " +
digitToNameMapping[int(characterInput)])
```

Q5. WAP to perform the following operations on a string:

- a. Find the frequency of a character in a string.
- b. Replace a character by another character in a string.
- c. Remove the first occurrence of a character from a string.
- d. Remove all occurrences of a character from a string.

```
strInput = input("Enter the String : ")
calculateFrequencyCharacter = input("Enter the character whose frequency is to be
Calculated : ")
```

```
# Frequency of each character in String
print("Frequency of " + calculateFrequencyCharacter + " Each Char present in the
input String : " + str(strInput.count(calculateFrequencyCharacter)))
```

```
# Replace each occurrence of character with new one
replaceCharacter = input("Enter the character to be replaced in the string : ")
newCharacter = input("Enter the new Character : ")
strInputCopy = strInput
strInputCopy = strInputCopy.replace(replaceCharacter, newCharacter)
print(strInputCopy)
```

Q6. WAP to swap first n characters of two strings.

```
strOne = input("Enter the first String : ")
strTwo = input("Enter the second String : ")

n = int(input("Length of the string to be swapped : "))

if n <= min(len(strOne), len(strTwo)):
    temp = strOne[0: n]
    strOne = strOne.replace(temp, strTwo[0: n])
    strTwo = strTwo.replace(strTwo[0: n], temp)
    print("First String :", strOne)
    print("Second String :", strTwo)
else:
    print("Wrong Input!!")
```