Series of Prime number

6)
```
def prime (num):
    for i in range (2,(num//2)+1):
        if num%i ==0:
            return False
    return True
for i in range(2,11):
    ans = prime(i)
    if ans ==True:
        print(i)
```
O/P:- 2
3
5
7

nth prime number

7)
```
def prime(num):
    for i in range(2,(num//2)+1):
        if num%i ==0:
            return False
    return True
count =0
i=2
num = 10
while count != num:
    ans = prime(i)
    if ans == True:
        count +=1
    i+=1
print(i-1)
```
o/p :- 29

8) Prime nu
def
if

8) Prime number or not check by using function.

```
def Prime(num):
    if num==1 or num==0:
        return False
    for i in range(2, num//2+1):
        if num%i==0:
            return False
    else:
        return True
num = 13
answer = prime(num)
if answer:
    print("prime number")
else:
    print("not an prime number")
```

★ Conversion Programs :-

1) Binary → 0, 1 → 0b
2) Octal → 0, 7 → 0o
3) Decimal → 0, 9 → by default it will be decimal only
4) Hexadecimal → 0, 9, A–F → 0x values

★ Representation of different number System in Python. :-

```
binary = 101
print(binary)
binary = 0b101
print(binary)
octal = 25
print(octal)
octal = 0o25
print(octal)
hexa = 0xd
print(hexa)
```

O/P :-
5
25
21
13

1) WAP to convert decimal number to binary number

Steps :-

1) Extract the remainder of the given number with 2
2) Store the remainder in the output variable
3) make the quotient as the next number
4) Repeat step1 to step3 till num != 0 and stop when num == 0
5) return the stored remainders

⟹

```
def decitoBin(num):
    out = ""
    while num != 0:
        rem = ~~str(num) %2~~ num%2
        out = str(num) + out
        num = num//2
    return int(out)
num = 25
print("input=", num)
a = decitoBin(num)
print("output=", a)
```

2) decimal to Hexadecimal

⟹

```
def decitoHexa(num):
    out = ""
    while num != 0:
        rem = num%16
        if rem > 9:
            rem = chr(55 + rem)
        out = str(rem) + out
        num = num//16
    return out
num = 183
print("input=", num)
```

a = deci
print("

A 6
B
C
D
E
F

3) def f

16 8 4 2 1
1 1 1 1

a = decitoHexa(num)
print("output =", a)

A 65 → 10 + 55 = 65
B 66 → 11 + 55 = 66
C 67 → 12 + 55 = 67
D 68 → 13 + 55 = 68
E 69 → 14 + 55 = 69
F 70 → 15 + 55 = 70

3) def decimaltoany(num, ns):
    out = ""
    while num != 0:
        rem = num % ns
        if rem > 9:
            rem = chr(55 + rem)
        out = str(rem) + out
        num = num // ns
    return out

num = 183
print("Input =", num)
a = decimaltoany(num, 2)
print("output =", a)

4) WAP to convert Binary number to decimal number

Steps:

1) Extract last digit
2) The last digit should be multiplied with temp variable
   add it to output variable.
3) multiply temp variable by 2.
4) eliminate the last digit
5) Repeat Step1 to Step4 till num != 0 Stop when num==0
   (while)
6) return out

```
def binarytodecimal(num):
    out = 0
    temp = 1
    while num != 0:
        last = num%10
        out = out + last * temp
        temp = temp * 2
        num = num//10
    return out
num = 1100
print("Input =", num)
a = binarytodecimal(num)
print("output =", a)
```

5) octal to decimal conversion

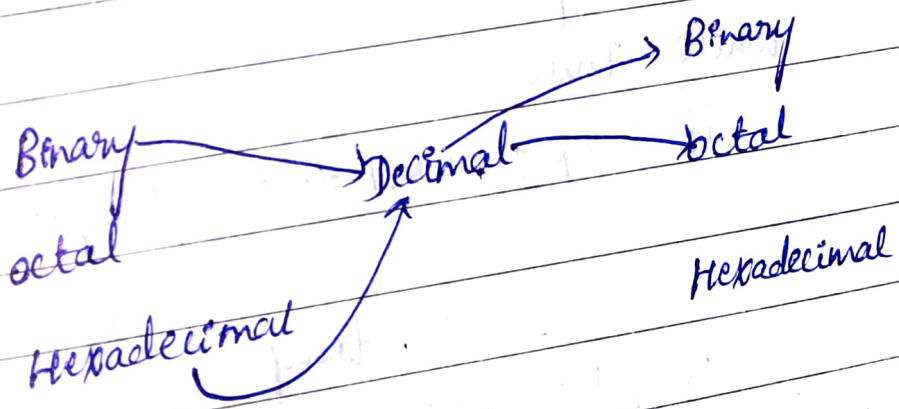
```
def octadeci(num):
    out = 0
    temp = 1
    while num != 0:
        rem = num%10
        out = out + rem * temp
```

6) WAP to conv

```
def hexad
```

6) WAP to convert hexadecimal number to decimal number

```
def hexadecitodeci(num):
    n = len(num)
    num1 = "0123456789ABCDEF"
    n = len(num)
    out = 0
    temp = 1
    for i in range(n-1, -1, -1):
        out += s2.index(s1[i]) * temp
        temp *= 16
    return out
ans = hexadecitodeci("A4C")
print(ans)
```



1) WAP to determine LCM of 2 numbers.

⟹
```
def isLCM(num1, num2):
    tempLCM = num2
    if num2 > num1
        tempLCM = num2
max = tempLCM
while True:
    if tempLCM % num1 == 0 and tempLCM %
                                num2 == 0:
        return tempLCM
    tempLCM += max
ans = isLCM(30, 45)
print(ans)
```
O/P :- 90

2) WAP to determine LCM of 3 numbers

⟹
```
def isLCM( num1, num2, num3):
    temp = num1
    if num2 > num1 and num2 > num3:
        temp = num2
    else:
        num3 > num1 and num3 > num2:
            temp = num3
    max = temp
    while True:
        if temp % num1 == 0 and temp % num2 and
                                temp % num3:
            return temp
        temp += max
ans = isLCM(2, 4, 10)
print(ans)
```
O/P :- 20

3) WAP to display first 10 fibonanaj Series.

→
```
def isfibo(count)
    num1 = 0
    print(num1)
    num2 = 1
    print(num2)
    for i in range(2, count):
        num3 = num1 + num2
        print(num1)
        num1 = num2
        num2 = num3
isfibo(10)
```

(or)

```
def isfibo(count)
    num1 = 0
    num2 = 1
    for i in range(count):
        num3 = num1 + num2
        print(num1)
        num1 = num2
        num2 = num3
isfibo(10)
```

o/p :- 0 1 1 2 3 5 8 13 21 34

4) WAP to reverse a given integer number.

→
```
def reverse(num):
    out = 0
    while num != 0:
        last = num % 10
        out = out * 10 + last
        num = num // 10
    return out
```

```
answer = reverse(2456)
print(answer)

def reverse(num):
    temp = num
    if num<0:
        num *= -1
    out = 0
    while num!=0:
        last = num%10
        out = out*10+last
        num//=10
    if temp>=0:
        return out
    else:
        return out*-1
ans = rev(-2561)
print(ans)
```

List Programs

1) o/p :- [2, 6, 8]

→
```
def fact(num):
    out=1
    for i in range(num, 1, -1):
        out *= i
    return out
s = [23, 45, 356, 76, 87, 1, 2]
answer = fact(len(s))
print(answer-1)
```

o/p: 5039

1) L1=[2,4,6,8]
print(L1) o/p: [2,4,6,8]

L2 = [None] * len(L1)
print(L2)
L2[0] = 23
L2[3] = 15
print(L2) o/p:- [None, None, None, None]
[23, None, None, 15]

2) L1 = [2,4,6,8] o/p:- [2,4,6,8]
print(L1)

L2 = [None] * len(L1) o/p:- [2,4,6,8]
for i in range(len(L1)):
    L2[i] = L1[i]
print(L2)

1) WAP to display missing numbers in the list
L1 = [12, 16, 19, 25]

⟹ L1 = [12, 16, 19, 25]
for i in range(len(L1)-1):
    for k in range(L1[i]+1, L1[i+1]):
        print(k)

Logic :-
i = 0
Start → L1[i]+1 ⟹ 12+1 = 13
end → L1[i+1]-1 ⟹ 16-1 = 15
i = 1
Start → L1[i]+1 → 17
end → L1[i+1]-1 → 18
i = 2
Start → L1[i]+1 → 20
end → L1[i+1]-1 → 24

2) WAP to Sort the list in ascending order.

⟹ i → (len(l1)-1) → (5,1) → 4

i = (0, 1, 2, 3)

i = 0

f (1, 5) → (1, 2, 3, 4)

f = 1

if (12 > 8)

temp = 12

l1[0] = l1[1] = 8

l1[1] = 12

f = 2

if (8 > 3) true Swap

f = 3

if (3 > 14) false

f = 4

3 > 5 false

⟹

```
l1 = [12, 8, 3, 14, 5]
print("input =", l1)
for i in range(len(l1)-1):
    for j in range(i+1, len(l1)):
        if l1[i] > l1[j]:
            temp = l1[i]
            l1[i] = l1[j]
            l1[j] = temp

print("output", l1)
```

3) The first line of the i/p consists of 2 space seperated integers num & random. Representing the numbers of flower sticks. and the number to given by rosy to emma respectively. The 2nd line consists of n space seperated integers. Representing the length of flower sticks.

Output :-

Print n space seperated integers in the final arrangement of the flower sticks in the bouquet

i/p → 11 7 5 10 46 23 16 8

o/p: → 5 7 11 46 23 16 10 8

⟹

```
l1 = [11, 7, 5, 10, 46, 23, 16, 8]
print ("Input=", l1)
random = 5
for i in range (random):
        for j in range (i+1, random):
                if l1[i] > l1[j]:
                        temp = l1[i]
                        l1[i] = l1[j]
                        l1[j] = temp
for i in range (random, len(l1)-1):
        for j in range (i+1, len(l1)):
                if l1[i] < l1[j]:
                        temp = l1[i]
                        l1[i] = l1[j]
                        l1[j] = temp
print ("Output =" l1)
```

4) WAP to reverse the list elements without using inbuilt method.

```
s = [1, 2, 3, 4, 5]
print(s)
j = len(s) - 1
for i in range(len(s)//2):
        temp = s[i]
        s[i] = s[j]
        s[j] = temp
        j -= 1
print(s)
```

5) WAP to reverse the string

```
s1 = "reverse"
print(s1)
s = list(s1)
j = len(s) - 1
for i in range(len(s)//2):
        temp = s[i]
        s[i] = s[j]
        s[j] = temp
        j -= 1
res = ''
for i in s:
    res += i
print(res)
```

7)
```
l1 = [ ]
size = int(input("enter the size of list"))
for i in range(size):
    a = int(input("enter the value"))
    l1 = l1 + [a]
print(l1)
j = len(l1)-1
for j in range(len(l1)//2):
    temp = l1[i]
    l1[i] = l1[j]
    l1[j] = temp
    j -= 1
print(l1)
```

8)
```
n = int(input("enter the number of lines"))
l2 = [[ ]] * n
for k in range(n):
    d = int(input("enter the no. of integers in the line"))
    l1 = [None] * d
    for i in range(d):
        a = int(input("enter the integer in list"))
        l1[i] = a
    l2[k] = l1
print(l2)
```

9)
```
q = int(input("enter the number of querries"))
for j in range(q):
    x = int(input("enter the line no."))-1
    y = int(input("enter the range"))-1
    if x < len(l2) and y < len(l2[x2]):
        print(l2[x][y])
    else:
        print("Error")
```

10) WAP to insert an element at the specified pos in the given list.

⟹
```
l1 = [12,8,7,46,8]
l2 = ((len(l1)+1) * None)
ele = 100
pos = 2
if pos > len(l1):
    print("cannot perform the specified operation")

else:
    l2 = [None] * (len(l1)+1)
    l2 [pos] = ele
    for i in range(pos):
        l2[i] = l1[i]
    for j in range(pos+1, len(l1)):
        l2[i] = l1[i-1]
    print(l2)
```

11) WAP to delete an element at the specified positio

⟹
```
l1 = [12,8,7,46,8]
pos = 2
if pos > len(l1):
    print("cannot perform the operation")
else:
    l2 = [None] * (len(l1)-1)
    for i in range(pos):
        l2[i] = l1[i]
    for j in range(pos, len(l2)):
        l2[i] = l1[i+1]
    print(l2)
```

ascending → print(l1[len(l1)-1])
descending → print(l1[n-1])

1)
```
n = int(input("enter the number of contacts"))
d2 = {}
for i in range(n):
    key = input("enter the name")
    value = int(input("enter the number"))
    d1 = {key: value}
    d2 = {**d2, **d1}
print(d2)
q = int(input("enter the number of querries"))
for i in range(q):
    name = input("enter the name")
    if name in d2:
        print(name, "=", d2[name])
    else:
        print("not found")
```

2)
```
n = int(input("enter the number of contacts"))
d2 = {}
for i in range(n):
    key = input("enter the name")
    value = int(input("enter the number"))
    d2[key] = value
print(d2)
```

Prmp 3) WAP to determine nth maximum number in the given list.

→
```
l1 = [11, 7, 5, 10, 46, 23, 16, 8]
print("input =", l1)
n = int(input("enter which maximum you want"))
for i in range(len(l1)-1):
    for j in range(i+1, len(l1)):
        if l1[i] > l1[j]:
```

```
                    temp = L1[i]
                    L1[i] = L1[j]
                    L1[j] = temp
print(L1[n-1])          # nth min number
print(L1[len(L1)-n])    # nth max number
```

4) WAP to determine the first maximum number along with it's index.

⟹

```
L1 = [12, 18, 3, 13, 5]
max1 = L1[0]
index = 0
for i in range(1, len(L1)):
        if max1 < L1[i]:
                max1 = L1[i]
                index = i
print("Value =", max1)
print("index =", index)
```

5) WAP to determine the first two maximum numbers and their index.

⟹

```
L1 = [12, 18, 3, 13, 5]
max1 = L1[0]
index1 = 0
max2 = L1[1]
index2 = 1
for i in range(1, len(L1)):
        if max1 < L1[i]:
                max2 = max1
                index2 = index1
                max1 = L1[i]
                index1 = i
        elif max2 < L1[i]:
```

```
        max2 = l1[i]
        index2 = i
print ("value =", max1)
print ("index1 =", index1)
print ("value2 =", value)
print ("index2 =", index2)
```

1) WAP to delete repeated elements in the list.

→
```
l1 = [3, 3, 34, 343, 3]
print (l1)
# nested while loop for comparision
start = 0
n = len(l1)
while start < n-1 :
        j = start+1
        while j < n :
            if l1[start] == l1[j]: # equality comp.
                # logic for delete
                # shifting the elements in then list
                i = j  # Initialization i value.
                while i < n-1 :
                    l1[i] = l1[i+1]
                    i += 1
                n -= 1 # ignore the last shifted index
                j -= 1 # next comparision should
                        start
            j += 1 # last line of inner while loop.
        start += 1 # last line of outer while loop
l2 = [NONE] * n
for i in range(n):
        l2[i] = l1[i]
print(l2)
```

2) WAP to count the number of repeated elements in the list.

⟹

```
l1 = [2,3,2,1,3]
print ("input =", l1)
n = len(l1)
Start = 0
while Start < n :
        Count = 0
        j = Start + 1
        while j < n :
            if l1[Start] == l1[j]
                Count += 1
                i = j
                while i < n-1 :
                    l1[i] = l1[i+1]
                    i += 1
                n -= 1
                j -= 1
            j += 1
        print (l1[Start], "is repeated for", count, "times")
        Start += 1
```

3) WAP to determine the sum of non repeated elements in the list.

⟹

```
l1 = [2,6,3,2,1,3,12,12,6,8,7]
print ("input =", l1)
n = len(l1)
Start = 0
Sum1 = 0
while Start < n :
        count = 0
        j = Start + 1
```

2) WAP to count the number of repeated elements in the list.

⟹
```
l1 = [2,3,2,1,3]
print ("input =", l1)
n = len(l1)
Start = 0
while Start < n :
        count = 0
        j = Start + 1
        while j < n :
            if l1[Start] == l1[j]
                count += 1
                i = j
                while i < n-1 :
                    l1[i] = l1[i+1]
                    i += 1
                n -= 1
                j -= 1
            j += 1
        print (l1[Start], "is repeated for", count, "times")
        Start += 1
```

3) WAP to determine the sum of non repeated elements in the list.

⟹
```
l1 = [2,6,3,2,1,3,12,12,6,8,7]
print ("input =", l1)
n = len(l1)
Start = 0
Sum1 = 0
while Start < n :
        count = 0
        j = Start + 1
```

```
while j<n :
    if l1[start] == l1[j] :
        count += 1
        i = j
        while i<n-1 :
            l1[i] = l1[i+1]
            i += 1
        n -= 1
        j -= 1
    j += 1
if count == 0 :
    print(l1[start])
    sum1 += l1[start]
start += 1
print("sum=", sum1)
```

4)
```
l1 = [2,2,2,1,2,1,1]
n = len(l1)
start = 0
while start<n :
    count = 1
    j = start + 1
    while j<n :
        if l1[start] == l1[j] and l1[start] ==
                                    l1[start+1] :
            count += 1
            i = j
            while i<n-1 :
                l1[i] = l1[i+1]
                i += 1
            n -= 1
            j -= 1
        j += 1
    print(l1[start], "is repeated for", count,
                                    "times")
```

5) s1 = "aabbaccddab"
l1 = list(s1)
Same as 4th program

6) l1 = ["hello", "hi", "hello"]
print("Input=", l1)
d1 = { }
:
d1[l1[start]] = count
start += 1
print(d1)

7) n number of people participated in a coding marathon where they were asked to solve some problems. each problem carried one mark and at the end of marathon the total marks that each person achieve was calculated. as an organizer you have the list of total marks that each person achieve you have to calculate the sum of marks of top k scorers from the list.

→ i/p Specification
1) n → no. of people in compitition
2) K → top scored
3) array of length N with the scores of all N participants.

→ o/p Specification
return sum of the marks of top k scorers from the list.

Test case
i/p 1) 4 3) {4, 1, 2, 5} | o/p :- 9
2) 2

2) i/p 1) 4
2) 3
3) {4, 3, 6, 1}
o/p :- 13.

⟹
```
n = 4
K = 2
l1 = [4, 1, 2, 5]
for i in range(len(l1)-1):
    for j in range(i+1, len(l1)):
        if l1[i] < l1[j]
            l1[i], l1[j] = l1[j], l1[i]

sum1 = 0
for i in range(k):
    sum1 += l1[i]
print(sum1)
```

3)
```
s1 = "Python is easy"
l1 = list(s1)
print(l1)
l2 = []
i = 0
while i < len(l1):
    temp = ""
    while i < len(l1) and l1[i] != ' ':
        temp += l1[i]
        i += 1
    l2 = l2 + [temp]
    i += 1
print(l2)
```

1) WAP to reverse each word in a given sentence.

⟹
```
s1 = "This is a program"
l1 = list(s1)
print("input=", s1)
i = 0
output = ""
while i < len(l1):
        tempword = ""
        while i < len(l1) and l1[i] != ' ':
                tempword = l1[i] + tempword
                i += 1
        output += tempword + " "
        i += 1
print("output = ", output)
```

2) o/p :- Program a is this

⟹
```
s1 = "This is a program"
l1 = list(s1)
print("input=", s1)
i = 0
output = ""
while i < len(l1):
        tempword = ""
        while i < len(l1) and l1[i] != ' ':
                tempword += l1[i]
                i += 1
        output = tempword + " " + output
        i += 1
print("output=", output)
```

1) WAP to reverse each word in a given sentence

```
s1 = "This is a program"
l1 = list(s1)
print("input = ", s1)
i = 0
output = ""
while i < len(l1):
        tempword = ""
        while i < len(l1) and l1[i] != ' ':
            tempword = l1[i] + tempword
            i += 1
        output += tempword + " "
        i += 1
print("output = ", output)
```

2) o/p :- Program a is this

```
s1 = "This is a program"
l1 = list(s1)
print("input=", s1)
i = 0
output = ""
while i < len(l1):
        tempword = ""
        while i < len(l1) and l1[i] != ' ':
            tempword += l1[i]
            i += 1
        output = tempword + " " + output
        i += 1
print("output = ", output)
```

3) s1 = "This is a program"
l1 = list(s1)
print("input =", s1)
i = 0
output = ""
while i< len(l1):
    count = 0
    tempword = ""
    while i< len(l1) and l1[i]!= ' ':
        count += 1
        tempword += l1[i]
        i+=1
    if count != 0:
        output += tempword + str(count)
    i+=1
print("output=", output)

4) i/p :- "Py3th406n i6s E4a2814"
o/p :- "Python13 is6 easy7"
→ s1 = "Py3th406n i6s e4a2814"
l1 = list(s1)
print(l1)
out = ""
i = 0
while i< len(l1):
    sum1 = 0
    tempword = ""
    while i< len(l1) and l1[i]!= ' ':
        if '0' <= l1[i] <= '9':
            sum1 += int(l1[i])
        else:
            tempword += l1[i]
        i+=1
    out += tempword + str(sum1) + " "

i += 1
print (out)

5) i/p → "Python is a programming language"
o/p → "Python si a gnimmargorp language"

6) i/p → Amit works in accolite and accolite is good
o/p → Amit accolite in works and works is good.

7) i/p → python is easy language
o/p → PYTHON IS EASY language

8) WAP to Convert numbers to words.
i/p :- 123456789
o/p :- Twelve crore
thirty four lakh
fifty six thousand
Seven hundred eighty nine

⟹
```
def display (num, str):
    l1 = ["", "one", "Two", "three", "four", "five", "Six",
    "Seven", "eight", "nine", "ten", "eleven", "twelve",
    "thirteen", "fourteen", "fifteen", "Sixteen", "Seventeen",
    "eighteen", "nineteen"]
    l2 = ["", "", "twenty", "thirty", "fourty", "fifty",
    "Sixty", "Seventy", "eighty", "ninty"]
    if n>19:
        print (l2[num//10], " ", l1[num%.10])
    else:
        print (l1)
    if num != 0 :
        print(l1)
num = 123456789
display((num//10000000)," crore")
```

display((n
display((
display(
display(

9) WAP
⟹ def

3

10)

⟹

```
display((num//100000)%1000),"lakh")
display((num//1000)%100),"thousand")
display(num//100)%10),"hundred")
display(num//100),' ')
```

9) WAP to remove the spaces from a given string

⟹
```
def removeSpace(s1):
    out = ""
    for i in s1:
        if i != " ":
            out += i
    return out
s1 = "this is python"
print("input =", s1)
output = removeSpace(s1)
print("output =", output)
```

10) WAP to convert all the characters in a string to uppercase without using inbuilt function.

⟹
```
s1 = "Apython"
out = ""
for i in s1:
    if 'a' <= i <= 'z':
        out += (chr(ord(i)-32)
    else:
        out += i
print(out)
```

11) WAP to determine whether the given 2 strings are anagram or not.

→ 1) remove Space
2) Compare length
   if != Stop
   if == go to 3
3) convert to case
4) Sort
5) compare

```
def removeSpace(s1):
    out = ""
    for i in s1:
        if i != " ":
            out += i
    return out
def uppercase(s1):
    out = ""
    for i in s1:
        if 'a' <= i <= 'z':
            out += chr(ord(i)-32)
        else:
            out += i
    return out
def Sorting(s1):
    l1 = list(s1)
    for i in range(len(l1)-1):
        for j in range(i+1, len(l1)):
            if l1[i] > l1[j]
                temp = l1[i]
                l1[i] = l1[j]
                l1[j] = temp
    return l1
```

```
s1 = "Hilter women"
s2 = "Mother in law"
s1 = removespace(s1)
s2 = removespace(s2)
print("input", s1)
print("input", s2)
print("after removing space")
if len(s1) != len(s2):
        print("String are not anagram")
else:
    s1 = uppercase(s1)
    s2 = uppercase(s2)
    print("after converting to upper case")
    print("input =", s1)
    print("input =", s2)
s1 = sorting(s1)
s2 = sorting(s2)
print("after sorting")
print("input", s1)
print("input", s2)
if s1 == s2:
        print("Strings are anagram")
else:
        print("Strings are not anagram")
```

12) WAP to count the number of words in a given String

→ I/p :- "this is string"

o/p :- 3

```
s1 = "this is string"
count = 1
if s1[0] == " ":
        count = 0
```

```
    if s1[i] == " " and s1[i+1] != " ":
        count += 1
print(count)
```

13)
```
s = "Py$th@n12#4@gmail"
count1 = 0
count2 = 0
count3 = 0
for i in s:
    if '0' <= i <= '9':
        count1 += 1
    elif 'a' <= i <= 'z' or 'A' <= i <= 'Z':
        count2 += 1
    else:
        count3 += 1
print("numeric :", count1)
print("alpha :", count2)
print("Special :", count3)
```

14) s = "karnataka@225"

⟶
```
rep = ""
out = ""
for i in s:
    if '0' <= i <= '9':
        out += i
        res = int(out) + 100
print(rep)
```

15)
```
s = "Delhi@840"
out = ""
rep = ""
for i in s:
    if '0' <= i <= '9':
```

16)
```
s1 = "this is a
s2 = "nga"
l1 = list(s1)
l2 = list(s
for i in ra
```

```
out +=
rep =
print(rep)
```

whether substring

```
else:
```

17)

```
        out += i
        rep = int(out) + 100
print(rep)
```

whether substring a is present or not.

16)
```
s1 = "this is a string"
s2 = "nga"
l1 = list(s1)
l2 = list(s2)
for i in range(len(l1)):
        k = i
        j = 0
        while k < len(l1) and j < len(l2) and
                                l1[k] == l2[j]:
                k += 1
                j += 1
        if j == len(l2):
                print("Yes it is found")
                break
else:
        print("no not found")
```

17)
```
l2 = list(s2)
count = 0
for i in range(len(l1)):
        k = i
        j = 0
        while k < len(l1) and j < len(l2) and
                                l1[k] == l2[j]:
                k += 1
                j += 1
        if j == len(l2):
                print("found in between", i, "and",
                                (k-1))
                count += 15
```

```
if count != 0 :
    print("it is present", count, "times")
else :
    print("it is not found")
```

18)

```
s1 = "Computer"
s2 = "Program"
ref = ""
for i in s1 :
    c = 0
    for j in s2 :
        if i == j :
            c += 1
            break
    if c == 0 :
        ref += i
print(ref)
```

```
s1 = "education"
s2 = "aeiouAEIOU"
```

19) WAP to remove vowels from the given string.

→

```
s1 = "education"
s2 = "aeiouAEIOU"
ref = ""
for i in s1 :
    c = 0
    for j in s2 :
        if i == j :
            ct = 1
            break
    if c == 0 :
        ref += i
print(ref)
```

20) if the 4 i/p numbers are 23792, 37221, 10270 and 73391 The smallest number that can be produced using 2 digits from each of these are 22, 12, 00 and 13 respectively and the sum of these smallest numbers will be 47 therefore the expected result is 47.

$\Rightarrow$ $\underset{22}{\underline{23792}}$ $\underset{12}{\underline{37221}}$ $\underset{00}{\underline{10270}}$ $\underset{13}{\underline{73391}}$ = 47

21) An array of size N denoting the required answer for each student

i/p 1 :- 6 o/p :- {0,0,1,3,4,0}

i/p 2 :- {4,9,5,3,2,10}

```
def least2(num):
    min1 = 10
    min2 = 10
    while num != 0:
        last = num % 10
        if last < min1:
            min2 = min1
            min1 = last
        elif (last < min2):
            min2 = last
        num = num // 10
    return min1 * 10 + min2
num1 = 23792
res1 = least2(num1)
print(num1)
print(res1)
num2 = 37221
res2 = least2(num2)
print(num2)
print(res2)
```

```
num3 = 10270
res3 = least2(num3)
print(num3)
print(res3)
num4 = 73391
res4 = least2(num4)
print(num4)
print(res4)
print(res1+res2+res3+res4)
```

22) An array of size N denoting the required answer for each student.

```
i/p1 = 6
i/p2 = {4,9,5,3,2,10}
l1 = [4,9,5,3,2,10]
l2 = []
for i in range(len(l1)):
        count=0
        for j in range(0,i):
            if l1[i]>l1[j]:
                count+=1
        l2+=[count]
print(l2)
```

23) WAP to print the numbers from 1 to 25 in clockwise direction

```
1   2   3   4   5
16  17  18  19  6
15  24  25  20  7
14  23  22  21  8
13  12  11  10  9
```

⇒ n = 9

```
l1 = [None] * n
for i in range(len(l1)):
        l2 = []
        for j in range(len(l1)):
                l2 += [0]
        l1[i] = l2
count = 1
start = 0
end = len(l1) - 1
while start < end:
        for i in range(start, end):
                l1[start][i] = count
                count += 1
        for i in range(start, end):
                l1[i][end] = count
                count += 1
        for i in range(end, start, -1):
                l1[end][i] = count
                count += 1
        for i in range(end, start, -1):
                l1[i][start] = count
                count += 1
        start += 1
        end -= 1
if len(l1) % 2 != 0:
        l1[len(l1)//2][len(l1)//2] = count
```