

1 Sum of all number

```
print("sum of all number from n1 to n2 ")
n1 = int(input("enter start number: "))
n2 = int(input("enter end number: "))
result = 0

for i in range(n1, n2+1):
    result = result + i

print("n1 =", n1)
print("n2 =", n2)
print("sum of all number from n1 to n2 is : ", result)
```

2 separted array

```
print("separted array element in +ve and -ve ")
p = []
n = []
num = int(input("enter limited array: "))
a = [int(input("Enter the elements: ")) for i in range(num)]

for i in a:
    if i >= 0:
        p.append(i)
    else:
        n.append(i)

print("all element :", a)
print("positive array element: ", p)
print("negative array element: ", n)
print("sum of positive element: ", sum(p))
print("sum of negative element: ", sum(n))
```

3 liner search

```
def linear_search(a, n, x):
    for i in range(n):
        if a[i] == x:
            return i
    return -1

n = int(input("enter length of list: "))
a = [int(input("Enter the elements: ")) for i in range(n)]
print(a)
x = int(input("Enter element to be searched: "))

r = linear_search(a, n, x)
if r == -1:
    print("Element not found")
else:
    print("Element is found at at position: ", r + 1)
```

4 binary search

```
def bs(a, l, h, x):
    if h >= l:
        m = (h + l) // 2
        if a[m] == x:
            return m
        elif a[m] > x:
            return bs(a, l, m-1, x)
        else:
            return bs(a, m+1, h, x)
    else:
        return -1

n = int(input("Enter length of list: "))
a = [int(input("Enter the elements: ")) for i in range(n)]
print(a)
x = int(input("Enter element to be searched: "))

r = bs(a, 0, n-1, x)
if r == -1:
    print("Element not found")
```

```
else:
    print("Element is found at at position: ", r + 1)
```

5 stack operation

```
print("Stack operations")
a = []
n = int(input("Enter limit of array: "))
i = 1
while i:
    ch = int(input("Select your choice given below\n 1.Insertion\n 2.Deletion\n 3.Display\nEnter Choice: "))
    if ch == 1:
        if len(a) >= n:
            print("Stack is overflow")
        else:
            a.append(int(input("Enter an element: ")))
    elif ch == 2:
        if a:
            print("Deleted element is ", a.pop())
        else:
            print("Stack is empty")

    elif ch == 3:
        print("Stack elements are: ", a)
    else:
        print("Invalid choice")
    i = int(input("Do you want to continue press 1 or press 0: "))
```

6 *evaluate experience*

```
def evaluate_expression(expression):
    operand = []
    operator = []
    precedence = ['+', '-', '*', '/']

    for token in expression:
        if token.isdigit():
            operand.append(int(token))
        elif token in precedence:
            operator.append(token)

    while operator:
        right_operand = operand.pop()
        left_operand = operand.pop()
        opr = operator.pop()

        if opr == '+':
            result = left_operand + right_operand
        elif opr == '-':
            result = left_operand - right_operand
        elif opr == '*':
            result = left_operand * right_operand
        elif opr == '/':
            result = left_operand / right_operand
        operand.append(result)

    print(eval(expression.replace(' ', '')))
    return operand.pop()
```

```
expression = input("Enter an expression: ")
result = evaluate_expression(expression)
```

7 Write a program to multiply two matrices.

```
matrix1 = [[1, 2], [3, 4], [5, 6]]
matrix2 = [[7, 8, 9], [10, 11, 12]]

result = [[sum(a*b for a,b in zip(row, col)) for col in zip(*matrix2)] for row in matrix1]

rows1, cols1 = len(matrix1), len(matrix1[0])
rows2, cols2 = len(matrix2), len(matrix2[0])

if cols1 == rows2:
    for row in result:
        print(row)
else:
    print("The matrices cannot be multiplied.")
```

8 Write a program to find the roots of a quadratic equation.

```
import math
a = float(input("Enter the coefficient of x^2: "))
b = float(input("Enter the coefficient of x: "))
c = float(input("Enter the constant term: "))

discriminant = b**2 - 4*a*c
sqr=math.sqrt(abs(discriminant))
root1 = (-b + sqr) / (2*a)
root2 = (-b - sqr) / (2*a)

if discriminant > 0:
    print("The two real and different roots are:", root1, "and", root2)
elif discriminant == 0:
    print("The repeated root is:", -b / (2*a))
else:
    print("The two complex roots are:", root1, "and", root2)
```

9 Write a program to insert a number in sorted array.

```
print("Sorting an array")
arr = []
n = int(input("Enter limit of an array:"))
print("Enter elements:")
i = 0
while (i < n):
    ele = int(input())
    arr.append(ele)
    i = i + 1
print("Entered elemets are:")
print(arr)
arr.sort()
print("Sorted elemets are:")
print(arr)
ele = int(input("Enter a new element: "))
index = int(input("Enter index value: "))
arr.insert(index, ele)
print(arr)
arr.sort()
print("Sorted elemets are:")
print(arr)
```

10 Write a Python Program to check whether the given string is palindrome or not using built in string manipulation methods.

```
string = input("Enter the string:")
```

```
def palindrome(str):
    if string == string[::-1]:
        print("This is palindrome")
    else:
        print("This is not Palindrome")
```

```
palindrome(str)
```

11 Write a Python program to read a word and print the number of letters, vowels and percentage of vowels in the word using dictionary.

```
d1 = {}
n = int(input("enter a number of values to insert into dictionary:"))
for i in range(n):
    key = input("enter key:")
    d1[key] = input("enter a value:")
print(d1)

def dic(dictx):
    for j in (dictx.values()):
        vowels = 0
        for i in j:
            if (
                i == 'a' or i == 'e' or i == 'i' or i == 'o' or i == 'u' or i == 'A' or i == 'E' or i == 'I' or i == 'O' or i
                == 'U'):
                vowels = vowels + 1
        print("Number of characters in word are:", len(j))
        print("Number of vowels in words are:", vowels)
        vper = (vowels / len(j) * 100)
        print("percentage:", vper)

dic(d1)
```

12 Write a Python Program to check a given sentence is a pangram or not using function/Module.

```
import string
string = input("Enter the string:")
def ispangram(str):
    alphabet = "abcdefghijklmnopqrstuvwxyz"
    for char in alphabet:
        if char not in str.lower():
            return False
    return True

if (ispangram(string) == True):
    print("This is pangram")
```

else:

print("This is not pangram")