```
# 1. Program to find the square root of a number

def countSquares(x): sqrt = x**0.5 result = int(sqrt) return result



x=int(input("Enter the number : "))
print(countsquares(x))

        Enter the number : 144
        12
```

```
# • 2.-Program-to-find*the-smallest-divisor*of-a-number

num-=-int(input("Enter-a-number-: • "))
for-i-in-range(2,-num+1):
 •  •  • -if-num-%-i-==-0:
           print • ("The-smallest-divisor-for-{} • is-{}".format(num,-i))     break

  [}» Enter a number : 522
        The smallest divisor for 522 is 2
```

```
#   3. Program to find GCD of two numbers

def hcf(a, b):
      if(b == 0):
            return a
      else:
            return hcf(b, a % b)

a = int(input("Enter a : "))
b = int(input("Enter b : "))
print("The gcd of a and b is : ", end="")
print(hcf(a, b))

        Enter a : 16
        Enter b : 12
        The gcd of a and b is : 4
```

```
#   4. Program to generate prime numbers

lower = int(input("Enter the lower value:"))
upper = int(input("Enter the upper value:"))
for number in range(lower,upper+l):
      if number>l:
                                    for i in range(2,number):
                                        if (number%i)==0:

                         break
            else:
                  print(number)
```

```
Enter the lower value:10 Enter the upper value:100 11
13
17
19
23
29
31
37
41
43
47
53
59
61
67
71
73
79
83
89
97
```

```
#   5. Program to compute the prime factors of an integer numb =int(input("Enter

the number : "))

print("The prime factors of the given number are : ") value = 1
while(value <= numb):
    k = 0
    if(numb % value == 0):
        1 = 1
        while(j <= value):
            if(value % j == 0):
                k = k+1
            j = j+1
        if(k == 2):
            print(value)
    value = value+1
```

```
Enter the number : 150
The prime factors of the given number are :
2
3
5
```

```
#   6. Program to genereate pseudo random number import random num =
random.random() print(num)
```

```
0.6897279641465557
```

```
#   7. Program for raising a number to the power of a number
```

```
power = 2
numl = 3**power
num2 = 4**power
print('expontial of numl :',numl)
print('expontial of num2 :', num2)

floatnuml = 3**2.5
floatnum2 = 4**3.5

print('\n Exponentiation operator with float exponent') print('expontial of floatnuml
 :',floatnuml ) print('expontial of floatnum2 :', floatnum2 )
```

```
        expontial of numl : 9
        expontial of num2 : 16

          Exponentiation operator with float exponent expontial of floatnuml :
        15.588457268119896 expontial of floatnum2 : 128.0
```

```
#   8. Program to find the nth term of fibonacci series

nterms = int(input("Enter a number: "))

nl = 0
n2 = 1

print("\n The fibonacci sequence is :") print(nl,  n2, end=", ")

for i in range(2, nterms):
    next = nl + n2
    print(next, end=", ")

    nl = n2
    n2 = next
```

```
        Enter a number: 15

          The fibonacci sequence is :
        0 , 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377,
```

```
#implement the main program that would supply and receive data from the push and pop procedure
stack=[]
stack.append("python")
stack.append(5)
stack.append('v')
print("initial stack")
print(stack)
print("element poped from stack")
print(stack.pop())
print(stack.pop())
print(stack.pop())
print("stack after element are popped:")
print(stack)

#y=(a+b)*(c+d)/((x*y)/(w+z) check wether it is balenced or not
open_list=["[","{","("]
close_list=["]","}",")"]
#fn to check parentheses
def check(myStr):
  stack=[]
  for i in myStr:
    if i in open_list:
      stack.append(i)
    elif i in close_list:
      pos=close_list.index(i)
      if((len(stack)>0) and(open_list[pos]==stack[len(stack)-1])):
        stack.pop()
      else:
        return "unbalanced"
  if len(stack)==0:
    return "balanced"
  else:
    return "Unbalanced"

#driver code
string="(a+b)*(c+d)/((x*y)/(w+z)"
print(string,"-",check(string))

# Implement two stacks in a list
class twoStacks:

    def __init__(self, n):     #constructor
        self.size = n
        self.arr = [None] * n
        self.top1 = -1
        self.top2 = self.size

    # Method to push an element x to stack1
    def push1(self, x):
        # There is at least one empty space for new element
        if self.top1 < self.top2 - 1 :
            self.top1 = self.top1 + 1
            self.arr[self.top1] = x

        else:
            print("Stack Overflow ")
            exit(1)

    # Method to push an element x to stack2
    def push2(self, x):

        # There is at least one empty space for new element
        if self.top1 < self.top2 - 1:
            self.top2 = self.top2 - 1
            self.arr[self.top2] = x

        else :
           print("Stack Overflow ")
           exit(1)

    # Method to pop an element from first stack
    def pop1(self):
        if self.top1 >= 0:
            x = self.arr[self.top1]
            self.top1 = self.top1 -1
            return x
        else:
            print("Stack Underflow ")
            exit(1)

    # Method to pop an element from second stack
    def pop2(self):
        if self.top2 < self.size:
            x = self.arr[self.top2]
            self.top2 = self.top2 + 1
            return x
        else:
            print("Stack Underflow ")
            exit()

# Driver program to test twoStacks class
ts = twoStacks(5)
ts.push1(5)
ts.push2(10)
ts.push2(15)
ts.push1(11)
ts.push2(7)
```

```
print("Popped element from stack1 is " + str(ts.pop1()))
ts.push2(40)
print("Popped element from stack2 is " + str(ts.pop2()))

# demonstrate queue implementation
# using list

# Initializing a queue
queue = []

# Adding elements to the queue
queue.append('a')
queue.append('b')
queue.append('c')

print("Initial queue")
print(queue)

# Removing elements from the queue
print("\nElements dequeued from queue")
print(queue.pop(0))
print(queue.pop(0))
print(queue.pop(0))

print("\nQueue after removing elements")
print(queue)

#dequeue implimentation
import collections

DoubleEnded = collections.deque(["Mon","Tue","Wed"])
DoubleEnded.append("Thu")

print ("Appended at right - ")
print (DoubleEnded)

DoubleEnded.appendleft("Sun")
print ("Appended at right at left is - ")
print (DoubleEnded)

DoubleEnded.pop()
print ("Deleting from right - ")
print (DoubleEnded)

DoubleEnded.popleft()
print ("Deleting from left - ")
print (DoubleEnded)

#implimentation of queue using linkedlist
```

```python
class Node:

    def __init__(self, data):
        self.data = data
        self.next = None

# A class to represent a queue

# The queue, front stores the front node
# of LL and rear stores the last node of LL


class Queue:

    def __init__(self):
        self.front = self.rear = None

    def isEmpty(self):
        return self.front == None

    # Method to add an item to the queue
    def EnQueue(self, item):
        temp = Node(item)

        if self.rear == None:
            self.front = self.rear = temp
            return
        self.rear.next = temp
        self.rear = temp

    # Method to remove an item from queue
    def DeQueue(self):

        if self.isEmpty():
            return
        temp = self.front
        self.front = temp.next

        if(self.front == None):
            self.rear = None


# Driver Code
if __name__ == '__main__':
    q = Queue()
    q.EnQueue(10)
    q.EnQueue(20)
    q.DeQueue()
    q.DeQueue()
    q.EnQueue(30)
    q.EnQueue(40)
    q.EnQueue(50)
    q.DeQueue()
    print("Queue Front : " + str(q.front.data))
    print("Queue Rear : " + str(q.rear.data))

#design a list searching algorithm that incorporates a sential
def orderedSequentialSearch(alist,item):
  pos=0
  found=False
  stop=False
  while pos<len(alist) and not found and not stop:
    if alist[pos]==item:
      found=True
    else:
      if alist[pos]>item:
        stop=True
      else:
        pos=pos+1
  return found
testlist=[0,1,2,8,13,17,19,32,42]
print(orderedSequentialSearch(testlist,3))
print(orderedSequentialSearch(testlist,13))
```

```
False
True
```

```
#binary search
def binary_search(alist, start, end, key):
    """Search key in alist[start... end - 1]."""
    if not start < end:
        return -1

    mid = (start + end)//2
    if alist[mid] < key:
        return binary_search(alist, mid + 1, end, key)
    elif alist[mid] > key:
        return binary_search(alist, start, mid, key)
    else:
        return mid


alist = input('Enter the sorted list of numbers: ')
alist = alist.split()
alist = [int(x) for x in alist]
key = int(input('The number to search for: '))

index = binary_search(alist, 0, len(alist), key)
if index < 0:
    print('{} was not found.'.format(key))
else:
    print('{} was found at index {}.'.format(key, index))
```

```
Enter the sorted list of numbers: 5 6  8 9 12 14
The number to search for: 6
6 was found at index 1.
```

```
#Python program to select the ith smallest element from a list in
expected linear time
def select(alist, start, end, i):
    """Find ith smallest element in alist[start... end-1]."""
    if end - start <= 1:
        return alist[start]
    pivot = partition(alist, start, end)

    # number of elements in alist[start... pivot]
    k = pivot - start + 1

    if i < k:
        return select(alist, start, pivot, i)
    elif i > k:
        return select(alist, pivot + 1, end, i - k)

    return alist[pivot]

def partition(alist, start, end):
    pivot = alist[start]
    i = start + 1
    j = end - 1

    while True:
        while (i <= j and alist[i] <= pivot):
            i = i + 1
        while (i <= j and alist[j] >= pivot):
            j = j - 1

        if i <= j:
            alist[i], alist[j] = alist[j], alist[i]
        else:
            alist[start], alist[j] = alist[j], alist[start]
            return j


alist = input('Enter the list of numbers: ')
alist = alist.split()
alist = [int(x) for x in alist]
i = int(input('The ith smallest element will be found. Enter i: '))

ith_smallest_item = select(alist, 0, len(alist), i)
print('Result: {}.'.format(ith_smallest_item))
```

```
#find_max_subarray
def find_max_subarray(alist, start, end):
    """Returns (l, r, m) such that alist[l:r] is the maximum subarray
in
    A[start:end] with sum m. Here A[start:end] means all A[x] for
start <= x <
    end."""
    max_ending_at_i = max_seen_so_far = alist[start]
    max_left_at_i = max_left_so_far = start
    # max_right_at_i is always i + 1
    max_right_so_far = start + 1
    for i in range(start + 1, end):
        if max_ending_at_i > 0:
            max_ending_at_i += alist[i]
        else:
            max_ending_at_i = alist[i]
            max_left_at_i = i
        if max_ending_at_i > max_seen_so_far:
            max_seen_so_far = max_ending_at_i
            max_left_so_far = max_left_at_i
            max_right_so_far = i + 1
    return max_left_so_far, max_right_so_far, max_seen_so_far


alist = input('Enter the list of numbers: ')
alist = alist.split()
alist = [int(x) for x in alist]
start, end, maximum = find_max_subarray(alist, 0, len(alist))
print('The maximum subarray starts at index {}, ends at index {}'
      ' and has sum {}.'.format(start, end - 1, maximum))
```

1.Python Program to calculate the square root

```
num = float(input('Enter a number: '))

num_sqrt = num ** 0.5
print('The square root of %0.3f is %0.3f'%(num ,num_sqrt))
```

1. find the smallest divisor of an integer.

```
n=int(input("Enter an integer:"))
a=[]
for i in range(2,n+1):
    if(n%i==0):
        a.append(i)
a.sort()
print("Smallest divisor is:",a[0])
```

1. Recursive function to return gcd of a and b

```
def gcd(a, b):

      if (a == 0):
            return b
      if (b == 0):
            return a

      if (a == b):
            return a

      if (a > b):
            return gcd(a-b, b)
      return gcd(a, b-a)

a = 98
b = 56
if(gcd(a, b)):
      print('GCD of', a, 'and', b, 'is', gcd(a, b))
else:
      print('not found')
```

4.Generating the Prime Number between the given Interval.

```
lower_value = int(input ("Please, Enter the Lowest Range Value: "))
upper_value = int(input ("Please, Enter the Upper Range Value: "))

print ("The Prime Numbers in the range are: ")
for number in range (lower_value, upper_value + 1):
    if number > 1:
        for i in range (2, number):
            if (number % i) == 0:
                break
        else:
            print (number)
```

1. computing prime factors of a number python

```
n=int(input("Enter an integer:"))
print("Factors are:")
i=1
while(i<=n):
    k=0
    if(n%i==0):
        j=1
        while(j<=i):
            if(i%j==0):
                k=k+1
            j=j+1
        if(k==2):
            print(i)
    i=i+1
```

1. Generation of pseudo prime numbers

```
def countPrimesUpto(n):

      count = 0;

      arr1 = [0 for _ in range(n + 1)];

      arr2 = [1 for _ in range(n + 1)];

      d = 5;

      arr1[2] = 1
      arr2[2] = 1;
      arr1[3] = 1
      arr2[3] = 1;

      while (d <= n) :

            for i in range(d, len(arr1), 6):
                  arr1[i] = 1;

            for i in range(d + 2, len(arr1), 6):
                  arr1[i] = 1;

            d = d + 6;


      for i in range(5, 1 + int(n ** 0.5), 6):
            j = 0;
```

```
        while (1) :
            flag = 0;

            temp1 = 6 * i * (j + 1) + i;

            temp2 = ((6 * i * j) + i * i);

            temp3 = ((6 * (i + 2) * j) + ((i + 2) * (i + 2)));

            temp4 = ((6 * (i + 2) * (j + 1)) + ((i + 2) * (i + 2)) - 2
* (i + 2));

            if (temp1 <= n):
                arr2[temp1] = 0;

            else :
                flag += 1;


            if (temp2 <= n) :
                arr2[temp2] = 0;

            else :
                flag += 1;


            if (temp3 <= n) :
                arr2[temp3] = 0;

            else :
                flag += 1;

            if (temp4 <= n) :
                arr2[temp4] = 0;

            else :
                flag += 1;


            if (flag == 4) :
                break;

            j += 1

    if (n >= 2):
        count +=1

    if (n >= 3):
        count += 1

    for p in range(5, n + 1, 6):
        if (arr2[p] == 1 and arr1[p] == 1):
            count += 1

        if arr2[p + 2] == 1 and arr1[p + 2] == 1:
            count += 1

    return count;

n = 100;
print(countPrimesUpto(n));
```

1. Writing power function for large numbers

```
MAX=100000

def multiply(x, res, res_size):

    carry = 0

    for i in range(res_size):
        prod = res[i] * x + carry

        res[i] = prod % 10

        carry = prod // 10

    while (carry):
        res[res_size] = carry % 10
        carry = carry // 10
        res_size+=1

    return res_size


def power(x,n):

    if (n == 0) :
        print("1")
        return

    res=[0 for i in range(MAX)]
    res_size = 0
    temp = x

    while (temp != 0):
        res[res_size] = temp % 10;
```

```
      res_size+=1
      temp = temp // 10

   for i in range(2, n + 1):
      res_size = multiply(x, res, res_size)

   print(x , "^" , n , " = ",end="")
   for i in range(res_size - 1, -1, -1):
      print(res[i], end="")

exponent = 100
base = 2
power(base, exponent)
```

1. Function for nth Fibonacci number

```
def Fibonacci(n):
   if n<= 0:
      print("Incorrect input")

   elif n == 1:
      return 0

   elif n == 2:
      return 1
   else:
      return Fibonacci(n-1)+Fibonacci(n-2)

print(Fibonacci(10))
```

CHAPTER 2 : FUNDAMENTAL ALGORITHMS

Question No 1 :- Design an algorithm that reads the list of numbers and finds the count of non negative numbers in it .

```
l=[]
print("enter the length of list of numbers :")
n=int(input())
count=0
for i in range(n):
  k=int(input())
  l.append(k)
  if(l[i]<0):
    count+=1
print("the number of negative numbers int list are :",count)
```

```
enter the length of list of numbers :
5
-1
-5
6
```

```
4
-2
the number of negative numbers int list are : 3
```

Question No 2:- Design an algorithm to swap the values of the 3 variables a,b,c.

```
a=int(input("enter the first number :"))
b=int (input("enter the second number :"))
c=int(input("enter the third number :"))
a,b,c=c,a,b
print("the values of a,b,c respectively are :",a,b,c)
```

```
enter the first number :10
enter the second number :20
enter the third number :30
the values of a,b,c respectively are : 30 10 20
```

Question No:- 3 write a program that reads the number of students n and marks of n students and returns the number of students passed .

```
n=int ( input("enter the number :"))
print("enter the marks of ",n," students:")
for i in range(1,n+1):
    k=int(input())
    if(k<35):
        n-=1
print("the number of students passed is :",n)
```

```
enter the number :5
enter the marks of  5  students:
100
50
32
3
42
the number of students passed is : 3
```

Question no:-4 find the average of n numbers.

```
# This is formatted as code
```

```
n=int(input("enter the value of n:"))
sum=0
for i in range(n):
    k=int(input("enter the number :"))
    sum+=k
print("average of the numbers is :" , sum/n)
```

```
enter the value of n:5
enter the number :1
enter the number :2
enter the number :3
enter the number :4
enter the number :5
average of the numbers is : 3.0
```

Question no:-5 Design an algorithm to compute sum of square of n natural numbers.

```
n=int(input("enter the value of n:"))
sum=0
for i in range (1,n+1):
    sum+=i**2
print("Sum of suares of ",n,"natural numbers is :",sum)
```

```
enter the value of n:5
Sum of suares of  5 natural numbers is : 55
```

Question no:- 6 Find the harmonic mean of n values of a data

```
n=int(input("enter a number to find the harmonic mean:"))
sum=0.0
print("enter the numbers :")
for i in range (n):
    k=int(input())
    sum+=(1/k)
print("harmonic mean = ",n/sum)
```

```
enter a number to find the harmonic mean:5
enter the numbers :
1
2
3
4
5
harmonic mean =  2.18978102189781
```

Question no 7:- Devlop an algorithm which finds the n terms of a sequence : 1 -1 1 -1 1 -1 -1

```
n=int(input("enter the number:"))
for i in range(n):
    print(int((-1)**(i-1)),end=' ')
```

```
enter the number:10
-1 1 -1 1 -1 1 -1 1 -1 1
```

Question no 8:- Devlop an algorithm which finds the n terms of a sequence : 1 , -3 ,5 ,-7 , 9 ....

```
n=int(input("enter the number:"))
print("the series is :")
sum=0
for i in range(n):
    k=int(((-1)**(i))*(2*i+1))
    print(k,end=' ')
    sum+=k
print("\nsum of the sequence is :",sum)
```

```
enter the number:10
the series is :
1 -3 5 -7 9 -11 13 -15 17 -19
sum of the sequence is : -10
```

Question 9:- Find : s=1+2+3+4.....

```
n=int(input("enter the number:"))
sum=0
for i in range(1,n+1):
    sum+=i
print("sum of the sequence is :",sum)
```

```
enter the number:5
sum of the sequence is : 15
```

Question No:-10 Find : s=2+4+8+12......

```
n=int(input("enter the number:"))
sum=0
for i in range(2,2*n+1,2):
    sum+=i
print("sum of the sequence is :",sum)
```

```
enter the number:5
sum of the sequence is : 30
```

Question No:-11 Find : s=1+3+5+7......

```
n=int(input("enter the number:"))
sum=0
for i in range(1,2*n,2):
    sum+=i
print("sum of the sequence is :",sum)
```

```
enter the number:5
sum of the sequence is : 25
```

Question 12:- Find factorial of a number.

```
def fact(n):
    if(n==0):
        return 1
    return n*fact(n-1)
n=int(input("enter the number:"))
f=fact(n)
print("factorial of ",n,"is :",f)
```

```
enter the number:5
factorial of  5 is : 120
```

Question 13:- for a given n find 1/n!

```
def fact(n):
    if(n==0):
        return 1
    return n*fact(n-1)
n=int(input("enter the number:"))
f=fact(n)
print("1/n! of ",n,"is :",1/f)
```

```
enter the number:5
1/n! of  5 is : 0.008333333333333333
```

Question 13:- for a given x and n find x^n/n!

```
def fact(n):
    if(n==0):
        return 1
    return n*fact(n-1)
n=int(input("enter the value of n:"))
x=int(input("enter the value of x:"))
f=fact(n)
print("x^n / n! =",(x**n)/f)
```

```
enter the value of n:5
enter the value of x:2
x^n / n! = 0.26666666666666666
```

Question 14:- find weather the entered number is a factorial or not

```
n=int(input("enter the value of n:"))
f=True
i=1
while(f):
    if(n%i!=0):
        f=False
    else:
        n=n/i
        i+=1
if(n==1):
    print("yes it is a factorial number!")
else:
    print("no it is not a factorial number!")
```

```
enter the value of n:120
yes it is a factorial number!
```

Question 15:- find the value of : 0! + 1! + 2! + 3! + 4! .....

```
def fact(n):
    if(n==0):
        return 1
    return n*fact(n-1)
n=int(input("enter the value of n:"))
sum=0
for i in range (1,n+1):
    sum+=fact(i)
print("the value of sum of series is :",sum)
```

```
enter the value of n:5
the value of sum of series is : 153
```

Question 16:- find the value of : 1/0! + 1/1! + 1/2! + 1/3! + 1/4! .....

```
def fact(n):
    if(n==0):
        return 1
    return n*fact(n-1)
n=int(input("enter the value of n:"))
sum=0
for i in range (1,n+1):
    sum+=1/fact(i)
print("the value of sum of series is :",sum)
```

```
enter the value of n:5
the value of sum of series is : 1.7166666666666668
```

Question 17:- Implement an algorithm which reads two concicative fibonacii numbers and outputs the next number in the series.

```
a=int(input("enter the first number :"))
b=int(input("enter the second number :"))
print("the third number is  :",a+b)
```

```
enter the first number :10
enter the second number :20
the third number is  : 30
```

Question 18:- Design an algorithm to print luca's sequence: 1 3 4 7 ....

```
n=int(input("enter the number :"))
a=1
b=3
if (n==1):
    print(a)
else:
    print(a,b,end=' ')
    a=a+b
    for i in range(2,n):
        print(a,end=' ')
        a=a+b
        b=a
```

```
enter the number :5
1 3 4 7 14
```

Question 19:- Print fibonacci sequence

```
n=int(input("enter the first number :"))
a=1
b=1
if (n==1):
    print(a)
else:
    print(a,b,end=' ')
    a=a+b
    for i in range(2,n):
        print(a,end=' ')
        a=a+b
        b=a
```

```
enter the first number :5
1 1 2 3 6
```

Question 20 :- print the sequence in which each term is the sum of preceding 3 terms the first thee terms are given as 0 , 1 , 1

```
n=int(input("enter the first number :"))
a=0
b=1
c=1
if(n==1):
    print(0)
elif(n==2):
    print(0,1)
elif(n==3):
    print(0,1,1)
else:
    print(0,1,1,end=' ')
    for i in range(3,n):
        t=c
        c=a+b+c
        print(c,end=' ')
        a=b
        b=t
```

```
enter the first number :10
0 1 1 2 4 7 13 24 44 81
```

Question : 21 write a program to stimulate multiplication using addition.

```
a=int(input("enter the first number:"))
b=int(input("enter the second number:"))
ans=0
```

```
for i in range(b):
    ans+=a
print(a,"x",b," = ",ans)
```

```
enter the first number:5
enter the second number:2
5 x 2  =  10
```

Question 22:- design an algorithm which converts a binary number to a octal number.

```
s=input("enter the binart number :")
l=len(s)
numd=0
num=""
k=0
for i in range(l-1,-1,-1):
    if(s[i]=='1'):
        numd+=2**k
    k+=1
while(numd>8):
    num=num+str(numd%8)
    numd=numd//8
num=num+str(numd)
num=num[::-1]
print("octal equivalent = ",num)
```

```
enter the binart number :1100110
octal equivalent =  146
```

Question 23:- find GCD of two numbers.

```
def Gcd(a,b):
    sum=a+b
    g=1
    if(a==1 or b==1):
        return 1
    if(a==0 or b==0):
        return sum
    if(a>b):
        g=Gcd(a%b,b)
    else:
        g=Gcd(a,b%a)
    return g
a=int(input("enter the first number:"))
b=int(input("enter the second number:"))
print("GCD of",a,"and ",b , "is",Gcd(a,b))
```

```
enter the first number:12
enter the second number:15
GCD of 12 and  15 is 3
```

Question 24:- write a program to find all the prime numbers from 1 to n

```
n=int(input("enter a number :"))
l=[]
for i in range(1,n+2):
    l.append(0)
for i in range(2,n+1):
    if(l[i]==0):
        b=2
        while(i*b<=n):
            l[i*b]=1
            b+=1
for i in range(2,n+1):
    if(l[i]==0):
        print(i,end=" ")
```

```
enter a number :20
2 3 5 7 11 13 17 19
```

Question 25:- Merge two array into a sorted array

```
a=int(input("entert the length of the first array :"))
b=int(input("entert the length of the second array :"))
l1=[]
l2=[]
l3=[]
print("enter the first list :")
for i in range(a):
    k=int((input()))
    l1.append(k)
print("enter the second list :")
for i in range(b):
    k=int((input()))
    l2.append(k)
pos=0
i=0
j=0
while(pos<a+b):
    if(i<a and j<b):
        if(l1[i]<l2[j]):
            l3.append(l1[i])
            pos+=1
            i+=1
        else:
            l3.append(l2[j])
            pos+=1
            j+=1
    elif(i<a):
            l3.append(l1[i])
            pos+=1
            i+=1
    elif(j<b):
            l3.append(l2[j])
            pos+=1
            j+=1
print("lists after merging is :",l3)
```

```
entert the length of the first array :3
entert the length of the second array :5
enter the first list :
1
3
5
enter the second list :
2
4
6
8
10
lists after merging is : [1, 2, 3, 4, 5, 6, 8, 10]
```

Question 26 :- Implement Binary Search.

```
n=int(input("enter the length of the array :"))
t=int(input("enter the target to be searched :"))
lst=[]
print("enter the array elemets :")
for i in range(n):
    lst.append(int(input()))
l=0
h=n-1
index=-1
while(l<=h):
    mid=(l+h)//2
    if(lst[mid]==t):
        index=mid
        break
    elif(lst[mid]<t):
        l=mid+1
    else:
        h=mid-1
if(index==-1):
    print("elemnt not found!")
else:
    print("element found at index :",index)
```

```
enter the length of the array :8
enter the target to be searched :1
enter the array elemets :
1
2
3
```

```
4
5
6
7
8
element found at index : 0
```

Question 27:- Find the largest element in the array.

```python
n=int(input("enter the length of the array :"))
lst=[]
print("enter the array elemets :")
for i in range(n):
    lst.append(int(input()))
m=lst[0]
for i in range(n):
    if(lst[i]>m):
        m=lst[i]
print("maximum element in the array is :",m)
```

```
enter the length of the array :8
enter the array elemets :
10
23
4
56
90
21
5
9
maximum element in the array is : 90
```

Question 27:- Find the smallest element in the array.

```python
n=int(input("enter the length of the array :"))
lst=[]
print("enter the array elemets :")
for i in range(n):
    lst.append(int(input()))
m=lst[0]
for i in range(n):
    if(lst[i]<m):
        m=lst[i]
print("minimum element in the array is :",m)
```

1. Python code for bubble sort

```
def bubbleSort(arr):
    n = len(arr)
    swapped = False
    # Traverse through all array elements
    for i in range(n-1):
        # range(n) also work but outer loop will
        # repeat one time more than needed.
        # Last i elements are already in place
        for j in range(0, n-i-1):

            # traverse the array from 0 to n-i-1
            # Swap if the element found is greater
            # than the next element
            if arr[j] > arr[j + 1]:
                swapped = True
                arr[j], arr[j + 1] = arr[j + 1], arr[j]

        if not swapped:
            # if we haven't needed to make a single swap, we
            # can just exit the main loop.
            return


# Driver code to test above
arr = [64, 34, 25, 12, 22, 11, 90]
```

```
bubbleSort(arr)


print("Sorted array is:")
for i in range(len(arr)):
    print("% d" % arr[i], end=" ")
```

2. Python program for implementation of Insertion Sort

```
# Function to do insertion sort
def insertionSort(arr):

    # Traverse through 1 to len(arr)
    for i in range(1, len(arr)):

        key = arr[i]

        # Move elements of arr[0..i-1], that are
        # greater than key, to one position ahead
        # of their current position
        j = i-1
        while j >=0 and key < arr[j] :
                arr[j+1] = arr[j]
                j -= 1
        arr[j+1] = key
```

3 . sorting the array [12, 11, 13, 5, 6] using insertionSort

```
arr = [12, 11, 13, 5, 6]

insertionSort(arr)

print(arr)
```

```
def unsortedarray (a, b, res, n, m):


  a.sort()

  b.sort()

  # Merge two sorted arrays into res[]

  i, j, k = 0, 0, 0

  while (i < n and j < m):

    if (a[i] <= b[j]):

      res[k] = a[i]

      i += 1

      k += 1

    else:

      res[k] = b[j]

      j += 1

      k += 1

  while (i < n):  # Merging remaining

    # elements of a[] (if any)

    res[k] = a[i]

    i += 1
```

```
        k += 1
    while (j < m):  # Merging remaining
        # elements of b[] (if any)
        res[k] = b[j]
        j += 1
        k += 1
# Driver code
A=list()
n=int(input("Enter the size of the First List ::"))
print("Enter the Element of First List ::")
for i in range(int(n)):
    k=int(input(""))
    A.append(k)
B=list()
m=int(input("Enter the size of the Second List ::"))
print("Enter the Element of Second List ::")
for i in range(int(n)):
    k=int(input(""))
    B.append(k)
# Final merge list
res = [0 for i in range(n + m)]
unsortedarray(A, B, res, n, m)
print ("Sorted merged list :")
for i in range(n + m):
    print (res[i],)
```

```
def partition(array, low, high):

    # Choose the rightmost element as pivot
    pivot = array[high]

    # Pointer for greater element
    i = low - 1

    # Traverse through all elements
    # compare each element with pivot
    for j in range(low, high):
        if array[j] <= pivot:
            # If element smaller than pivot is found
            # swap it with the greater element pointed by i
            i = i + 1

            # Swapping element at i with element at j
            (array[i], array[j]) = (array[j], array[i])

    # Swap the pivot element with
    # e greater element specified by i
    (array[i + 1], array[high]) = (array[high], array[i + 1])

    # Return the position from where partition is done
    return i + 1
```

4. Function to perform quicksort

```
def quick_sort(array, low, high):
    if low < high:

        # Find pivot element such that
        # element smaller than pivot are on the left
        # element greater than pivot are on the right
        pi = partition(array, low, high)

        # Recursive call on the left of pivot
        quick_sort(array, low, pi - 1)

        # Recursive call on the right of pivot
        quick_sort(array, pi + 1, high)


# Driver code
array = [10, 7, 8, 9, 1, 5]
quick_sort(array, 0, len(array) - 1)

print(f'Sorted array: {array}')
```

5. Selection sort using python

```
def selectionSort(array, size):

    for step in range(size):
        min_idx = step

        for i in range(step + 1, size):

            # to sort in descending order, change > to < in this line
            # select the minimum element in each loop
            if array[i] < array[min_idx]:
                min_idx = i

        # put min at the correct position
        (array[step], array[min_idx]) = (array[min_idx], array[step])



data = [-2, 45, 0, 11, -9]
size = len(data)
selectionSort(data, size)
print('Sorted Array in Ascending Order:')
print(data)
```

6. Shell sort using python

```
def shell_sort(inp, n):


    h = n // 2
    while h > 0:
        for i in range(h, n):
            t = inp[i]
            j = i
            while j >= h and inp[j - h] > t:
                inp[j] = inp[j - h]
                j -= h


            inp[j] = t
        h = h // 2




inp = [34, 12, 20, 7, 13, 15, 2, 23]
n = len(inp)
print('Array before Sorting:')
print(inp)
shell_sort(inp, n)
print('Array after Sorting:')
print(inp)
```