WEEK - 7

Lists

1) An array is monotonic if it is either monotone increasing or monotone decreasing. An array A is monotone increasing if for all i <= j, A[i] <= A[j]. An array A is monotone decreasing if for all i <= j, A[i] >= A[j].

Write a program if n array is monotonic or not. Print "True" if is monotonic or "False" if it is not. Array can be monotone increasing or decreasing.

Input	Result
4 6 5 4 3	True

PROGRAM :

```
n=int(input())
a=[]
for i in range(n):
        a.append(int(input()))
mo=True
if a[0]<a[1]:
        for i in range(1,n-1):
                if a[i]>=a[i+1]:
                        mo=False
                        break
elif a[0]>a[1]:
        for i in range(1,n-1):
                if a[i]<=a[i+1]:
                        mo=False
                        break
if mo:
        print(True)
else:
        print(False)
```

2) Given two arrays of positive integers, for each element in the second array, find the total number of elements in the first array which are *less than or equal to* that element. Store the values determined in an array.

For example, if the first array is *[1, 2, 3]* and the second array is *[2, 4]*, then there are *2* elements in the first array *less than or equal to 2*. There are *3* elements in the first array which are *less than or equal to 4*. We can store these answers in an array, *answer = [2, 3]*.

Input	Result
4 1 4 2 4 2 3 5	2 4

PROGRAM :

```
n=int(input())
 a=[]
for i in range(n):
         a.append(int(input()))
m=int(input())
b=[]
for i in range(m):
        b.append(int(input()))
c=[]
for i in range(m):
        co=0
        for j in range(n):
                if a[j]<=b[i]:
                        co+=1
        c.append(co)
for i in c:
        print(i)
```

3) Program to print all the distinct elements in an array.

Input Format:

First line take an Integer input from stdin which is array length n.

Second line take n Integers which is inputs of array.

Output Format:

Print the Distinct Elements in Array in single line which is space Separated

Input	Result
5 1 2 2 3 4	1 2 3 4
6 1 1 2 2 3 3	1 2 3

PROGRAM :

```
n=int(input())
 a=[]
for i in range(n):
        a.append(int(input()))
c=[]
for i in range(n):
        if a[i] not in c:
                c.append(a[i])
print(*c)
```

4) Given an integer n, return an list of length n + 1 such that for each i (0 <= i <= n), ans[i] is the number of 1's in the binary representation of i.

Test	Result
print(CountingBits(5))	[0, 1, 1, 2, 1, 2]

5) Determine the factors of a number (i.e., all positive integer values that evenly divide into a number) and then return the pth element of the list, sorted ascending. If there is no pth element, return 0.

Eg,: n = 20 p = 3

The factors of 20 in ascending order are {1, 2, 4, 5, 10, 20}. Using 1-based indexing, if p = 3, then 4 is returned. If p > 6, 0 would be returned.

Input	Result
10 3	5
10 5	0
1 1	1

PROGRAM :

```
def CountingBits(n):
        b=[]
        for i in range(n+1):
                c=0
                while i:
                        if i &1:
```

$$c+=1$$

$$i=i>>1$$

b.append(c)

return b

6) The program must accept N integers and an integer K as the input. The program must print every K integers in descending order as the output.

Note: If N % K != 0, then sort the final N%K integers in descending order.

Input Format:

The first line contains the values of N and K separated by a space.

The second line contains N integers separated by space(s).

Output Format:

The first line contains N integers.

Input	Result
7 3 48 541 23 68 13 41 6	541 48 23 68 41 13 6

PROGRAM :

```
n,k=map(int,input().split())
a=list(map(int,input().split()))
0=[]
for i in range(0,n,k):
        o.extend(sorted(a[i:i+k],reverse=True))
print(*o)
```

7) Given an array A of sorted integers and another non negative integer k, find if there exists 2 indices i and j such that A[i] - A[j] = k, i != j.

Input	Result

1 3 1 3 5 4	1
1 3 1 3 5 99	0

PROGRAM :

```
t=int(input())
while(t!=0):
        n=int(input())
        a=[]
        for i in range(n):
                a.append(int(input()))
        k=int(input())
        flag=0
        for i in range(n):
                for j in range(n):
                        if a[i]-a[j]==k and i!=k:
                                flag=1
        if flag:
                print("1")
        else:
                print('0')
        t--1
```

8) Given a matrix mat where every row is sorted in strictly increasing order, return the smallest common element in all rows.

If there is no common element, return -1.

Input	Result
4 5 1 2 3 4 5 2 4 5 8 10 3 5 7 9 11 1 3 5 7 9	5

PROGRAM :

```
x=input().split()
n,m=int(x[0]), int(x[1])
a=[]
for i in range(n):
        x=input().split()
        b=[]
        for j in x:
                b.append(int(j))
                a.append(b)
1=set (a[0])
for i in range(1,n):
        1=1&set(a[i]) print(min(list(1)))
```

9) Assume you have an array of length *n* initialized with all 0's and are given *k* update operations.

Each operation is represented as a triplet: [startIndex, endIndex, inc] which increments each element of subarray A[startIndex ... endIndex] (startIndex and endIndex inclusive) with inc.

Return the modified array after all *k* operations were executed.

Input	Result
5	-2 0 3 5 3
3	
1 3 2	
2 4 3	
0 2 -2	

PROGRAM   :

```
n=int(input())
k=int(input())
a=[0]*n
for i in range(k):
        s,e,inc-map(int,input().split())
        a[s:e+1]=[x+inc for x in a[s:e+1]]
print(*a)
```

10) Complete the program to count the frequency of each element of an array. Frequency of a particular element will be printed once.

Input	Result
7	23 occurs 3 times
23	45 occurs 2 times
45	56 occurs 1 times
23	40 occurs 1 times
56	
45	
23	
40	

PROGRAM   :

```
n=int(input())
```

```python
a=[]
for i in range(n):
        a.append(int(input()))
        freq={}
        for i in a:
                if i in freq:
                        freq[i] +=1
                        freq[i]=1
for i in freq:
        print(i, 'occurs', freq[i], 'times')
```