⭕ INFOSYS SP(9.5LPA) & DSP(6.25LPA) Exam ⭕

📌 Selection Process:

1) **Online Test (3 Coding Question - 3 Hours) (Easy, Medium & Hard).**
2) **Technical Interview/Behavioral Interview.**

📌 Online Test:

Each Question will have a different Difficulty Level:

a) Easy (all the test cases should run).

b) Medium (80% of the test cases should run).

c) Hard (75% of the test cases should run).

📌 Programming languages allowed For All 3 Questions:

a) C

b) C++

c) Java

d) Python

e) JavaScript

☑ **Note: You can use different programming languages to solve different coding problems.**

📌 Easy: That can be solved by applying the basic application of aptitude, Algorithm, and Data Structures:

Easy level Question based on the following Topics:

a) Aptitude

b) Algorithm

c) Stack

d) Queue

e) Tree

f) Graph

g) Hash Map

h) Linked List

📌 Medium: That can be solved by applying the basics of the Greed algorithm.

a) Pure Greedy Algorithms.

b) Orthogonal Greedy Algorithms.

c) Relaxed Greedy Algorithms.

📌 Hard: Usually a question based on Dynamic Programming.

Some common examples of Dynamic Problems are :

a) Overlapping Subproblem.

b) Optimal substructure properties.

c) Knapsack problem

d) Fibonacci Numbers

e) Palindromic Subsequence

f) Longest Common Substring

g) Dijkstra's algorithm

⭕ Previous Question Papers ⭕

1) Khaled has an array A of N elements. It is guaranteed that N is even. He wants to choose at most N/2 elements from array A. It is not necessary to choose consecutive elements. Khaled is interested in XOR of all the elements he chooses. Here, XOR denotes the bitwise XOR operation.

For example:

If A=[2,4,6,8], then Khaled can choose the subset [2,4,8] to achieve XOR=(2 XOR 4 XOR 8)=14.

Khaled wants to maximize the XOR of all the elements he chooses. Your task is to help Khaled to find the max XOR of a subset that he can achieve by choosing at most N/2 elements?

Input format:

The first line contains an integer, N, denoting the number of elements in A.

Each line i of the N subsequent lines (where 0<=i<=N) contains an integer describing Ai.

Constraints

1<=N<=120

1<=A[i]<=10^6

Sample Input 1

2

1

2

Sample Output 1

2

Explanation:

N=2, A=[1,2] Khaled can choose the subset[2]. The xor of the elements in the subset is 2. And the number of elements in the subset is 1 which is less than N/2.

Sample Input 2

4

1

2

4

7

Sample Output 2

7

Explanation:

N=4, A=[1,2,4,7] Khaled can choose the subset [7]. The xor of the elements in the subset is 7, and the number of elements in the subset is 1 which is less than N/2.

main.py

```python
from itertools import combinations


def fun(arr, N):
    sub = []
    max_xor = max(arr)
    for i in range(1, N // 2):
        comb = combinations(arr, i + 1)
        for i in comb:
            sub.append(list(i))
    for a in sub:
        z = 0
        for b in a:
            z = z ^ b
        if z > max_xor:
            max_xor = z
    return max_xor


N = int(input("Enter Length : "))
arr = []
for i in range(N):
    arr.append(int(input(f"Enter {i+1} Element : ")))
if N < 3:
    print("Output :", max(arr))
else:
    print("Output :", fun(arr, N))
```

2) You have been given a string S of length N. The given string is a binary string which consists of only 0's and '1's. Ugliness of a string is defined as the decimal number that this binary string represents.

Example:

"101" represents 5.

"0000" represents 0.

"01010" represents 10.

There are two types of operations that can be performed on the given string.

Swap any two characters by paying a cost of A coins.

Flip any character by paying a cost of B coins

flipping a character means converting a '1'to a '0'or converting a '0' to a '1'.

Initially, you have been given coins equal to the value defined in CASH. Your task is to minimize the ugliness of the string by performing the above mentioned operations on it. Since the answer can be very large, return the answer modulo 10^9+7.

Note:

You can perform an operation only if you have enough number of coins to perform it.

After every operation the number of coins get deducted by the cost for that operation.

Input Format

The first line contains an integer, N, denoting the number of character in the string

The next line contains a string, S, denoting the the binary string

The next line contains an integer, CASH, denoting the total number of coins present initially

Next will contains an integer, A, denoting the cost to swap two characters.

Then the next line contains an integer, B, denoting the cost to flip a character.

Constraints

1 <= N <= 10^5

1< len(S)<= 10^5

1<=CASH <=10^5

1<=A<=10^5

1<=B<=10^5

Sample Input 1 :

4

1111

7

1

2

Sample Output 1 :

1

Explanation:

3 flips can be used to create “0001” which represents 1.

Sample Input 2:

6

111011

7

1

3

Sample Output 2:

7

Explanation:

First swap 0 with the most significant 1, then use flip twice first on index one and then on index two “111011”=>”0111111”=>”001111”=>”000111” the value represented is 7.

Sample Input 3:

6

111011

7

3

2

Sample Output 3:

3

Explanation:

Flip the 3 most significant characters to get “000011” : the value represented by this string is 3.N

hain.py

```python
N = 6
S = list("111011")
Cash = 7
A = 1
B = 3


def swap():
    global Cash
    Rs = S.copy()
    S[S.index('1')], S[''.join(S).rindex('0')] = S[''.join(S).rindex('0')], S[S.index('1')]
    if Rs == S:
        flip()
    else:
        Cash -= A


def flip():
    global Cash
    S[S.index('1')] = '0'
    Cash -= B


while Cash > A or Cash > B:
    if A < B and '0' in S:
        swap()
    else:
        flip()
print(int(''.join(S), 2))
# N = int(input())
# S = list(input())
# Cash = int(input())
# A = int(input())
# B = int(input())
```

3) You have an interesting string S of length N. It is interesting because you can rearrange the characters of this string in any order. You want to cut this string into some contiguous pieces such that after cutting, all the pieces are equal to one another.

You can't rearrange the characters in the cut pieces or join the pieces together. You want to make the number of pieces as large as possible. What is the maximum number of pieces you can get?

Note: You can observe that you may not want to cut the string at all, therefore the number of pieces is 1. Hence, the answer always exists.

Input Format

S :: STRING

The first line contains a string, S, denoting the string.

length(S) :: 1 -> 2 * 10^5

Sample Input 1:

zzzzz

Sample Output 1 :

5

Sample input 2:

ababcc

Sample Output 2:

2

Sample input 2:

abccdcabacda

Sample Output 2:

2

main.cpp

```cpp
#include<iostream>
#include<unordered_map>
#include<string>
#include<math.h>
using namespace std;

int ans(string s){
    unordered_map map;
    for(int i =0; i map[s[i]]){
            val = map[s[i]];
            key = map[s[i]];
            //we will get the smallest number.
        }
    }
    return val;
}

int main()
{
    string s;
    cin>>s;
    cout<< ans(s);


    return 0;
}
```

4) Today you decided to go to the gym. You currently have energy equal to E units. There are N exercises in the gym. Each of these exercises drains Ai amount of energy from your body.

You feel tired if your energy reaches 0 or below. Calculate the minimum number of exercises you have to perform such that you become tired. Every unique exercise can only be performed at most 2 times as others also have to use the machines.

If performing all the exercises does not make you feel tired, return -1.

Input Format

E :: INTEGER

The first line contains an integer, E, denoting the Energy.

E :: 1 -> 10^5

N :: INTEGER

The next line contains an integer, N, denoting the number of exercises. N :: 1 -> 10^5

A :: INTEGER ARRAY

Each line i of the N subsequent lines (where 0 ≤ i < N) contains an integer describing the amount of energy drained by i-th exercise.

A[i] :: 1 -> 10^5

Sample Input 1:

6

2

1

2

Sample Output 1 :

4

Sample input 2:

10

2

1

2

Sample Output 2:

-1

Sample input 3:

2

3

1

5

2

Sample Output 3:

1

main.cpp

```cpp
#include <iostream>

using namespace std;

int exercise(int arr[], int test, int energy){

    //check if some input == energy , if yess then return 1.
    for(int i =0; i<test; i++){
        if(arr[i] == energy){
            return 1;
        }
    }
    int sum =0;
    int ans =0;
    int ans1=0;
    for(int i=0; i<test; i++){ sum = sum + arr[i]; if(sum >=energy){
            ans = i+1;
            return ans;
        }
    }
    sum =0;
    for(int i =0; i<test; i++){ sum = sum + arr[i]; if(2*sum >= energy){
            ans1 = 2*(i+1);
            return ans1;
        }
    }
    return -1;



}
int main()
{
 int energy ;
 cin>>energy;
 int test;
 cin>>test;

  int arr[test];
  for(int i =0; i<test; i++){ cin>> arr[i];
  }

 int ans = exercise(arr, test, energy);
 cout << ans;

}

```

5) There is a battle between heroes and villains going on. You have M heroes, all of them have the same health H. There are N villains, health of the i-th villain is Vi.

When a hero, with health H battles a villain with health Vi, one of the three scenarios can happen:

if H > Vi: The villain is defeated, and the health of the hero is decreased by Vi if H < Vi: The villain wins, his health is not affected, and the hero is no longer able to fight. if H = Vi: Both are considered defeated, and neither can fight.

The heroes start fighting villains one by one in the same order, first villain 1 then villain 2 and so on. It might be possible that before defeating all the villains, all the heroes are defeated. Therefore, to ensure the victory of the heroes, you want to remove some villains from the front.

Your task is to find the minimum number of villains you need to remove from the front such that the victory of the heroes is guaranteed.

Note: If in the last battle, both the hero and villain are defeated and no more heroes or villains remain, it would still be considered a victory since all the villains are defeated.

Input Format

N :: INTEGER

The first line contains an integer, N, denoting the number of villains N :: 1 -> 2*10^5

M :: INTEGER

The next line contains an integer, M, denoting the number of heroes M :: 1 -> 2*10^5

H :: INTEGER

The next line contains an integer, H, denoting the health of each of the heroes H :: 1 -> 10^9

array :: INTEGER ARRAY

Each line i of the N subsequent lines (where 0 ≤ i < N) contains an integer describing the health of each of the villains.

array[i] :: 1 -> 10^9

Sample Input 1:

4

4

3

3

1

3

3

Sample Output 1 :

0

Sample input 2:

5

3

3

1

2

3

1

1

Sample Output 2:

0

Sample input 3:

5

1

4

1

2

3

1

3

Sample Output 3:

3

main.cpp

```cpp
#include <iostream>

using namespace std;
int ans(int arr[], int n, int m, int h){
    int total = 0;
    for(int i =0; i0; i--){
            sumh= sumh+arr[i];
            if(sumh >= totalhero){
                return i;
            }
        }


    }

}

int main()
{
    int n;//number of villians.
    cin>>n;
    int m ; //number of heroes
    cin>>m;
    int h;//health of hero.
    cin>>h;
    int arr[n];
    for(int i =0; i>arr[i];//health of villians
    }
    cout<
```

6) You are given an array of size N. You need to change this array into a mountain. By mountain we mean, the either ends of the array should have equal elements. Then as we move towards the middle from both ends, the next element is just one more than the previous one. So, it would have a peak in the middle and decrease if you go towards either end, just like a mountain.

Examples of mountains are [1, 2, 3, 2, 1] or [6, 7, 8, 8, 7, 6]. But the array [1, 2, 4, 2, 1] is not a mountain because from 2 to 4 the difference is 2. The array [1, 2, 3, 1] is also not a mountain because the elements 2 and 3 are not equal from both ends.

You need to find the minimum number of elements that should be changed to make the array a mountain. You can make the elements negative or zero as well.

Input Format

N :: INTEGER

The first line contains an integer, N, denoting the number of elements in array. N :: 1 -> 10^5

array :: INTEGER ARRAY

Each line i of the N subsequent lines (where 0 ≤ i < N) contains an integer describing i-th element of array. array[i] :: 1 -> 10^6

Sample Input 1:

5

1

2

3

4

5

Sample Output 1 :

2

Sample input 2:

9

1

1

1

2

3

2

1

1

1

Sample Output 2:

4

main.cpp

```cpp
#include <iostream>

using namespace std;
int mountain(int arr[], int n){
    int count=0;
    if(n%2==0){
        int m2 = n/2;
        int m1 = m2-1;

        if(arr[m1] == arr[m2]){
            for(int i =m1-1; i>=0; i--){
                if(arr[i] != (arr[i+1] -1)){
                    count++;
                    arr[i] = (arr[i+1]-1);
                }
            }
        }
        else if(arr[m1] != arr[m2]){
            arr[m2] = arr[m1];
              for(int i =m1-1; i >= 0; i--){
                if(arr[i] != (arr[i+1] -1)){
                    count++;
                    arr[i] = (arr[i+1]-1);
                }
            }

        }
        int j = m1-1;
        for(int i =m2+1; i < n; i++){
            if(arr[i] != arr[j]){
                count ++;
            }
            j--;
        }
        return count;
    }
    else{
        int mid = n/2;
        for(int i = mid-1; i >= 0; i--){
            if(arr[i] =! (arr[i+1]-1)){
                    count++;
                    arr[i] = (arr[i+1]-1);
                }
        }
        int j = mid-1;
        for(int i = mid+1;i> n;
    int arr[n];
    for(int i =0; i> arr[i];
    }
    cout << mountain(arr,n);
}

```

7) You need to build a road in a rugged terrain. You know the sea level of each segment of the rugged terrain, i.e., the i-th segment is Li meters from sea level.

You need to transform the terrain into a strictly downward sloping terrain for the road, i.e., for each i-th segment where 2 <= i <= N, resultant Li-1 > Li. To do so, you employ a powerful digging team to help you dig and reduce the sea level of the segments. On day D, the team can reduce the sea level for each segment that you scheduled that day by 2D-1 meters each.

You are allowed to assign the team to dig on multiple segments and/or dig on the same segments for multiple days.

Your task is to find the minimum number of days needed to transform the terrain as per your requirements.

Input Format

N :: INTEGER

The first line contains an integer, N, denoting the number of elements in L. N :: 1 -> 10^5

L :: INTEGER ARRAY

Each line i of the N subsequent lines (where 0 < i ≤ N) contains an integer describing Li, the sea level of the i-th segment. L[i] :: -10^9 -> 10^9

Sample Input 1:

2

3

3

Sample Output 1 :

1

Sample input 2:

2

5

-3

Sample Output 2:

0

main.cpp

```cpp
#include <bits/stdc++.h>
#define int long long int
using namespace std;
int32_t main()
{
    int n;
    cin>>n;
    int a[n];
    for(int i =0; i < n; i++) { cin >> a[i];
    }
        int max_dig = 0;
        for(int i =0; i < n-1; i++)
        {
            if(a[i] <= a[i+1])
            {
                max_dig = max(max_dig,(a[i+1]-a[i]+1));
                a[i+1] = a[i] -1;
            }
        }
        int ans = ceil(sqrt(max_dig));
        cout<<  ans << endl;

}

```

8) Andy wants to go on a vacation to de-stress himself. Therefore he decides to take a trip to an island. It is given that he has as many consecutive days as possible to rest, but he can only make one trip to the island. Suppose that the days are numbered from 1 to N. Andy has M obligations in his schedule, which he has already undertaken and which correspond to some specific days. This means that ith obligation is scheduled for day Di. Andy is willing to cancel at most k of his obligations in order to take more holidays.

Your task is to find out the maximum days of vacation Andy can take by canceling at most K of his obligations.

Input Format

The first line contains an integer N, denoting the total number of days

The next line contains an integer M denoting the total number of obligations.

The next line contains an integer K denoting the largest number of obligations he could cancel

Each line i of the M subsequent lines (where 0<=i<=M) contains an integer describing Di.

Constraint

1<=N<=10^6

1<=M<=2*10^6

1<=K<=2*10^6

1<=D[i]<=10^6

Sample Input 1:

10

5

2

6

9

3

2

7

Sample Output 1 :

5

Explanation:

Here he could cancel his 3rd and 4th obligation which makes vacation length 5.

Sample input 2:

7

2

0

3

4

Sample Output 2:

3

Explanation:

Here he could not cancel any obligation since K=0, so the vacation length is 3.

main.py

```
n = int(input())
m = int(input())
k = int(input())
arr = [0] * n
for i in range(m):
    arr[i] = int(input())
ans = 0
arr.sort()
if k > 0:
    for i in range(k + 1, m + 3, 1):
        ans = max(ans, arr[i] - arr[i - k - 1] - 1)
else:
    j = 0
    while arr[j] == 0:
        j = j + 1
    count = 0
    for i in range(1, n + 1, 1):
        count += 1
        if j < n and (i == arr[j]):
            count = 0

            j += 1
        ans = max(count, ans)
print(ans)
```

9) One of the first lessons IT students learn is the representation of natural numbers in the binary number system (base 2) This system uses only two digits, 0 and 1. In everyday life we use for convenience the decimal system (base 10) which uses ten digits, from 0 to 9. In general, we could use any numbering system.

Computer scientists often use systems based on 8 or 16. The numbering system based on K uses K digits with a value from 0 to K-1. Suppose a natural number M is given, written in the decimal system To convert it to the corresponding writing in the system based on K, we successively divide M by K until we reach a quotient that is less than K

The representation of M in the system based on K is formed by the final quotient (as first digit) and is followed by the remainder of the previous divisions For example :

If M=122 and K=8, 122 in base 10= 172 in base 8 This means that the number

In decimal system = 172 in octal system.

172 in base 8 = 1*8^2 + 7*8 + 2 = 122

You made the following observation in applying the above rule of converting natural numbers to another numbering system

In some cases in the new representation all the digits of the number are the same. For example 63 in base 10= 333 in base 4

Given a number M in its decimal representation, your task is find the minimum base B such that in the representation of M at base B all digits are the same.

Input Format

The first line contains an integer, M, denoting the number given

Constraints

1 <= M = 10^12

Sample Input 1 :

41

Sample Output 1 :

40

Explanation :

Here 41 in base 40. will be 11 so it has all digits the same, and there is no smaller base satisfying the requirements

Sample Input 2 :

34430

Sample Output 2 :

312

Explanation :

Here 34430 in base 312 will have all digits the same and there is no smaller base satisfying the requirements.

main.py

```
1  def convertBase(m, base):
2      rem = m % base
3      m = m // base
4      while m >= base and (m % base == rem):
5          m = m // base
6      if m == rem:
7          return True
8      return False
9
10
11 m = int(input())
12 base = 2
13 while not convertBase(m, base):
14     base = base + 1
15 print(base)
```

10) Wael is well-known for how much he loves the bitwise XOR operation, while kaito is well known for how much he loves to sum numbers, so their friend Resli decided to make up a problem that would enjoy both of them. Resil wrote down an array A of length N, an integer K and he defined a new function called Xor- sum as follows

Xor-sum(x)=(x XOR A[1])+(x XOR A[2])+(x XOR A[3])+……..+(x XOR A[N])

Can you find the integer x in the range [0,K] with the maximum Xor-sum (x) value?

Print only the value.

Input format

The first line contains integer N denoting the number of elements in A.

The next line contains an integer, k, denoting the maximum value of x.

Each line i of the N subsequent lines(where 0<=i<=N) contains an integer describing Ai.

Constraints

1<=N<=10^5

0<=K<=10^9

0<=A[i]<=10^9

Sample Input 1

1

0

989898

Sample Output 1

989898

Explanation:

Xor_sum(0)=(0^989898)=989898

Sample Input 2

3

7

1

6

3

Sample Output 2

14

Explanation

Xor_sum(4)=(4^1)+(4^6)+(4^3)=14.

Sample Input 3

4

9

7

4

0

3

Sample Output 3

46

Explanation:

Xor_sum(8)=(8^7)+(8^4) +(8^0)+(8^3)=46.

main.py

```
def Xor_sum(x, arr):
    xorSum = sum(arr)

    for i in range(1, x):
        s = 0
        for j in arr:
            s += i ^ j
        if s > xorSum:
            xorSum = s
    return xorSum


n = 4
x = 9
arr = [7, 4, 0, 3]
print(Xor_sum(x, arr))
```

⭕ Infosys Interview Experience ⭕

📌 Technical Round:

Asked Questions Based on their Resume and their area of Interest.

knowledge of at least one programming language.

Computer Fundamentals Like:

- **a) Operating Systems.**
- **b) Data Structures & Algorithms.**

Awareness Of the Latest Emerging Technologies.

Questions From your Projects/Internships and the role you played in them.

📌 HR Round:

The Objective here is essentially to assess whether you are the best fit for the company.

Question from your:

- **a) Background.**
- **b) Education.**
- **c) Hobbies.**
- **d) Even your View Of life!**

You must prepare for some Infosys company-related questions.

📌 Specialist Programmer:

1) Tell me About yourself.

2) Question From DBMS.

3) What is BCNF.

4) What is 3NF.

5) What are CPU Scheduling and its types.

6) What is Deadlock.

7) Prevention of Deadlock.

8) Multithreading Concept.

9) Some Coding Question.

10) 2-3 networking Question.

11) Question From Oops Concept (Inheritance, Polymorphism, Abstraction, and Encapsulation).

12) Do You Have Any Question to Me.

📌 Digital Specialist Engineer:

1) Tell me about yourself.

2) Explain projects you have done.

1) What is oops?

2) What are the four main principles of OOPS?

3) What is polymorphism?

4) What is inheritance?

5) What are the different types of inheritance?

6) What is abstraction?

7) What is encapsulation and how will you do it?

8) What is overloading and how it works?

9) What is method overriding?

10) What are the different types of access modifiers and explain each of them.

11) Will all properties of a parent can acquire by child class in inheritance?

1) What is DBMS?

2) How data is stored?

3) What is normalization?

4) Explain various normal forms in normalization?

5) Explain 2NF with an example.

6) What is a primary key?

7) What is a foreign key?

1) What is an algorithm?

2) Explain any sorting algorithm orally without code?

3) What is a linked list?

4) Explain the structure of the linked list.

5) What is linear data structure?

6) What is a non-linear data structure and examples?

1) What is IP Address?

2) What is MAC Address?

3) What are different types of protocols?