

Set 1

1) Nobita went for a coding competition. The first question he faced was to count the number of local minima in given array of integers. Of course, it was tough for him! Help him solve at least the first question.

Input Format:

The first line of input contains the number N, which represents the number of integers in array.

Next line consists of the integers which are part of array of size N.

Output Format:

Print the number of local minima in the array.

Note: Ignore the first and last elements of array while counting for local minima.

Sample Case:

Input:

7

45 12 34 15 32 43 21

Output:

2

Explanation:

Since local minima means number should be less than its neighbors, we have two such integers 12 and 15. 21 is not counted as we need to ignore first and last element of array.

Solution:

```
#include<bits/stdc++.h>
```

```
using namespace std;
```

```
int main()
```

```
{
```

```
    int n,c=0;
```

```
    cin>>n;
```

```
    vector<int>v(n);
```

```

for(int i=0;i<n;i++)
{
    cin>>v[i];
}
for(int j=1;j<n-1;j++)
{
    if(v[j]<v[j-1]&&v[j]<v[j+1])
        c++;
}
cout<<c;
return 0;
}

```

2) Suneo lives is a rich boy who lives in Tokyo. He is going for a trip to Hawaii. He will also take his friends along with him for the trip. His friends have to pass a test to be able to go for trip. They have been given an array of integers and their task is to find out the number of Squiro numbers in the array.

Squiro numbers – The number which is square of the index on which it is present on array. Indexing starts at 0.

Input Format:

The first line of input contains the number of integers in the array.

The second line of input contains N space separated integers which are members of array.

Output Format:

Print the number of Squiro numbers you found in array.

Sample Case:

Input:

6

3 1 7 9 32 12

Output:

2

Explanation:

Since we have to find numbers which are square of their indexes, we have two such numbers 1 and 9 which are squares of their indexes 1 and 3 respectively.

Solution:

```
#include<bits/stdc++.h>

using namespace std;

int main()
{
    int n,c=0;

    cin>>n;

    vector<int>v(n);

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

    for(int j=0;j<n;j++)
    {
        if(v[j]==j*j)
            c++;
    }

    cout<<c;

    return 0;
}
```

3) Nobita is someone who often plays with numbers. He is having N numbers with him which are all consecutive starting from 1 to N. But one of those numbers fell from his bag while he was running away from Gian. So, as he did not have much time, he forgot to pick the number and ran away. Later, he realized one of his numbers is missing. So, help him find the missing number.

Input Format:

The first line of input contains an integer N which denotes number of elements of array.

The second line of input contains an integral array of N-1 integers which is remaining with him.

Output Format:

Print the missing integer.

Sample Case:

Input:

5

1 5 4 2

Output:

3

Explanation:

Since numbers are consecutive (rearranged) and starts with 1 so 1, 2, 3, 4, 5 would be the numbers. 3 is missing from the set given to us, so it is the missing number.

Solution:

```
#include<bits/stdc++.h>
```

```
using namespace std;
```

```
int main()
```

```
{
```

```
    int n,s=0,so=0,mi=0;
```

```
    cin>>n;
```

```
    s=n*(n+1)/2;
```

```
    vector<int>v(n);
```

```
    for(int i=0;i<n-1;i++)
```

```
    {
```

```
        cin>>v[i];
```

```
    }
```

```
    for(int j=0;j<n-1;j++)
```

```
    {
```

```
        so=so+v[j];
```

```
}  
mi=s-so;  
cout<<mi;  
return 0;  
}
```

4) Dekisugi is good with numbers. He came up to you for help because he was feeling a program would help sort out things for him. He has an array of N integers and he wants to minimize the largest number of the array. You can only perform K number of operations on the array.

Each operation means choosing the largest element of array and decreasing it by 1.

Your goal is to minimize the biggest number of the array.

Input Format:

First line of input contains N and K.

Second line contains N space separated integers which are part of array.

Output Format:

Print the largest integer of array after performing these operations.

Sample Case:

Input:

3 5

5 8 2

Output:

4

Explanation: We have 5 operations to perform so performing we get the array as:

- 1) 5 7 2
- 2) 5 6 2
- 3) 5 5 2
- 4) 5 4 2
- 5) 4 4 2

So, we got 4 as the required result after we decreased the largest number by 1 in each step.

Solution:

```

#include<bits/stdc++.h>
#define ll long long int
using namespace std;
int main()
{
    ll n,k;
    cin>>n>>k;
    vector<ll>arr(n);
    for(ll i=0;i<n;i++)
    {
        cin>>arr[i];
    }
    for(ll j=0;j<k;j++)
    {
        *max_element(arr.begin(), arr.end())=*max_element(arr.begin(), arr.end())-1;
    }
    cout<<*max_element(arr.begin(), arr.end());
    return 0;
}

```

5) Doraemon and Nobita are each given a string by Doraemi. She tells that the number of characters common between both strings would signify their friendship. So, by this strange method by Doraemi, help them to calculate the number of common characters in both the strings.

Note: All types of character will be counted even if it's a space. If a character is repeated multiple times, count it 1 time only.

Input Format:

First and second line of input contains one string each, A in 1st line and B in 2nd line.

Output Format:

Print the number of common characters between those strings.

Sample Case:

Input 1:

Coding is fun

Code Programming

Output 1:

7

Explanation 1:

The common characters are 'C', 'o', 'd', 'i', 'n', 'g', ' ' (space)

Input 2:

Goodies

Good work

Output:

3

Explanation:

The common characters are 'G', 'o', 'd'. 'o' is present 2 times in string A and 3 times in string B. But it's counted only once.

Solution:

```
#include<bits/stdc++.h>
```

```
using namespace std;
```

```
std::vector<char> strIntersect(std::string const&s1, std::string const&s2){
```

```
    std::vector<bool> presents(256, false); //Assuming ASCII
```

```
    std::vector<char> intersection;
```

```
    for (auto c : s1) {
```

```
        presents[c] = true;
```

```
    }
```

```
    for (auto c : s2) {
```

```
        if (presents[c]){
```

```
            intersection.push_back(c);
```

```
            presents[c] = false;
```

```
        }
```

```

    }
    return intersection;
}

int main() {
    int count=0;
    std::vector<char> result;
    std::string s1 = "Coding is fun";
    std::string s2 = "Code Programming";
    result = strIntersect(s1, s2);
    for (auto c : result) {
        count++;
    }
    std::cout << count;
    return 0;
}

```

6) Shizuka was given an array of integers by her teacher and her task was to find out whether a pair of integers in the array whose difference equals to the target difference provided by her teacher exists or not. She has approached you for your help.

Input Format:

First line of input contains the number of integers N and the target difference D.

Second line contains the elements of array.

Output Format:

Print YES or NO as applicable.

Note: Try to solve this problem in $O(N)$ complexity.

Sample Case:

Input:

5 3

8 2 7 4 6

Output:

YES

Explanation:

(7, 4) is the pair we were looking for since their difference is 3. As, we got at least 1 such pair we would print YES.

Solution:

```
#include<bits/stdc++.h>

using namespace std;

bool isPair(vector<int>arr, int diff)
{
    unordered_set<int>comp;
    for(int value:arr)
    {
        if(comp.find(value)!=comp.end())
            return true;
        comp.insert(diff+value);
    }
    return false;
}

int main()
{
    int n,k;
    cin>>n>>k;
    vector<int>a(n);
    for(int i=0;i<n;i++)
        cin>>a[i];
    if(isPair(a, k))
        cout<<"YES";
    else
```

```
    cout<<"NO";  
    return 0;  
}
```

7) Nobita was once solving a coding problem which was given as a challenge to him by Suneo. Since he wanted to prove himself in front of his friends, he was trying hard to solve the problem but was not successful. So, he approached you for his help. Suneo has given Nobita a string of characters and his task is to find the length of longest integral substring (substring which has only integers).

Input Format:

The first line of input contains a string S.

Output Format:

Print the length of longest integral substring.

Sample Case:

Input:

Code1710now20is00good5

Output Format:

4

Explanation:

Since there are many integral parts but longest continuous integral substring was 1710 which consisted of 4 characters. Others like 20, 00 and 5 are shorter.

Solution:

```
#include <bits/stdc++.h>
```

```
using namespace std;
```

```
int longestInteger(string str, int l)
```

```
{
```

```
    int count = 0, max = 0, pos = -1, pre_pos, pre_len, len = 0;
```

```
    // Traverse the string
```

```
    for (int i = 0; i < l; i++) {
```

```
        // Store the previous position and previous length
```

```

        // of the digits encountered.
        pre_pos = pos;
        pre_len = len;
        count = 0;
        len = 0;
        // If first digit occurs, store its position in pos
        if (isdigit(str[i]))
            pos = i;
        // Traverse the string till a character occurs.
        while (isdigit(str[i])) {
            count++;
            i++;
            len++;
        }
        // Check if the length of the string is
        // greater than the previous ones or not.
        if (count > max) {
            max = count;
        }
        else {
            pos = pre_pos;
            len = pre_len;
        }
    }

    return ((str.substr(pos, len)).length());
}

int main()
{
    string str;

```

```

        cin>>str;

        int l = str.length();

        cout << longestInteger(str, l);

        return 0;

}

```

8) Dekisugi is working on a new coding problem in which his task is to accept an array of integers from the user and a target number. His task is to find the length of smallest contiguous subarray whose sum equals to or greater than the target number.

Input Format:

The first line of input contains the number of integers in array N and target number M.

The second line contains N space separated integers which are elements of array.

Output Format:

Print the length of smallest such subarray.

Sample Case:

Input:

5 9

3 1 6 2 5

Output:

3

Explanation:

1 6 2 is the contiguous sub array whose sum equals or greater than 9 so it is chosen. Its length is 3.

Solution:

```
#include <iostream>
```

```
using namespace std;
```

```
int smallestSubWithSum(int arr[], int n, int x)
```

```
{
```

```
    int curr_sum = 0, min_len = n+1;
```

```

int start = 0, end = 0;
while (end < n)
{
    // Keep adding array elements while current sum
    // is smaller than x
    while (curr_sum < x && end < n)
        curr_sum += arr[end++];

    // If current sum becomes greater than x.
    while (curr_sum >= x && start < n)
    {
        // Update minimum length if needed
        if (end - start < min_len)
            min_len = end - start;

        // remove starting elements
        curr_sum -= arr[start++];
    }
}

return min_len;
}

int main()
{
    int n1,x;
    cin>>n1>>x;
    int arr1[n1];
    for(int i=0;i<n1;i++)
        cin>>arr1[i];

    int res1 = smallestSubWithSum(arr1, n1, x);
    (res1 == n1+1)? cout << "Not possible\n" :
        cout << res1 << endl;
}

```

```
        return 0;
    }
```

9) Shizuka is working on her summer project. She was developing a program which took an integral array as input and her task was to find Nobinal numbers in the array.

Nobinal Number – A number is said to be a Nobinal number if it occurs more than $N/3$ times in the array.

Count the number of Nobinal numbers and print the number of Nobinal numbers in the array.

Input Format:

The first line of input is an integer N which denotes number of elements in array.

The second line contains N space separated integers which are elements of array.

Output Format:

Print any one Nobinal number in the array.

Sample Case:

Input:

5

2 7 2 3 7

Output:

2 (or 7)

Explanation:

Since $5/3 = 1.67$ so there are two numbers in array which occurs more than 1.67 times. The numbers are 2 and 7.

Solution:

```
#include <bits/stdc++.h>
```

```
using namespace std;
```

```
int appearsNBy3(int arr[], int n)
```

```
{
```

```
    int count1 = 0, count2 = 0;
```

```
    int first=INT_MAX , second=INT_MAX ;
```

```

for (int i = 0; i < n; i++) {
    // if this element is previously seen,
    // increment count1.
    if (first == arr[i])
        count1++;

    // if this element is previously seen,
    // increment count2.
    else if (second == arr[i])
        count2++;

    else if (count1 == 0) {
        count1++;
        first = arr[i];
    }

    else if (count2 == 0) {
        count2++;
        second = arr[i];
    }

    // if current element is different from
    // both the previously seen variables,
    // decrement both the counts.
    else {
        count1--;
        count2--;
    }
}

count1 = 0;
count2 = 0;

// Again traverse the array and find the
// actual counts.

```

```

        for (int i = 0; i < n; i++) {
            if (arr[i] == first)
                count1++;

            else if (arr[i] == second)
                count2++;
        }

        if (count1 > n / 3)
            return first;

        if (count2 > n / 3)
            return second;

        return -1;
    }

    int main()
    {
        int n;

        cin>>n;

        int arr[n];

        for(int i=0;i<n;i++)
            cin>>arr[i];

        cout << appearsNBy3(arr, n) << endl;

        return 0;
    }

```

10) Nobita and Doraemon are building some new programs again! This time they created some program which would take input in the form of an array of strings and their program will create a new string which would contain all the letters of all those substrings. Find the length of shortest such string.

Input format:

First line of input contains the number of strings in the array.

Second line of input contains the string inputs.

Output Format:

Print the length of shortest string which contains all characters used in all the strings.

Sample Case:

Input:

4

Goodboyishere

Output:

10

Explanation:

Godbyisher is the required string which contains all characters of the strings.

Solution:

```
#include<bits/stdc++.h>
using namespace std;
int main()
{
    string s;
    cin>>s;
    set <char>s1;
    for(int i=0;i<s.length();i++)
    {
        s1.insert(s[i]);
    }
    cout<<s1.size();
    return 0;
}
```

