Problem no.: 2

Problem link: http://codeforces.com/problemset/problem/141/A

Problem description:

So, the New Year holidays are over. Santa Claus and his colleagues can take a rest and have guests at last. When two "New Year and Christmas Men" meet, their assistants cut out of cardboard the letters from the guest's name and the host's name in honor of this event. Then the hung the letters above the main entrance. One night, when everyone went to bed, someone took all the letters of our characters' names. Then he may have shuffled the letters and put them in one pile in front of the door. The next morning it was impossible to find the culprit who had made the disorder. But everybody wondered whether it is possible to restore the names of the host and his guests from the letters lying at the door?
That is, we need to verify that there are no extra letters, and that nobody will need to cut more letters. Help the "New Year and Christmas Men" and their friends to cope with this problem. You are given both inscriptions that hung over the front door the previous night, and a pile of letters that were found at the front door next morning.

Input:

The input file consists of three lines: the first line contains the guest's name, the second line contains the name of the residence host and the third line contains letters in a pile that were found at the door in the morning. All lines are not empty and contain only uppercase Latin letters. The length of each line does not exceed 100.

Output:

Print "YES" without the quotes, if the letters in the pile could be permuted to make the names of the "New Year and Christmas Men". Otherwise, print "NO" without the quotes.

Sample Input:

Case 1:
SANTACLAUS
DEDMOROZ
SANTAMOROZDEDCLAUS

Case 2:
PAPAINOEL
JOULUPUKKI
JOULNAPAOILELUPUKKI

Case 3:
BABBONATALE
FATHERCHRISTMAS
BABCHRISTMASBONATALLEFATHER

Case 4:
OI
M
IMO

Sample Output:

Case 1: YES
Case 2: NO
Case 3: NO
Case 4: YES

Code:

```
#include<iostream>
#include<cstring>
using namespace std;
int main(void)
{
    //declaring variables
    char guest[102];
    char host[102];
    char pile[102];
    int i,j;
    char temp;
    //input
    cin>>guest;
    cin>>host;
    cin>>pile;
    //processing
    strcat(guest,host);
    for(i=0;guest[i]!='\0';i++)
    {
        for(j=i;guest[j]!='\0';j++)
        {
            if(guest[i]>guest[j])
            {
                temp=guest[i];
                guest[i]=guest[j];
                guest[j]=temp;
            }
        }
    }
    for(i=0;pile[i]!='\0';i++)
    {
        for(j=i;pile[j]!='\0';j++)
        {
            if(pile[i]>pile[j])
            {
                temp=pile[i];
                pile[i]=pile[j];
                pile[j]=temp;
            }
        }
    }
    i=strcmp(guest,pile);
    if(i==0)
    cout<<"YES"<<endl;
    else
    cout<<"NO"<<endl;
    return 0;
}
```

Analysis:

1. Read input for the array guest.
2. Read input for the array host.
3. Read input for the array pile.
4. Add the array host with the array guest using strcat() function.
5. Sort the array guest, here, Bubble sort has been used.
6. Sort the array pile, here, Bubble Sort has been used.
7. Compare the array pile and guest using strcmp() function and read the output to the variable i.
8. if i=0, then print YES
   else, print NO.

Complexity:

1. Line 13-15: O(1)
2. Line 18-29: O(n^2)
3. Line 30-41 :O(n^2)
4. Line 42: O(1)
5. Line 43: O(1)

   So, total complexity is O(n^2)

Problem no.: 3

Problem link: http://codeforces.com/problemset/problem/149/A

Problem description:

What joy! Petya's parents went on a business trip for the whole year and the playful kid is left all by himself. Petya got absolutely happy. He jumped on the bed and threw pillows all day long, until...
Today Petya opened the cupboard and found a scary note there. His parents had left him with duties: he should water their favourite f lower all year, each day, in the morning, in the afternoon and in the evening.
"Wait a second!" — thought Petya. He know for a fact that if he fulfills the parents' task in the i-th ($1 \le i \le 12$) month of the year, then the flower will grow by a centimeters, and if he doesn't water the flower in the i-th month, then the flower won't grow this month. Petya also knows that try as he might, his parents won't believe that i he has been watering the flower if it grows strictly less than by k centimeters.
Help Petya choose the minimum number of months when he will water the flower, given that the flower should grow no less than by k centimeters.

Input

The first line contains exactly one integer k ($0 \le k \le 100$). The next line contains twelve space-separated integers: the i-th ($1 \le i \le 12$) number in the line represents a ($0 \le a \le 100$).

Output

Print the only integer — the minimum number of months when Petya has to water the flower so that the flower grows no less than by k centimeters. If the flower can't grow by k centimeters in a year, print -1.

Sample Input:
Case 1:
5
1 1 1 1 2 2 3 2 2 1 1 1

Case 2:
15
20 1 1 1 1 2 2 1 2 2 1 1

Case 3:
52
1 12 3 11 4 5 10 6 9 7 8 2

Case 4:
50
2 2 3 4 5 4 4 5 7 3 2 7

Sample Output:

Case 1: 2
Case 2: 1
Case 3: 6
Case 4: -1

Code:

```
#include<bits/stdc++.h>
using namespace std;

int main(void)
{
      int k,i;
      int j,temp;
      int a[12];
      int month=0;
      cin>>k;
      for(i=0;i<12;i++)
      cin>>a[i];
      for(i=0;i<12;i++)
      {
            for(j=i;j<12;j++)
            {
                  if(a[i]<a[j])
                  {
                        temp=a[i];
                        a[i]=a[j];
                        a[j]=temp;
                  }
            }
      }
      i=0;
      while(k>0&&i<12)
      {
            k -= a[i];
            ++i;
            ++month;
      }
       if (k > 0)
            cout << "-1";
      else
            cout << month;
      return 0;
}
```

Analysis:

1. Set the value of the variable value to 0;
2. Input value to the variable k.
3. Input value to the array a.
4. Sort the array a, here, Bubble Sort has been used.
5. Set the value of the variable to 0.
6. Subtract the value of a[i] from the variable k.
7. Increase the value of the variable i by 1.
8. Increase the value of the variable month by 1.
9. if the value of k is greater than 0 and the value of i is less than 12, then repeat from the step 6.
   else, go to next step.
10. if the value of k is greater than k, then print -1.
    else, print the value of the variable month.

Complexity:

1. Line 9: O(1)
2. Line 11-12: O(n)
3. Line 13-24: O(n^2)
4. Line 25: O(1)
5. Line 26-31: O(n)

   So, total complexity is O(n^2).

Problem no.: 4

Problem link: http://codeforces.com/problemset/problem/432/A

Problem Description:

The Saratov State University Olympiad Programmers Training Center (SSU OPTC) has n students. For each student you know the number of times he/she has participated in the ACM ICPC world programming championship. According to the ACM ICPC rules, each person can participate in the world championship at most 5 times. The head of the SSU OPTC is recently gathering teams to participate in the world championship. Each team must consist of exactly three people, at that; any person cannot be a member of two or more teams.
What maximum number of teams can the head make if he wants each team to participate in the world championship with the same members at least k times?

Input:

The first line contains two integers, n and k ($1 \le n \le 2000$; $1 \le k \le 5$). The next line contains n integers: y_1, y_2, ..., y_3($0 \le y \le 5$), where y shows the number of times the i-th person participated in the ACM ICPC world championship.

Output:

Print a single number — the answer to the problem.

Sample Input:

Case 1:

```
5 2
0 4 5 1 0
```

Case 2:

```
12 2
0 1 2 3 4 0 1 2 3 4 0 1
```

Case 3:

```
15 2
0 1 2 3 4 0 1 2 3 4 0 1 2 3 4
```

Case 4:

```
2000 2
0 0 1 2 2 5 3 3 5 2 0 3 4 3 5 4 0 4 5 3 4 1 2 5 3 4 4 3 4 0 5 1 0 1 4 5 3 2 4 5 3 0 3 4 4 1 2 3 3 1 1 1 0 5 3 0 3
4 0 1 2 2 4.................
```

Sample Output:

```
Case 1: 1
Case 2: 3
Case 3: 4
Case 4: 437
```

Code:

```
#include<bits/stdc++.h>
using namespace std;

int main(void)
{
    int n,k;
    cin>>n>>k;
    int y[n];
    int i,j,temp;
    for(i=0;i<n;i++)
    {
        cin>>temp;
        y[i]=temp+k;
    }
    for(i=0;i<n;i++)
    {
        for(j=i;j<n;j++)
        {
            if(y[i]>y[j])
            {
                temp=y[i];
                y[i]=y[j];
                y[j]=temp;
            }
        }
    }
    for(i=0;y[i]<6;)
    {
        ++i;
    }
    cout<<i/3;

    return 0;
}
```

Analysis:

1. Input to the variable n and k.
2. Declare an array of size n.
3. Input to the variable temp.
4. Input the sum of the value of the variable temp and the variable k to the array.

5. Continue step 3 and 4 until the array is full.
6. Sort the array in descending order, here, Bubble Sort has been used.
7. Increase the value of i by one until value of element of the array is less than 6.
8. Output the value of i divided by 3.

Complexity:

1. Line 7: O(2)
2. Line 10-14: O(n)
3. Line 15-26: O(n^2)
4. Line 27-30: O(n)
   So, total complexity is O(n^2).

Problem no.: 5

Problem link: http://codeforces.com/problemset/problem/723/A

Problem description:

There are three friends living on the straight line Ox in Lineland. The first friend lives at the point x, the second friend lives at the point x, and the third friend lives at the point x. They plan to celebrate the New Year together, so they need to meet at one point.
What is the minimum total distance they have to travel in order to meet at some point and celebrate the New Year? It's guaranteed that the optimal answer is always integer.

Input:

The first line of the input contains three distinct integers x , x and x (1 ≤ x , x , x ≤ 100) — the coordinates of the houses of the first, the second and the third friends respectively.

Output:

Print one integer — the minimum total distance the friends need to travel in order to meet together.

Sample Input:

Case 1: 7 1 4
Case 2: 30 20 10
Case 3: 1 4 100
Case 4: 71 85 88

Sample Output:

Case 1: 6
Case 2: 20
Case 3: 99
Case 4: 17

Code:

```
#include<bits/stdc++.h>
using namespace std;

int main(void)
{
        int x[3];
        int i,j,temp;
        for(i=0;i<3;i++)
                cin>>x[i];
        for(i=0;i<3;i++)
```

```
    {
        for(j=i;j<3;j++)
        {
            if(x[i]>x[j])
            {
                temp=x[i];
                x[i]=x[j];
                x[j]=temp;
            }
        }
    }
    cout<<(x[1]-x[0])+(x[2]-x[1]);
    return 0;
}
```

Analysis:

1. Input to the array.
2. Sort the array in descending order.
3. Output the sum of difference between the middle element and the other elements.

Complexity:

1. Line 8-9: O(n)
2. Line 10-21: $O(n^2)$

   So, total complexity is $O(n^2)$.