

Problem No: 01**Problem Link:** <http://codeforces.com/problemset/problem/69/A>**Problem Description:**

A guy named Vasya attends the final grade of a high school. One day Vasya decided to watch a match of his favorite hockey team. And, as the boy loves hockey very much, even more than physics, he forgot to do the homework. Specifically, he forgot to complete his physics tasks. Next day the teacher got very angry at Vasya and decided to teach him a lesson. He gave the lazy student a seemingly easy task: You are given an idle body in space and the forces that affect it. The body can be considered as a material point with coordinates (0; 0; 0). Vasya had only to answer whether it is in equilibrium. "Piece of cake" — thought Vasya, we need only to check if the sum of all vectors is equal to 0. So, Vasya began to solve the problem. But later it turned out that there can be lots and lots of these forces, and Vasya can not cope without your help. Help him. Write a program that determines whether a body is idle or is moving by the given vectors of forces.

Input:

The first line contains a positive integer n ($1 \leq n \leq 100$), then follow n lines containing three integers each: the x_i coordinate, the y_i coordinate and the z_i coordinate of the force vector, applied to the body ($-100 \leq x_i, y_i, z_i \leq 100$).

Output:

Print the word "YES" if the body is in equilibrium, or the word "NO" if it is not.

Sample Input:

Case 1:

3

4 1 7

-2 4 -1

1 -5 -3

Case 2:

10

21 32 -46

43 -35 21

42 2 -50

22 40 20

-27 -9 38

.....

Case 3:

13

-3 -36 -46

-11 -50 37

42 -11 -15

9 42 44

-29 -12 24

3 9 -40

.....

Sample Output:

Case 1: NO

Case 2: NO

Case 3: YES

Code:

```
#include<bits/stdc++.h>
using namespace std;
class Space
{
public:
    int x;
    int y;
    int z;
};
int main(void)
{
    int n;
    int i;
    int fx=0;
    int fy=0;
    int fz=0;
    cin>>n;
    Space force[n];
    for(i=0;i<n;i++)
    {
        cin>>force[i].x>>force[i].y>>force[i].z;
        fx+=force[i].x;
        fy+=force[i].y;
        fz+=force[i].z;
    }
    if(fx==0&&fy==0&&fz==0)
        cout<<"YES";
    else
        cout<<"NO";
}
```

Analysis:

01. Declare a class with three public integer members.
02. Set the value of the variables fx, fy and fz to 0.
03. Take input to the variable n.
04. Take inputs to the object members n times and add them to the corresponding variables.
05. If fx, fy and fz, all are 0, then print YES; else print NO.

Complexity:

01. Line 14: O(1)

- 02. Line 15: $O(1)$
- 03. Line 16: $O(1)$
- 04. Line 17: $O(1)$
- 05. Line 19-25: $O(n)$
- 06. Line 26: $O(3)$
- 07. Line 27: $O(1)$
- 08. Line 29: $O(1)$

So, complexity of this program is $O(n)$.

Problem no.: 02

Problem Link: <http://codeforces.com/problemset/problem/546/A>

Problem Description:

A soldier wants to buy w bananas in the shop. He has to pay k dollars for the first banana, $2k$ dollars for the second one and so on (in other words, he has to pay $i \cdot k$ dollars for the i -th banana). He has n dollars. How many dollars does he have to borrow from his friend soldier to buy w bananas?

Input:

The first line contains three positive integers k, n, w ($1 \leq k, w \leq 1000$, $0 \leq n \leq 10^9$), the cost of the first banana, initial number of dollars the soldier has and number of bananas he wants.

Output:

Output one integer — the amount of dollars that the soldier must borrow from his friend. If he doesn't have to borrow money, output 0.

Sample Input:

Case 1: 3 17 4
Case 2: 1 2 1
Case 3: 1 5 6
Case 4: 1000 0 1000

Sample Output:

Case 1: 13
Case 2: 0
Case 3: 16
Case 4: 500500000

Code:

```

#include<bits/stdc++.h>
using namespace std;
int main(void)
{
    int w,i,k;
    unsigned long long int n,total=0;
    cin>>k>>n>>w;
    for(i=1;i<=w;i++)
        total+=(k*i);
    if(total>n)
        cout<<total-n;
    else
        cout<<"0";
}

```

Analysis:

01. The variable total is initialized to 0.
02. Take inputs to k, n and w.
03. Starting the value of I from 1, multiply it with k and add the result to the variable total.
04. Increase the value of i by 1 and continue step 3 until I is greater than w.
05. If total is greater than n, then print the result of difference between the variables total and n; else, print 0.

Complexity:

01. Line 06: $O(1)$
 02. Line 07: $O(3)$
 03. Line 08-09: $O(n)$
 04. Line 10-11: $O(1)$
 05. Line 12-13: $O(1)$
- So, Complexity of this program is $O(n)$.

Problem no.: 03

Problem Link: <http://codeforces.com/problemset/problem/486/A>

Problem Description:

For a positive integer n let's define a function f :

$$f(n) = -1 + 2 - 3 + \dots + (-1)^n n$$

Your task is to calculate $f(n)$ for a given integer n .

Input:

The single line contains the positive integer n ($1 \leq n \leq 10^{15}$).

Output:

Print $f(n)$ in a single line.

Sample Input:

Case 1: 4
Case 2: 5
Case 3: 1000000000
Case 4: 1000000001

Sample Output:

Case 1: 2
Case 2: -3
Case 3: 500000000
Case 4: -500000001

Code:

```
#include<bits/stdc++.h>
using namespace std;
int main(void)
{
    long long int n;
    cin>>n;
    if(n%2==0)
        cout<<n/2;
    else
        cout<<(n/2)-n;
}
```

Analysis:

01. Take input to the variable n.
02. If modulus of n by 2 is 0, then, print the value of n divide by 2. Else, print the value of difference between n divided by 2 and n.

Complexity:

01. Line 06: $O(1)$
 02. Line 07-08: $O(2)$
 03. Line 09-10: $O(2)$
- So, the program is of linear complexity.

Problem no.: 04

Problem Link: <http://codeforces.com/problemset/problem/617/A>

Proble Description:

An elephant decided to visit his friend. It turned out that the elephant's house is located at point 0 and his friend's house is located at point x ($x > 0$) of the coordinate line. In one step the

elephant can move 1, 2, 3, 4 or 5 positions forward. Determine, what is the minimum number of steps he need to make in order to get to his friend's house.

Input:

The first line of the input contains an integer x ($1 \leq x \leq 1\,000\,000$) – The coordinate of the friend's house.

Output:

Print the minimum number of steps that elephant needs to make to get from point 0 to point x .

Sample Input:

Case 1: 5
Case 2: 12
Case 3: 999999
Case 4: 1000000

Sample Output:

Case 1: 1
Case 2: 3
Case 3: 200000
Case 4: 200000

Code:

```
#include<bits/stdc++.h>
using namespace std;
int main(void)
{
    unsigned int x,i,step=0;
    cin>>x;
    for(i=5;i>0;i--)
    {
        step+=x/i;
        x%=i;
        if(x==0)
            break;
    }
    cout<<step;
}
```

Analysis:

01. Initialize the value of the variable to 0.
02. Take input to the variable x .
03. Initialize the value of i to 5.
04. Divide x by i and add the integer value to $step$.
05. Take the modulus of x by i and store it in the variable x .
06. If the value of x is equal to 0, then, skip step 7.

07. Decrease the value of i by 1 and continue step 4,5 and 6 till the value of i is greater then 0.
08. Print the value of the variable step.

Complexity:

01. Line 05: $O(1)$
 02. Line 06: $O(1)$
 03. Line 07-13: $O(n)$
 04. Line 14: $O(1)$
- So, complexity of this program is $O(n)$.

Problem No.: 05

Problem Link: <http://codeforces.com/problemset/problem/750/A>

Problem Description:

Limak is going to participate in a contest on the last day of the 2016. The contest will start at 20:00 and will last four hours, exactly until midnight. There will be n problems, sorted by difficulty, i.e. problem 1 is the easiest and problem n is the hardest. Limak knows it will take him $5 \cdot i$ minutes to solve the i -th problem.

Limak's friends organize a New Year's Eve party and Limak wants to be there at midnight or earlier. He needs k minutes to get there from his house, where he will participate in the contest first.

How many problems can Limak solve if he wants to make it to the party?

Input:

The only line of the input contains two integers n and k ($1 \leq n \leq 10$, $1 \leq k \leq 240$) — the number of the problems in the contest and the number of minutes Limak needs to get to the party from his house.

Output:

Print one integer, denoting the maximum possible number of problems Limak can solve so that he could get to the party at midnight or earlier.

Sample Input:

Case 1: 3 222
Case 2: 4 190
Case 3: 10 135
Case 4: 9 240

Sample Output:

Case 1: 2

Case 2: 4

Case 3: 6

Case 4: 0

Code:

```
#include<bits/stdc++.h>
using namespace std;
int main(void)
{
    int n,k,cnt;
    int time=0;
    int i=0;
    cin>>n>>k;
    k=240-k;
    while(i<=n)
    {
        time+=(i*5);
        if(time>k)
            break;
        else
        {
            cnt=i;
            i++;
        }
    }
    cout<<cnt;
}
```

Analysis:

01. Initialize the value of the variable time to 0.
02. Initialize the value of the variable i to 0.
03. Take input to the variable n and k.
04. Store the difference between 240 and k to k.
05. Multiply i by 5 and add it to the variable time.
06. If the value of the variable time is greater than k, then move to step 7. Else, set the value of the variable cnt equal to i, increase the value of i by 1 and continue from step 5 till the value of i is less than or equal to the value of the variable n.
07. Print the value of the variable of cnt.

Complexity:

01. Line 06: $O(1)$
 02. Line 07: $O(1)$
 03. Line 08: $O(2)$
 04. Line 09: $O(2)$
 05. Line 10-20: $O(n)$
 06. Line 21: $O(1)$
- So, complexity of this program is $O(n)$.