

Junior Division

Problem A: Welcome to BUBT IUPC

Setter : Quwsar Ohi

Alternate Solution Writer: None

Solution: As the problem description says, just copy the code and submit it. This was the hardest problem of this contest.

Number of Acceptance: Junior-68, Senior-29.

Problem B: Homework

Setter : Nazmul Hoda

Alternate Solution Writer: Maruf Ahmed Rahad, Walid Khan Jim, M M Mehedi Hasan

Solution: The problem requires 'if'-conditional operation. Just take two inputs **x**, and **y**. And check if **y < x**. If the condition is true, then print 'Hello,BUBTian', otherwise print 'Nothing'.

Number of Acceptance: Junior-68

Problem C: Easy??

Setter : Md. Saifur Rahman

Alternate Solution Writer: Maruf Ahmed Rahad, Quwsar Ohi

Solution: The problem setter asks you to find the divisible of 3, that is greater than the input value **x**. It can be done by checking three conditions such as,

```
if((x+1)%3 == 0) {  
    printf("%d\n", x+1);  
}  
else if((x+2)%3 == 0) {  
    printf("%d\n", x+2);  
}  
else {  
    printf("%d\n", x+3);  
}
```

Number of Acceptance: Junior-51.

Problem D: **The Gourmandiser Initiative!**

Setter : Nazmul Alam Nayan

Alternate Solution Writer: Walid Khan Jim, Quwsar Ohi

Solution: By reading the problem statement carefully, we can understand that there are fire persons and you have to buy food for six persons. So if you buy one food, you need to buy for six people. So, if you have to buy 'n' unit of food, you have to buy $n*6$ of them. So the final answer is $n*6$, which is to be printed using printf function.

Number of Acceptance: Junior-61.

Problem E: **Three Consecutive Numbers**

Setter : Elias Hassan Naim

Alternate Solution Writer: Walid Khan Jim, Maruf Ahmed Rahad, M M Mehedi Hasan

Solution: Let's solve the three equations,

$$1. \quad X = Y-1 \quad (1)$$

$$2. \quad Z = Y+1 \quad (2)$$

$$3. \quad X+Y+Z = N \quad (3)$$

By placing eqn (1) and (2) in equation 3, we get,

$$\Rightarrow Y-1+Y+Y+1 = N$$

$$\Rightarrow 3Y = N$$

$$\Rightarrow Y = N/3$$

As X, Y, and Z are integers, $N\%3$ must be zero. I.e., $N\%3 == 0$.

So find the solution you must check if N is divisible by 3. If so, Y would be $N/3$. So, $X = Y-1$, and $Z = Y+1$. If N is not divisible by 3, then there is no solution, in this case, print -1.

Number of Acceptance: Junior-46, Senior-20.

Problem F: **Selecting Guard (I)**

Setter : Maruf Ahmed Rahad

Alternate Solution Writer: M M Mehedi Hasan

Solution: Let's assume there are 10 stations, so $N=10$. If we place a guard at 2nd stations, he can cover 1st and 3rd stations. As every person can guard 2 stations, when N is even the answer will be $N/2$. When N is odd, answer will be $(N-1)/2$. Except when $N=1$, the answer will be 1 as we need at least one guard.

```
if(N==1) {
    printf("1\n");
} else if(N%2 == 0) {
    printf("%d", N/2);
} else {
    printf("%d", (N-1)/2);
}
```

Number of Accepted Submissions: Junior-30.

Problem G: **Emu's Favorite**

Setter : Nazmul Hoda

Alternate Solution Writer: Jim, Maruf, Mehedi

Solution: It can be proved that if any number $N\%7=0$, then the number N can be obtained by repetitive addition of 7. The same goes for -7 too.

```
if(N%7 == 0){
    printf("YES\n");
} else{
    printf("NO\n");
}
```

Number of Accepted Solutions: 44

Problem H: **K'th Element (I)**

Setter : Quwsar Ohi

Alternate Solution Writer: Maruf, Mehedi

Solution: The input numbers are already sorted. Take the numbers in an array and then for each query simply print the $(L+k-1)^{\text{th}}$ value.

Number of Accepted Solutions: 7

Problem I: **Easy Balance**

Setter : Khondokar Tanbir Ahmed

Alternate Solution Writer: Jim, Maruf, Mehedi

Solution: The integers that need to be transformed may range from -100 to 100(included). As the possible range is only 200, you can assume a value and then calculate the required cost for that value. Thus we can find out the minimum possible cost to transform.

```
minCost = 10^18
for(i=-100; i<101; i++){
    cost = 0
    for(j=0; j<N; j++){
        cost = cost + (i - a[j])^2
    }
    minCost = min(cost, minCost)
}
```

Number of Accepted Solutions: 0

Problem J: **My Younger Sister**

Setter : M M Mehedi Hasan

Alternate Solution Writer: Jim, Maruf

Solution: Given an array of N integers, you have to find the sum of integers in index range L to R. After taking the values in an array, you have to calculate **Cumulative Sum** and save that in another array, **Arr**. For each query with L and R, the required answer is , **Arr[R]-Arr[L-1]**

Let the numbers be: 3 6 2 9 10

Cumulative Sum: 3 9 11 20 30

If, L=2 and R = 4, then the answer will be, **Arr[R]-Arr[L-1]** or **20-3=17**

Note that the summation can be $10^5 * 10^5$

Number of Accepted Solutions: Junior-8.

Problem K: **Century!!**

Setter : Walid Khan Jim

Alternate Solution Writer: Maruf, Ohi, Mimsad

Solution: You should take the input number as a string because the size of this number is 10^{100} . If both of the last 2 digits are 0 then the result will be the number without the last 2 digits, Otherwise you should add 1 with the number without the last 2 digits.

Number of Accepted Solutions: Junior-2.

Problem L: **Sorting a Binary Array**

Setter : Elias Hassan Naim

Alternate Solution Writer: Mehedi, Jim

Solution: It's a constructive algorithm. If the value of any index is 0, then You should count the total number of 1, which are located in the left side of this 0. You can apply the "Cumulative Sum" method to get the count of 1.

Number of Accepted Solutions: Junior-2, Senior-5.

Problem M: **Rectangle Counting**

Setter : Md. Farid Miah

Alternate Solution Writer: M M Mehedi Hasan

Solution: As all rectangles are parallel to the axis, we should check all pairs of points, which are located in the same X axis. If we find such a pair of points like as $(x1, y1)$ and $(x2, y2)$, which are parallel to the X axis, then You have to find such two points for making a rectangle are $(x1, y3)$ and $(x2, y3)$.

Accepted Solutions: Junior-0, Senior-0.

Senior Division

Problem C: Hardest Calculation Ever!!!

Setter : Monzurul Islam

Alternate Solution Writer: Jim, Maruf, Mehedi

Solution: It's a straight forward problem , The ans is valid , if($Y \% X == 0$) then the ans is valid, Otherwise Invalid. ($Y \geq X$). if $X > Y$, then first swap(X, Y).

Number of Accepted Solutions: Senior-6.

Problem D: Little Hackers

Setter : Mimsad Ahmed Hridoy

Alternate Solution Writer: Jim, Maruf, Mehedi

Solution: You have 2 strings s_1 and s_2 . For every possible index of s_1 , You have to find the maximum length of substring from s_2 .

You can solve it with nested loops.

Number of Accepted Solutions: Senior-1.

Problem E: Selecting Guard (II)

Setter : Maruf Ahemd Rahad

Alternate Solution Writer: Mehedi, Mimsad

Solution: This is a straight forward Bipartite graph check problem. If it is a Bipartite graph, then You have to bicolor(black and white) it and print the minimum number of the same color for every component. If a component has only 1 node, then count 1.

So,

$$\text{result} = \max(1, \min(\text{number_of_black_color}, \text{number_of_white_color}))$$

Number of Accepted Solutions: Senior-2.

Problem F: Mr. MMMH and Suborno

Setter : M M Mehedi Hasan

Alternate Solution Writer: Maruf Ahmed Rahad

Solution: You have to pre-calculate all the summation of divisors from 1 to n in $(n * \log(n))$ time. You can do it by divisor sieve. Then you have to sort all in ascending order. Now You have to apply cumulative sum and find the answer for K amount by binary search.

Number of Accepted Solutions: Senior-1.

Problem H: **Cost Count**

Setter : Murad Al Wazed

Alternate Solution Writer: Mimsad Ahmed Hridoy

Solution: The problem states that there is a matrix of size $(H*W)$. This matrix states that there is a building that takes an area of $(H*W)$. In each index of the matrix, it states the number of blocks/bricks that are placed. We need to find the minimum area from the four sides of the building (left, right, forward, backward).

Number of Accepted Solutions: Senior-0.

Problem I: **K'th Element (II)**

Setter: Quwsar Ohi

Alternate Solution Writer: Quwsar Ohi

Solution: There are three ways to solve the problem.

1. Segment Tree
2. Trie
3. Policy Based DataStructure

Among the three types, the third one (Policy Based DataStructure) is easy to implement. In C++ there are some hidden features that is named as Policy Based DataStructure. In this data structure, we can insert, erase any values. Also, it is possible to find K 'th element from the data structure easily. To solve the problem the contestants are requested to learn this data structure from the following link :

1. [C++ STL: Policy based data structures](#)
2. [C++ STL: Policy based data structures. Part 2](#)

Also, to solve each query, you will need a sliding query optimization. The problem says that the queries are sorted. For each query, you will need to first insert the values in the data structure and find the answer. Then, for the next query, the previous values are to be erased from the data structure and the new value is to be added in the data structure.

Number of Accepted Solutions: Senior-0.