

Problem E - Binary Removals

- **Question Analysis:**

আমাকে একটা **Binary String** দেওয়া আছে। আর ২ টা **integer X and K** দেওয়া আছে।
আমার কাজ হলো - **String** টা কে **remove** করা।

Operation -

- একটা **operation** - এ , একটা **subsequence choose** করা। যাতে **number of Inversion** , $[0, X]$ এই **range** এর মধ্যে থাকে এবং **K** দিয়ে **Divisible** হয়।
- **Inversion** - কোনো **index i,j** যেখানে $i < j$, and $a_i > a_j$ । এখানে যেহেতু শুধু 0, 1 আছে, তাই $a_i = 1, a_j = 0$ - হলেই এটা একটা **inversion**

Minimum কয়টা **operations** লাগবে , যাতে **String** - টা **remove** হয়।

- **Observation:**

1. মনে করি, আমার **inversion count = cnt**.

2. এখন এখানে 2 টা **Case**:

Case-1 : যদি $cnt \% K == 0$ and $cnt \leq X$ হয়ে যায়, তাহলে পুরো **array** টা **select** করব, so **answer = 1**.

Case-2 : যখন আমার শর্ত পূরণ হবে না।

- আমাকে কিন্তু **subsequence** এ 0 & 1 ২টাই থাকতে হবে বলে নাই।
- আমি 1 সব একসাথে , 0 সব একসাথে নিয়ে **subsequence creat** করতে পারি, তাহলে **answer = 1 + 1 = 2**

3. তারমানে , আমার কিন্তু কখনো **answer 1 or 2** ছাড়া অন্যকিছু হওয়ার সুযোগ নেই।

- **Implementation:**

Observation টা **implement** করা।

- **Code:**

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

#define ll long long

int main(){
    int tc=1;
    cin >> tc;
    while(tc--){
        int n,x,k;
        cin >> n >> x >> k;
        string s;
        cin >> s;

        ll cnt = 0, one=0;
        for(auto c: s){
            if(c=='1') one++;
            else cnt += one;
        }

        if(cnt%k==0 && cnt<=x) cout << 1 << endl;
        else cout << 2 << endl;
    }

    return 0;
}
/* Author: Hridoy Barua (CS Instructor Phitron) */
```

Problem F - Lucky Number Game

- **Question Analysis:**

Alice & Bob একটা Game খেলবে।

- একটা array **A** দেওয়া আছে।
- দুজনের কাছেই ২ টা **Lucky Number** আছে। Alice -> b, Bob -> a
- ২জনই - array থেকে **element remove** করবে, যাতে **element** টি, তার **lucky number** এর **multiple**

যে কোনো **move** দিতে পারবে না, সে হেরে যাবে।

**** Bob** আগে **Start** করবে।

- **Observation:**

1. যদি কেউই একটা **number** ও **remove** করতে না পারে, তাহলে **Alice** জিতবে কারণ **Bob** কিছুই **remove** করতে পারবে না।
 2. যদি শুধুমাত্র এমন সংখ্যা থাকে যা তাদের উভয়েই **remove** করতে পারে, তাহলে **Bob** জিতবে কারণ **Bob** সবকিছু **remove** করতে পারবে।
 3. যদি এমন কোনো সংখ্যা না থাকে যা তাদের উভয়েই **remove** করতে পারে, তাহলে **Bob** জিতবে যদি $cnt_bob > cnt_alice$, নাহলে **Alice** জিতবে
 4. যদি এমন কোনো সংখ্যা না থাকে যা তাদের উভয়েই **remove** করতে পারে, তাহলে **Bob** জিতবে যদি $cnt_bob \geq cnt_alice$, নাহলে **Alice** জিতবে
-

- **Implementation:**

Observation টা **implement** করা।

- **Code:**

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

#define ll long long

int main(){
    int tc=1;
    cin >> tc;
    while(tc--){
        int n,a,b;
        cin >> n >> a >> b;

        int all=0, alice=0, bob=0;
        for(int i=0; i<n; i++){
            int x;
            cin >> x;
            if(x%a==0 && x%b==0) all++;
            else if(x%a==0) bob++;
            else if(x%b==0) alice++;
        }

        if(alice==0 && bob==0 && all==0) cout << "ALICE\n";
        else if(alice==0 && bob==0 && all!=0) cout << "BOB\n";
        else if(all==0){
            if(bob>alice) cout << "BOB\n";
            else cout << "ALICE\n";
        }
        else{
            if(bob>=alice) cout << "BOB\n";
            else cout << "ALICE\n";
        }
    }

    return 0;
}

/* Author: Hridoy Barua (CS Instructor Phitron) */
```