NAVPRAYAS
(A GROUP OF INNOVATIVE THOUGHTS)

Code Mantra

-Debug Your soul

Editorial

1.Muskan is a good tutor

Problem Statement is in given link provided:
https://www.hackerrank.com/contests/code-mantra-debug-your-soul-1622011685/challenges/muskan-is-a-good-tutor

Explanation:

In this two observations can be observed,

a)
In AND operation,
0 & 0 gives 0
1 & 1 gives 1
Therefore, if both bits are the same, then it gives the same bit.

b)
In XOR operation,
0 & 1 gives 1
1 & 0 gives 1
Therefore, if both bits are different, then it gives 1 that means maximum bit.

Suppose X_i = 12 which is 1 1 0 0
And Z_i = 10 which is 1 0 1 0

1 1 0 0 - X_i
1 0 1 0 - Z_i
0 0 1 0 - P
3 2 1 0 - bit positions

To find P, two things we have to keep in mind, the bit remains same on taking AND and on taking XOR resultant bit should be maximum simultaneously.

At 0th position, only 0 is option and 1 ^ 0 gives 1, so at 0th bit in P should be 0.
At 1st position, 1 or 0 are options but 0 ^ 1 gives 1, so at 1th bit in P should be 1.
At 2nd position, only 0 is option and 1 ^ 0 gives 1, so at 2nd bit in P should be 0.
At 3rd position, 1 or 0 are options but 0 ^ 1 gives 1, so at 1th bit in P should be 0.

So, we get P = 1 and then we can find X_i ^ P which gives 14
Similarly, we can do for other elements in the X array.
Thus, we can find the total sum of X_i ^ P.

So, it can be concluded that if the bit in Z_i is 1 and the bit in X_i is 0 then at that position, the bit in P should be 1 otherwise 0.

Code in CPP:-

```cpp
#include<bits/stdc++.h>
#define ll long long
using namespace std;


int main(){
    ll n;
    cin>>n;
    ll a[n],b[n];
    ll sum=0;
    for(ll i=0;i<n;i++)cin>>a[i];
    for(ll i=0;i<n;i++)cin>>b[i];
    for(ll i=0;i<n;i++){
        ll p=0;
        for(ll j=0;j<30;j++){
            if(((1<<j)&a[i])==0 && ((1<<j)&b[i])!=0)p+=(1<<j);
        }
        sum+=a[i]^p;
    }
    cout<<sum<<endl;
}
```

Code in Python3 :-

```python
n = int(input())
sum = 0
a = list(map(int, input().split()))
b = list(map(int, input().split()))
for i in range(n):
    p = 0
    for j in range(30):
        if ((1<<j) & a[i])==0 and ((1<<j) & b[i]) != 0:
            p += (1<<j)
    sum += a[i]^p
print(sum)
```

2.Puzzle for chocolates

Problem Statement is in given link provided:
https://www.hackerrank.com/contests/code-mantra-debug-your-soul-1622011685/challenges/puzzles-for-chocolates

Explanation:

Brute Force Approach:

We will check for each value of **Y** iteratively using a for loop. We will calculate the value of S for each value of y and when **S** will become less than X we will get our desired value of **Y.**

Time Complexity: O(P*N*10^{12})

The possible values of y can be from 0 to 10^{12}.

Here the search space is of the range 10^{12}. So, if we find the value of Y using brute force approach it will give TLE. since the for loop will run 10^{16} times.

Optimized Approach: Binary Search

For each possible value of y we have to calculate the value of S.

We will do this by making a function:

```
long long int S(long long int x){
    long long int res = 0;
    for(int i = 1; i <= n; i++)
        res+=max(ar[i] - i * x , (long long int)0);
    return res;
}
```

Now we will make a function for implementing binary search and finding the maximum possible value of y:

```
void BS (){
    long long int  L = 0, R = 1e12;
    long long int s,res;
    while( L <=  R){
        long long int mid =(L+R)/2;
        long long int y =S(mid);
```

```
        if(y<K)   R = mid - 1;
        else
        if(S(mid+1) < K){
            s = y;
            res = mid;
            break;
        }
        else
        L = mid + 1;
    }
    cout<<res<<" "<<s<<endl;
}
```

Time Complexity: O(P*N*log(10^{12}))

The Maximum value of this expression is 100*100*log(10^{12}) , it will always remain less than 10^6 . So, this algorithm will work here i.e in 1 sec.

Final Code:

```
#include<bits/stdc++.h>
#define mod 1000000007
using namespace std;
long long int ar[100001];
int n;
long long int K;

long long int S(long long int x){
    long long int res = 0;
    for(int i = 1; i <= n; i++)
        res += max(ar[i] - i * x , (long long int)0);
```

```
    return res;
}
void BS (){
    long long int  L = 0, R = 1e12;
    long long int s,res;
    while( L <=  R){
        long long int mid = (L+R)/2;
        long long int y = S(mid);
    if(y<K)  R = mid - 1;
    else
    if(S(mid+1) < K){
        s = y;
        res = mid;
        break;
    }
    else
    L = mid+1;
    }
    cout<<res<<" "<<s<<endl;
}
int main()
{
    int t;
    cin>>t;
    while(t--){
        cin>>n>>K;
        for(int i = 1; i <= n; i++) cin>>ar[i];
        BS();
    }
}
```

3.Game between Sisters

Problem Statement is in given link provided:
https://www.hackerrank.com/contests/code-mantra-debug-your-soul-1622011685/challenges/game-between-sisters

Explanation:

Suppose the first element in the array is greater than 1. If decreasing the element to 0 is winning, Khushboo will do that.
Otherwise, Khushboo can decrease that element to 1, forcing Muskan to decrease it, leaving Khushboo in the winning position.
Otherwise, if the first element is 1, then it is forced to decrease it to 0.

So, whichever player gets the first positive integer which is greater than 1 wins. That is, let k be the maximum number such that
a1 = a2 = ... = ak = 1. If k is even, Khushboo will win. Otherwise, Muskan will win. The only exception is when all elements are 1. In that case,
Khushboo wins when k is odd.

Time complexity is O(n).

CPP Solution

```cpp
#include <bits/stdc++.h>
using namespace std;
#define ll long long

int main()
{
    ll t;
    cin >> t;
    while (t--)
    {
        ll n, p = -1, flag = 0;
        cin >> n;
        vector<ll> a(n);
        for (ll i = 0; i < n; i++)
        {
            cin >> a[i];
            if (a[i] > 1 && flag == 0)
            {
                p = i + 1;
                flag = 1;
            }
        }
        if (p == -1)
        {
            if (n % 2 == 0)
            {
                cout << "Muskan\n";
```

```
            }
            else
            {
                cout << "Khushboo\n";
            }
        }
        else
        {
            if (p % 2 == 0)
            {
                cout << "Muskan\n";
            }
            else
            {
                cout << "Khushboo\n";
            }
        }
    }
   return 0;
}
```

Python3 Solution

```
t=int(input())
while(t>0):
   n=int(input())
   p=-1
   x = [int(x) for x in input().split()]
   for i in range(n):
       e=x[i]
       if(e>1):
           p=i+1
           break
   if(p==-1):
       if(n%2==0):
           print("Muskan")
       else:
           print("Khushboo")
   else:
       if(p%2==0):
           print("Muskan")
       else:
           print("Khushboo")
   t-=1
```

4.Jumping Jack 3

Problem Statement is in given link provided:

https://www.hackerrank.com/contests/code-mantra-debug-your-soul-1622011685/challenges/jumping-jack-3-1

Explanation:

Jack has two choices at i^{th} second, either he can jump by 2^{i-1} units or not. So he can jump by a distance of 1,2,4,8,16,...and so on. As we know that every number can be converted into the summation of distinct powers of 2.
Eg: N=11 -> 8 + 2 + 1 = binary_form(1000 + 10 + 1) = binary_form(1011)

If we write the binary form of N and for every set bit in binary representation of N , Jack will have to jump by 2^{i-1} units at the i^{th} second. So now we know that the number of jumps Jack has to take is equal to the number of set bits in the binary form of N.

Problem Setter's code :

Code in CPP

```cpp
#include <bits/stdc++.h>
using namespace std;
int main(){
    int N;
    cin >> N;
    cout << __builtin_popcount(N) << endl;
}
```

Code in Python3

```python
n = int(input())
cnt = 0
while n:
    n = n & (n-1)
    cnt += 1
print(cnt)
```

5.Game of Even and Odd

Problem Statement is in given link provided:

https://www.hackerrank.com/contests/code-mantra-debug-your-soul-1622011685/challenges/even-or-odd-36

Explanation:

If all the numbers are odd, the game ends immediately and results in a Draw (as Atul is not able to pick an even number and start the game). If all the numbers are even, then Atul wins the game as Dev would not be able to pick up any number. Now, since Atul makes the most optimal move, he will choose the maximum among all the even numbers in the list. If all the odd numbers are less than the maximum even number, Atul wins as Dev will not be able to make a move. If the maximum odd number exists which is greater than the maximum even number, Atul will not be able to make the next move, and the score of Dev would be higher than the score of Atul. Hence, Dev wins the game in this case. So, we need to figure out the maximum odd and even numbers respectively.

Examples:

1. **CASE 1:** All numbers are odd.
   List[] = {1, 3, 5, 7, 11}
   Here, Atul is not able to pick any even number and start the game. So the result will be Draw.
2. **CASE 2:** All numbers are even.
   List[] = {2, 4, 6, 8, 10}
   Here,Atul will win as Dev isn't able to pick any odd number.
3. **CASE 3:** For general case.
   List[] = {1, 2, 3, 4, 5}
   We know that both the players play optimally. In the first move, Atul will pick the maximum even number i.e 4 and then in the second move Dev will pick the maximum odd number i.e 5. In the next move Atul will not be able to pick any even number that is greater than 5. So, the game stops here and Dev will win the game.

Problem Setter's code :

Code in CPP

```cpp
#include <iostream>
using namespace std;

int main()
{
    int n;
    cin >> n;
    int number, current_even_max = 0, current_odd_max = 0;
    for (int i = 0; i < n; i++)
    {
        cin >> number;
        if (number % 2 == 0)
```

```
        {
            if (current_even_max < number)
                current_even_max = number;
        }
        if (number % 2 == 1)
        {
            if (current_odd_max < number)
                current_odd_max = number;
        }
    }
    if (current_even_max == 0)
        cout << "Draw" << endl;
    else if (current_even_max > current_odd_max)
        cout << "Atul" << endl;
    else if (current_odd_max > current_even_max)
        cout << "Dev" << endl;
    return 0;
}
```

Code in Python3

```
n = int(input())
max_even,max_odd = 0,0
list_number = list(map(int, input().split()))
for data in list_number:
    if(data&1 and max_odd<data):
        max_odd = data
    elif(max_even < data):
        max_even = data

if max_even == 0:
    print("Draw")
elif(max_even < max_odd):
    print("Dev")
else:
    print("Atul")
```

Thank You!