

## АНАЛИЗ НА РЕШЕНИЕТО НА ЗАДАЧА МАРИ

Задачата е директно приложение на играта на Ним и търсенето на печеливш ход. Нека при дадено викане на операцията от вида В броя на елементите в текущото множество да бъде  $N$ , а самите те да са означени с  $a[0], a[1] \dots a[N - 1]$ . Нека  $xr = a[0] \wedge a[1] \wedge \dots \wedge a[N - 1]$ .

Както знаем, ако  $xr$  е равно на нула, то играча, който е пръв на ход, губи, т.е. извеждаме „-1“. Ако  $xr$  обаче не е 0, търсим числото с най-малък индекс, което да заместим с  $X$ .

Разглеждаме уравнението  $xr = a[0] \wedge a[1] \wedge \dots \wedge a[N - 1]$ . Ако от двете страни добавим  $\wedge xr$ , то от ляво ще се получи 0, а от дясно  $a[0] \wedge a[1] \wedge \dots \wedge a[i] \wedge xr \wedge \dots \wedge a[N - 1]$ . Т.е. ако някое число  $a[i]$  заменим с  $a[i] \wedge xr$ , ние ще получим нова редица, в която този, който е на ход, ще загуби. Следователно  $X = a[i] \wedge xr$ . Единственото условие е това  $X = a[i] \wedge xr$  да е по-малко от  $a[i]$ . И така с линейна сложност можем да намерим минималното  $i$ , за което то да бъде изпълнено, след което да изведем на екрана  $i$  и  $a[i] \wedge xr$ .

### Кодът на програмата:

```
#include <iostream>
#include <stdio.h>
#include <stdlib.h>

using namespace std;

int Q;
int N;
int a[1000];

void solveNim () {
    int xr = 0;
    for (int i = 0; i < N; i++)
        xr ^= a[i];

    if (xr == 0) {
        printf ("-1\n");
        return;
    }

    for (int i = 0; i < N; i++)
        if ((a[i] ^ xr) < a[i]) {
            printf ("%d %d\n", i, a[i] ^ xr);
            return;
        }
}

int main () {
    scanf ("%d\n", &Q);
    char op;
    while (Q--) {
        scanf ("%c", &op);
        if (op == 'A') {
```

```
    scanf ("%d\n", &a[N++]);
} else {
    scanf ("%c", &op);
    solveNim();
}
}
```