

题目 1: [Problem - 1838D - Codeforces](#)

给出一个偶数长度的括号序列，每次查询反转一个括号，判断括号序列是否有如下形式 $() () ((.....)) () () ()$

解：对于括号进行 01 染色，0 位置为左括号或 1 位置为右括号记为合法，维护一个不合法的集合，当第一个不合法的位置是偶数且最后一个不合法的位置为奇数时括号具有题目形式

Code:

```
#include<bits/stdc++.h>
using namespace std;
const int N = 2e5 + 10;
int n, q;
char ch[N];
set<int> st;

void solve()
{
    int x; scanf("%d", &x);
    if (st.count(x)) st.erase(x);
    else st.insert(x);

    if (n % 2) puts("NO");
    else if (st.size() && (*st.begin() % 2 == 1 || *st.rbegin() % 2 == 0)) puts("NO");
    else puts("YES");
}

int main()
{
    scanf("%d%d", &n, &q);
    scanf("%s", ch+1);
    for (int i=1; i<=n; i++)
    {
        if ((i % 2 == 1 && ch[i] == ')') || (i % 2 == 0 && ch[i] == '('))
        {
            st.insert(i);
        }
    }
    while (q--)
    {
        solve();
    }
}
```

题目 2: [Problem - K - Codeforces](#)

在一个长度为 n 的线段上有若干萤火虫，每个时刻每个萤火虫会向左或向右分裂如：010->101,但是两个萤火虫如果相遇就会抵消，超出边

界就会死亡如：101->000, 0110->0000,构造一个长度为 n 的序列，使得无论经过多久线段上都有萤火虫

解： n 为 1, 3 无解输出-1

n 为奇数则为 10001 + 10 + 01 + 10 + 01 这种形式

n 为偶数数则为 1001 + 10 + 01 + 10 + 01 这种形式

Code:

```

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

int main()
{
    int n;
    cin>>n;
    if (n == 2)
    {
        puts("10");
    }
    else if (n == 3)
    {
        puts("Unlucky");
    }
    else if (n % 2 == 0)
    {
        n -= 4;
        printf("1001");
        int flag = 1;
        for (int i=1;i<=n,i+=2)
        {
            if (flag)
            {
                printf("10");
            }
            else
            {
                printf("01");
            }
            flag ^= 1;
        }
    }
    else
    {
        n -= 5;
        printf("10001");
        int flag = 1;
        for (int i=1;i<=n,i+=2)
        {
            if (flag)
            {
                printf("10");
            }
            else
            {
                printf("01");
            }
            flag ^= 1;
        }
    }
}

```

题目 3: <https://ac.nowcoder.com/acm/contest/33194/E>

给定 $n \leq 1e9$, 要求构造一个长度为 $m \leq 100$ 的排列, 其中最长递增子段个数为 n

解：考虑先构造 $n=2^k$ 的答案，我们可以这么放：

21 43 65 87.....

那么对于 $2^k + 2^{(k-a1)} + 2^{(k-a2)}.....$ 的答案，我们可以考虑在其中增加一些数：

考虑我们在序列中可以插入的空位：

21 43 65 87

空位 0 空位 1 空位 2 空位 3

在没放入数时，最长递增子段长度为 4

可以观察到，若在空位 0 放入大于 8 的长度为 4 连续递增字段
对答案的贡献为 2^0

可以观察到，若在空位 0 放入大于 8 的长度为 3 连续递增字段
对答案的贡献为 2^1

可以观察到，若在空位 0 放入大于 8 的长度为 2 连续递增字段
对答案的贡献为 2^2

可以观察到，若在空位 0 放入大于 8 的长度为 1 连续递增字段
对答案的贡献为 2^3

但是注意放数的时候最长递增子段长度不能超过 4

当 n 形式为 $2^4 + 2^3 + 2^2$ 时，应该这么放：

21 43 9 65 10 87

空位 0 空位 1 空位 2 空位 3

即当后面的更高位有 1 时，应把自己放入空位中的数进行后移

Code(代码实现较为复杂):

```

#include<bits/stdc++.h>
using namespace std;
int n;
int a[150];

void solve()
{
    memset(a,0,sizeof(a));
    cin>>n;
    if (n == 1)
    {
        cout<<1<<endl<<1<<endl;
        return;
    }
    int len = 0,h = n;
    while (h)
    {
        len++;
        h /= 2;
    }len--;
    for (int i=len;i>=1;i--)
    {
        if ((n & (1 << (i - 1))) != 0)
        {
            for (int j=len;j>=i;j--)
            {
                if (a[j] == 0)
                {
                    a[j] = i;
                }
            }
        }
    }
    int ans = 2 * len + len;
    int cnt = 0;
    for (int i=1;i<=len;i++)
    {
        if (a[i] == 0) cnt++;
    }
    ans -= cnt;
    // for (int i=1;i<=len;i++) cout<<a[i]<<" ";cout<<endl;
    // cout<<"check: "<<cnt<<endl;
    cout<<ans<<endl;
    for (int i=1;i<=len;i++)
    {
        for (int j=1;j<=len;j++)
        {
            if (a[j] == i) cout<<j-cnt+2*len<<" ";
        }
        cout<<i*2<<" "<<i*2-1<<" ";
    }
    cout<<endl;
}

int main()
{
    int t1;
    cin>>t1;
    while (t1-->0)
    {
        solve();
    }
}

```

题目 4: [Problem - I - Codeforces](#)

交互题，要求你猜一个环的长度 $n \leq 1e9$ ，环上是 n 的排列，有一个人站在环上，每次你可以让他走 x 步，他会告诉你他脚底下的数，可以问 $q \leq 1e4$ 次

1. 先漫无目的的走 3333 次，记录脚下数的最大值 t , $E(t) = n * k / (k+1)$
2. 然后走 3333 次 1 步
3. 走 1 次 t 步
4. 走 3333 次 3333 步

```

#include<bits/stdc++.h>
using namespace std;
const int K=3333;
int random()
{
    return (int)(1.0*rand()/RAND_MAX*1e9);
}
map<int,int> ma;
int main()
{
    srand(time(0));
    int m=0,x;
    for(int i=1;i<=K;i++)
    {
        printf("walk %d\n",random());
        fflush(stdout);
        scanf("%d",&x);
        if(x>m) m=x;
    }
    for(int i=1;i<=K;i++)
    {
        printf("walk %d\n",1);
        fflush(stdout);
        scanf("%d",&x);
        if(ma[x])
        {
            printf("guess %d",i-1);
            return 0;
        }
        else ma[x]=i;
    }
    printf("walk %d\n",m);
    fflush(stdout);
    scanf("%d",&x);
    if(ma[x])
    {
        printf("guess %d",K+m-ma[x]);
        return 0;
    }
    for(int i=1;i<=K;i++)
    {
        printf("walk %d\n",K);
        fflush(stdout);
        scanf("%d",&x);
        if(ma[x])
        {
            printf("guess %d", (i+1)*K+m-ma[x]);
            return 0;
        }
    }
    return 0;
}

```