

题目描述

现在给你一个神奇的数组，这个数组的长度为 n ，这个数组需要满足两个条件

1. 数组的元素之和等于零，即 $a_1 + a_2 + \dots + a_n = 0$ 。

2. 数组的元素之积等于 n ，即 $a_1 * a_2 * \dots * a_n = n$ 。

现在给出这个数组的长度 n ，求这个数组。

题目输入

数据有多组输入，共有 $t+1$ 行。

第一行输入测试样例组数 t 。

第二到 $t+1$ 行分别为每次输入数组的长度 n 。

题目输出

一共输出 t 行，若输入的数组长度 n 有解，则输出这个数列，若误解则输出“NO”。

题解代码

```
#include <bits/stdc++.h>
#define int long long
#define endl "\n"
const int N = 2e5 + 10;
const int mod = 1e9 + 7;
const int INF = 0x3f3f3f3f3f3f3f3f;
inline void solve()
{
    int n;
    std::cin >> n;
    if (n % 4 == 0)
    {
```

```

        int res = n / 4;
        if (res % 2 == 1)
        {
            std::cout << 2 << " " << -2 * res;
            for (int i = 0; i < 3 * res - 2; i++) std::cout << " "
<< 1;
            for (int i = 0; i < res; i++) std::cout << " " << -1;
        }
        else
        {
            std::cout << -2 << " " << -2 * res;
            for (int i = 0; i < 3 * res; i++) std::cout << " " <<
1;
            for (int i = 0; i < res - 2; i++) std::cout << " " <<
-1;
        }
        std::cout << endl;
    }
    else
    {
        std::cout << "NO" << endl;
    }
}
signed main()
{
    std::ios_base::sync_with_stdio(false);
    std::cin.tie(nullptr);
    std::cout.tie(nullptr);
    int t = 1;
    std::cin >> t;
    while (t--) solve();
    return 0;
}

```

题解

解的情况

若数组n只有全为奇数，那么就不能满足数组之和为0；

H2

0；

若数组n只有一个偶数，等价于n-1个奇数+1个偶数，还是不能满足数组之和为

若数组n有两个偶数，等价于n-2个奇数+2个偶数，显然是可以满足的。

又因数组之积为n，那么n为4的倍数时有解。

构造数列

设 $n=4*k$

H2

那么当k为奇数时，数列构造为：

$$n = 2 \times (-2k) \times 1^{3k-2} \times (-1)^k$$

当k为偶数时，数列构造为：

$$n = (-2) \times (-2k) \times 1^{3k} \times (-1)^{k-2}$$