Solution: Odd or Even

Approach:
We can solve this problem using either the modulo operator or bitwise operations.

1. **Modulo Method:** Check $n \bmod 2$:
    - If $n \bmod 2 = 0$, then n is even.
    - Otherwise, n is odd.
2. **Bitwise Method:** Observe that the least significant bit (LSB) of n determines odd/even:
    - If $(n \,\&\, 1ll) = 0$, then n is even.
    - If $(n \,\&\, 1ll) = 1$, then n is odd.

Python Implementation:

```python
t = int(input())
for _ in range(t):
    n = int(input())
    if n % 2 == 0:
        print("EVEN")
    else:
        print("ODD")
```

C++ Implementation (bitwise):

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

int main() {
    int t;
    cin >> t;
    while (t--) {
        long long n;
        cin >> n;
        if (n & 1ll) cout << "ODD\n";
        else cout << "EVEN\n";
    }
    return 0;
}
```

C Implementation (bitwise):

```
#include <stdio.h>

int main() {
    int t;
    scanf("%d", &t);
    while (t--) {
        long long n;
        scanf("%lld", &n);
        if (n & 1ll) printf("ODD\n");
        else printf("EVEN\n");
    }
    return 0;
}
```