Get Odd

Problem statement :

Given an integer N, determine the number of pairs (A, B) such that:

- $1 \leq A,B \leq N$;
- A+B is odd.

Input Format

- The first line of input will contain a single integer T, denoting the number of test cases.
- Each test case consists of a single integer N.

Constraints

- $\mathbf{1 \leq T \leq 100}$
- $\mathbf{1 \leq N \leq 10^9}$

Output Format

- For each test case, output the number of required pairs.

Sample Input 0

```
5

2

3

1

300

201
```

Sample Output 0

```
2

4
```

```
0

45000

20200
```

Explanation 0

- Test Case-1: Possibilities are (1,2) and (2,1)
- Test Case-2: Possibilities are (1,2),(2,3),(2,1) and (3,2)

JAVA

```
import java.io.*;
import java.util.*;
import java.text.*;
import java.math.*;
import java.util.regex.*;

public class Solution {


    public static void main(String[] args) {
        Scanner s=new Scanner(System.in);
        int t=s.nextInt();
        while(t>0)
        {
            long n=s.nextLong();
            long even=n/2,odd=n-even;
            System.out.println(2*even*odd);
            t--;


        }
    }
}
```

C

```
#include <stdio.h>
int main(void) {
	int t;
	scanf("%d",&t);
	while(t--)
	{
	   long long int n;
	   scanf("%lld",&n);
	   long long int even,odd;
	   even=n/2;
	   odd=n-even;
	   printf("%lld\n",even*odd*2);
	}
	return 0;
}
```

C++

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

int main(void) {
	int t;
```

```
    cin>>t;
    while(t--)
    {
       long long int n,even,odd;
       cin>>n;
       even=n/2;
       odd=n-even;
       cout<<even*odd*2<<endl;
    }
    return 0;
}
```