

## Talent Battle 100 Days Coding Series

Kulyash is given an integer  $N$ . His task is to break  $N$  into some number of (integer) powers of 2.

To achieve this, he can perform the following operation several times (possibly, zero):

- Choose an integer  $X$  which he already has, and break  $X$  into 2 integer parts ( $Y$  and  $Z$ ) such that  $X=Y+Z$ .

Find the minimum number of operations required by Kulyash to accomplish his task.

### 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 line of input.
  - The first and only line of each test case contains an integer  $N$  — the original integer with Kulyash.

### Output Format

For each test case, output on a new line the minimum number of operations required by Kulyash to break  $N$  into powers of 2.

### Sample Input

2  
3  
4

### Sample Output

1  
0

C

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

```
#include <math.h>
```

```
int main(void) {
```

```
    int t, ans, temp, temp2, x;
```

```
    long long int n;
```

## Talent Battle 100 Days Coding Series

```
scanf("%d",&t);
while(t--)
{
    ans=0;
    scanf("%lld",&n);
    while(1)
    {
        temp=floor(log2(n));
        x=floor(pow(2,temp));
        temp2=n%x;
        ans++;
        n=temp2;
        if(n==0)
            break;
    }
    printf("%d\n",ans-1);
}
return 0;
}
```

### C++

```
#include <iostream>
#include <bits/stdc++.h>
using namespace std;
typedef long long int ll;
bool isPowerOf2(ll n){
    return (ceil(log2(n))==floor(log2(n)));
}
long long binpow(long long a, long long b) {
```

## Talent Battle 100 Days Coding Series

```
long long res = 1;
while (b > 0) {
    if (b & 1)
        res = res * a;
    a = a * a;
    b >>= 1;
}
return res;
}
```

```
int main() {
    int t; cin >> t;
    while (t--) {
        int n; cin >> n;
        if (isPowerOf2(n) == true) cout << "0\n";
        else {
            int cnt = 0;
            int temp = n;
            while (isPowerOf2(temp) == false) {
                cnt++;
                temp = temp - binpow(2, log2(temp));
            }
            cout << cnt << "\n";
        }
    }
    return 0;
}
```

## Talent Battle 100 Days Coding Series

### Java

```
import java.util.*;
import java.lang.*;
import java.io.*;

class Main
{
    static boolean isPowerOfTwo(int x)
    {
        return x != 0 && ((x & (x - 1)) == 0);
    }

    public static void main (String[] args) throws java.lang.Exception
    {
        Scanner sc=new Scanner(System.in);
        int t=sc.nextInt();
        while(t-->0){
            int n=sc.nextInt();
            int count = 0;
            while (n>0) {
                count += n & 1;
                n >>= 1;
            }
            System.out.println(count-1);
        }
    }
}
```

### Python

```
def solve(num):
    if num==1:
```

## Talent Battle 100 Days Coding Series

```
    return 0
else:
    return num%2+solve(num//2)
n=int(input())
for i in range(n):
    num=int(input())
    print(solve(num))
```



TalentBattle