**Minimum Operations: -**

**Easy Accuracy: 60.02% Submissions: 85K+ Points: 2**

Given a number N. Find the minimum number of operations required to reach **N** starting from **0**. You have 2 operations available:

* Double the number
* Add one to the number

**Example 1:**

**Input:**

N = 8

**Output:** 4

**Explanation**:   
0 + 1 = 1 --> 1 + 1 = 2 --> 2 \* 2 = 4 --> 4 \* 2 = 8.

**Example 2:**

**Input**:

N = 7

**Output:** 5

**Explanation**:   
0 + 1 = 1 --> 1 + 1 = 2 --> 1 + 2 = 3 --> 3 \* 2 = 6 --> 6 + 1 = 7.

**Your Task:**  
You don't need to read input or print anything. Your task is to complete the function **minOperation()**which accepts an integer **N**and return number of minimum operations required to reach N from 0.

**Expected Time Complexity:**O(LogN)  
**Expected Auxiliary Space:**O(1)

**Constraints:**  
1 <= N <= 106

**Code: -**

//{ Driver Code Starts

#include <bits/stdc++.h>

using namespace std;

// } Driver Code Ends

class Solution

{

public:

int minOperation(int n)

{

//code here.

// base case

if(n == 0) return 0;

if(n & 1){

return 1 + minOperation(n-1);

}

else{

return 1 + minOperation(n / 2);

}

}

};

//{ Driver Code Starts.

int main()

{

int t;

cin>>t;

while(t--)

{

int n;

cin>>n;

Solution ob;

cout<<ob.minOperation(n)<<endl;

}

}

// } Driver Code Ends

**T.C: - O(log N)**

**S.C: - O(1) without recursive call stack**