**Number is sparse or not**

[bit](http://www.practice.geeksforgeeks.org/tag-page.php?tag=bit&isCmp=0)

Given a number, check whether it is sparse or not. A number is said to be a sparse number if in binary representation of the number no two or more consecutive bits are set.

**Input:**  
The first line of input contains an integer T denoting the number of test cases. The first line of each test case is number 'N'.  
  
**Output:**  
Print '1' if the number is sparse and '0' if the number is not sparse.  
  
**Constraints:**  
1 <=T<= 100  
1 <=n<= 100  
  
**Example:**  
**Input:**  
4  
72  
12  
2  
3

**Output:**  
1  
0  
1  
0

\*\*For More Examples Use Expected Output\*\*

<http://www.practice.geeksforgeeks.org/problem-page.php?pid=501>

#include <iostream>

#include <stdio.h>

#include <math.h>

#include <vector>

using namespace std;

int main() {

    int T;

    scanf("%d", &T);

    while(T--) {

        int n;

        scanf("%d", &n);

        std::vector<int> aux;

        int i =0;

        while (n > 0)

        {

            aux.push\_back(n%2);

            if(i >0 && aux[i] == 1 && aux[i-1] == aux[i] ) {

                printf("0");

                break;

            }

            i++;

            n /= 2;

        }

        if(n == 0) {

            printf("1");

        }

        printf("**\n**");

    }

  system("pause");

 return 0;

}

----VARIANTE MAS SIMPLE PERO MAS LARGA-------

#include <iostream>

#include <stdio.h>

#include <math.h>

#include <vector>

using namespace std;

int main() {

int T;

scanf("%d", &T);

while(T--) {

int n;

scanf("%d", &n);

std::vector<int> binario;

while (n > 0)

{

binario.push\_back(n%2);

n /= 2;

}

std::reverse(binario.begin(), binario.end());

bool unos\_seguidos = false;

for (int i = 1; i < binario.size(); i++) {

if (binario[i] == 1 && binario[i - 1] == binario[i])

{

unos\_seguidos = true;

}

}

if (unos\_seguidos)

{

printf("0");

}

else

{

printf("1");

}

printf("\n");

}

system("pause");

return 0;

}