**Sum of all divisors from 1 to n: -**

**Easy Accuracy: 43.37% Submissions: 53K+ Points: 2**

Given a positive integer **N**., The task is to find the value of , where function ***F(i)*** for the number **i** is defined as the sum of all divisors of **i**.

**Example 1:**

**Input:**

**N = 4**

**Output:**

15

**Explanation:**

F(1) = 1

F(2) = 1 + 2 = 3

F(3) = 1 + 3 = 4

F(4) = 1 + 2 + 4 = 7

ans = F(1) + F(2) + F(3) + F(4)

= 1 + 3 + 4 + 7

= 15

**Example 2:**

**Input:**

N = 5

**Output:**

21

**Explanation:**

F(1) = 1

F(2) = 1 + 2 = 3

F(3) = 1 + 3 = 4

F(4) = 1 + 2 + 4 = 7

F(5) = 1 + 5 = 6

ans = F(1) + F(2) + F(3) + F(4) + F(5)

= 1 + 3 + 4 + 7 + 6

= 21

**Your Task:**  
You don't need to read input or print anything. Your task is to complete the function **sumOfDivisors()** which takes an integer **N** as an input parameter and returns an integer.  
  
**Expected Time Complexity:** O(N)  
**Expected Auxiliary Space:** O(1)

**Constraints:**  
1 <= N <= 106

**Code: -**

//{ Driver Code Starts

#include<bits/stdc++.h>

using namespace std;

// } Driver Code Ends

//User function Template for C++

class Solution

{

public:

long long sumOfDivisors(int N)

{

// Write Your Code here

long long sum = 0;

for(long long i = 1; i <= N; ++i){

long long countappear = floor(N / i);

sum += (countappear \* i);

}

return sum;

}

};

//{ Driver Code Starts.

int main()

{

int t;

cin >> t;

while (t--)

{

int N;

cin>>N;

Solution ob;

long long ans = ob.sumOfDivisors(N);

cout<<ans<<endl;

}

return 0;

}

// } Driver Code Ends

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

**S.C: - O(1)**