**Largest prime factor :-**

Medium Accuracy: 27.25% Submissions: 65K+ Points: 4

Given a number **N**, the task is to find the largest prime factor of that number.  
 **Example 1:**

**Input:**

N = 5

**Output:**

5

**Explanation:**

5 has 1 prime factor i.e 5 only.

**Example 2:**

**Input:**

N = 24

**Output:**

3

**Explanation:**

24 has 2 prime factors 2 and 3 in which 3 is greater.

**Your Task:**  
You don't need to read input or print anything. Your task is to complete the function **largestPrimeFactor()** which takes an integer **N** as input parameters and returns an integer, largest prime factor of N.

**Expected Time Complexity:** O(sqrt(N))  
**Expected Space Complexity:** O(1)

**Constraints:**  
2 <= N <= 109

**Code :-**

//{ Driver Code Starts

#include<bits/stdc++.h>

using namespace std;

// } Driver Code Ends

bool isprime(int num){

return (num==2 || num==3) ? true : (

(num%2==0 || num%3==0 || num<=1) ? false :({

bool ans=true;

for(int i=5; i<=sqrt(num)+1; i++){

if(num%i==0){ ans=false; break; }

}

(ans==true) ? true : false;

})

);

}

class Solution{

public:

long long int largestPrimeFactor(int N){

if(isprime(N)) return N;

int ans=INT\_MIN;

for(int item=2; item<=sqrt(N)+1; item++){

if(N%item==0){

if(isprime(item)){

ans=max(ans, item);

//cout<<item<<",";

}

if(isprime(N/item)){

ans = max(ans, N/item);

//cout<<(N/item)<<",";

}

}

}

return ans;

}

};

//{ Driver Code Starts.

int main()

{

int t;

cin>>t;

while(t--)

{

int N;

cin>>N;

Solution ob;

cout << ob.largestPrimeFactor(N) << endl;

}

return 0;

}

// } Driver Code Ends

**T.C :- O(root N \* root V)**

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