**Find The Cube Free Number**

A cube free number is a number who’s none of the divisor is a cube number (A cube number is a cube of a integer like 8 (2 \* 2 \* 2) , 27 (3 \* 3 \* 3) ). So cube free numbers are 1, 2, 3, 4, 5, 6, 7, 9, 10, 11, 12, 13, 14, 15, 17, 18 etc (we will consider 1 as cube free). 8, 16, 24, 27, 32 etc are not cube free number. So the position of 1 among the cube free numbers is 1, position of 2 is 2, 3 is 3 and position of 10 is 9. Given a positive number you have to say if its a cube free number and if yes then tell its position among cube free numbers.

#include<iostream>

#include<cmath>

using namespace std;

int main(){

int b[100];

for(int i=0;i<=100;i++)

{

b[i]=1;

}

for(int i=2;i<=100;i++)

{

if(b[i]==1){

for(int j=i\*i;j<=100;j=j+i)

{

b[j]=0;

}

}

}

int c[50];

int k=0;

c[k++]=2;

for(int i=3;i<=100;i+=2)

{

if(b[i]){

c[k++]=i;

}

}

int d[1000001];

for(int i=0;i<=1000000;i++)

{

d[i]=1;

}

for(int i=0;i<k;i++){

int p=c[i]\*c[i]\*c[i];

for(int j=p;j<=1000000;j=j+p)

{

d[j]=-1;

}

}

int sum=0;

for(int i=1;i<=1000000;i++){

if(d[i]!=-1){

sum=sum+1;

d[i]=sum;

}

}

int count=1;

int t;

cin>>t;

while(t--)

{

int n;

cin>>n;

if(d[n]==-1)

{

cout<<"Case"<<" "<<count<<":"<<" "<<"Not Cube Free"<<endl;

}

else{

cout<<"Case"<<" "<<count<<":"<<" "<<d[n]<<endl;

}

count++;

}

return 0;

}