

L17 : Number Theory 3

1-Tut : Totient Function

[Send Feedback](#)

You are given an integer N and are supposed to find the value of Euler toient function for N $\phi(N)$

Input Format:

First line of input will contain T(number of test case), each test case follows as.

An integer N in new line.

Output Format:

For each test case print the answer in new line

Constraints:

$1 \leq T \leq 100$

$1 \leq N \leq 10^9$

Sample Input:

2
20
21

Sample Output:

8
12

```
1. #include<bits/stdc++.h>
2. using namespace std;
3.
4. int main(){
5.
6.     int t;
7.     cin>>t;
8.     while(t--){
9.         int n;
10.        cin>>n;
11.        int temp=n;
12.        int ans=n;
13.        for( int i=2 ; i*i<=n ; i++)
14.        {
15.            if((temp%i)==0)
16.            {
17.                while((temp%i)==0)
18.                    temp/=i;
19.
20.                ans-=ans/i;
21.            }
```

```

22.     }
23.     if(temp>1)
24.     {
25.         ans-=ans/temp;
26.     }
27.     cout<<ans<<endl;
28.
29. }
30. }

```

2-Tut : Sum of LCM

[Send Feedback](#)

Given n, calculate and print the sum :

$LCM(1,n) + LCM(2,n) + \dots + LCM(n,n)$

where $LCM(i,n)$ denotes the Least Common Multiple of the integers i and n.

Input Format :

First line of input will contain T(number of test case), each test case follows as.

An Integer N

Output Format :

Required sum for each test case in newline.

Constraints :

$1 \leq T \leq 10^4$

$1 \leq n \leq 10^5$

Sample Input 1 :

1
5

Sample Output 1 :

55

Sample Input 2 :

1
2

Sample Output 2 :

4

```

1. #include<bits/stdc++.h>
2. using namespace std;
3. unsigned long long int phi(unsigned long long int n)
4. {
5.     unsigned long long int result = n;
6.     for (unsigned long long int i = 2; i * i <= n; i++)
7.     {
8.         if (n % i == 0)

```

```

9.         {
10.         while (n % i == 0)
11.             n /= i;
12.
13.             result = (result*(i-1))/i;
14.         }
15.     }
16.     if (n > 1)
17.         result = (result*(n-1))/n;
18.     return result;
19. }
20.
21. int main(){
22.
23.
24.     int t;
25.     cin>>t;
26.     while(t-->0)
27.     {
28.         long long n;
29.         cin>>n;
30.         vector<long long> divisors;
31.         for (long long i=2; i<=sqrt(n); i++)
32.         {
33.             if (n%i == 0)
34.             {
35.
36.                 if (n/i == i)
37.                 {
38.                     divisors.push_back(i);
39.                 }
40.                 else
41.                 {
42.                     divisors.push_back(i);
43.                     divisors.push_back(n/i);
44.                 }
45.             }
46.         }
47.         unsigned long long int sum=0;
48.         for(int i=0;i<divisors.size();i++)
49.         {
50.             sum=(sum+(phi(n/divisors[i])*((n*n)/divisors[i])));
51.         }
52.         sum=(sum+(phi(n)*(n*n)));

```

```

53.     sum=(sum+2*n);
54.     sum=sum/2;
55.     cout<<sum<<endl;
56.
57. }
58. return 0;
59. }

```

3-Tut : Segmented Sieve Problem

[Send Feedback](#)

In this problem you have to print all primes from given interval.

Input Format:

First line of input will contain T(number of test cases), each test case follows as.

On each line are written two integers L and U separated by a blank. L - lower bound of interval, U - upper bound of interval.

Output Format:

For each test case output must contain all primes from interval [L; U] in increasing order.

Constraints:

$1 \leq T \leq 100$

$1 \leq L \leq U \leq 10^9$

$0 \leq U - L \leq 10^5$

Sample Input:

```

2
2 10
3 7

```

Sample Output:

```

2 3 5 7
3 5 7

```

```

1. #include<bits/stdc++.h>
2. using namespace std;
3. int main(){
4.     long long n=1000000000;
5.     long long size=sqrt(n);
6.     bool arr[size];
7.     //cout<<size<<endl;
8.     arr[0]=false;
9.     arr[1]=false;
10.    for(long long i=2;i<=size;i++)
11.    {
12.        arr[i]=true;
13.    }
14.    for(long long i=2;i<=size;i++)

```

```

15.  {
16.    for(long long j=i*i;j<=size;j+=i)
17.    {
18.        arr[j]=false;
19.    }
20. }
21. int t;
22. cin>>t;
23. while(t--)
24. {
25.     long long l,u;
26.     cin>>l>>u;
27.     bool isPrime[u-l+1];
28.     for(int i=0;i<=u-l;i++)
29.     {
30.         isPrime[i]=true;
31.     }
32.     if(u>size)
33.     {
34.         for(long long i=2;i<=size;i++)
35.         {
36.             if(i*i>u)
37.             {
38.                 break;
39.             }
40.             if(arr[i]==true)
41.             {
42.                 long long base=(l/i)*i;
43.                 if(base<l)
44.                 {
45.                     base=base+i;
46.                 }
47.                 for(long long j=base;j<=u;j+=i)
48.                 {
49.                     isPrime[j-l]=false;
50.                 }
51.                 if(base==i)
52.                 {
53.                     isPrime[base-l]=true;
54.                 }
55.             }
56.         }
57.         for(long long i=0;i<=(u-l);i++)
58.         {

```

```

59.         if(isPrime[i]==true)
60.         {
61.             cout<<i+l<<" ";
62.         }
63.     }
64. }
65. else
66. {
67.     for(long long i=l;i<=u;i++)
68.     {
69.         if(arr[i]==true)
70.         {
71.             cout<<i<<" ";
72.         }
73.     }
74. }
75. cout<<endl;
76. }
77. return 0;
78. }

```

4-Ass : Ninja Factor

[Send Feedback](#)

Ninja is given an array of integers of size N and Q queries, each query will be consists of two integers l, r and ninja is supposed to calculate the number of i such that the ninja factor of array[i] is a prime number where $l \leq i \leq r$

Ninja factor of a Number A is defined as the number of integers B such that

$$1 \leq B \leq A$$

$$\text{LCM}(A, B) = A * B$$

Input Format:

Line1: contain two space-separated integers N and Q denoting the number of elements in array and number of queries.

Line2: contain N space-separated integers denoting the elements of the array

Next, Q lines contain two space-separated integers l, r describing the query.

Output Format:

For each query print the answer in a newline.

Constraints:

$$1 \leq N, Q \leq 10^5$$

$$1 \leq \text{arr}[i] \leq 10^9$$

$$1 \leq l, r \leq N$$

Sample Input:

```

10 6
8 8 6 8 6 7 10 7 9 9
2 10

```

7 7
5 7
7 8
6 10
3 4

Sample Output:

2
0
1
0
0
1

Explanation:

6 is the only number whose ninja factor is prime here(i.e. 2) {[LCM(1,6)=1*6; LCM(5, 6)=5*6]}

```
1. #include<bits/stdc++.h>
2. using namespace std;
3. typedef long long int ll;
4. int main()
5. {
6.     ll n,q;
7.     cin>>n>>q;
8.     vector<ll> a(n+1), dp(n+1, 0);
9.
10.    for(ll i=1;i<=n;++i)
11.    {
12.        cin>>a[i];
13.        if(a[i] == 3 || a[i]==4 || a[i]==6)
14.            dp[i] = 1;
15.
16.        dp[i] += dp[i-1];
17.    }
18.
19.    while(q--)
20.    {
21.        ll l,r;
22.        cin>>l>>r;
23.        cout<<dp[r] - dp[l-1]<<'\n';
24.    }
25.
26.    return 0;
27. }
```