# 6.3欧拉函数

**规定 phi为φ**

**6.3.1概述**

欧拉函数就是对于一个正整数n，小于n且和n互质的正整数（包括1）的个数，记作φ(n) 。

欧拉函数的通式：φ(n)=n\*(1-1/p1)(1-1/p2)(1-1/p3)\*(1-1/p4)……(1-1/pn)

其中p1, p2……pn为n的所有质因数，n是不为0的整数。φ(1)=1（唯一和1互质的数就是1本身）。

**6.3.2 性质**

① **当m,n互质时，有phi（m\*n）= phi（m）\*phi（n）；**

② **若i%p==0，有phi（i\*p） = p \* phi(i)；**

③ 对于互质x与p，有x^phi§≡1（mod p),因此x的逆元为x^(phi§-1)，**即欧拉定理**。

（特别地，当p为质数时，phi（p）=p-1,此时逆元为x^(p-2)，即费马小定理）

④ 当n为奇数时，phi(2n)=phi(n)

⑤ 若x与p互质，则p-x也与p互质，因此小于p且与p互质的数之和为phi(x)\*x/2;

⑥N>1，不大于N且和N互素的所有正整数的和是 1/2 \*N \*eular(N)。

**6.3.3 欧拉筛模板**

1. **void** euler(**int** n)
2. {
3. vis[1]=**true**;
4. **int** cnt=0;
5. **for**(**int** i=2;i<=n;i++)
6. {
7. **if**(!vis[i]) prime[++cnt]=i;
8. **for**(**int** j=1;prime[j]\*i<=n;j++)
9. {
10. vis[i\*prime[j]]=**true**;
11. **if**(i%prime[j]==0) **break**;
12. }
13. }
14. }

**6.3.4 欧拉函数模板**

1. **int** euler(**int** x)
2. {
3. **int** ans=x;
4. **for**(**int** i=2;i\*i<=x;i++)//模拟（1-1/p1）\*（1-1/p2）……过程，pi为x的质因数
5. {
6. **if**(x%i==0)
7. {
8. ans=ans-ans/i;
9. }
10. **while**(x%i==0)
11. {
12. x/=i;
13. }
14. }
15. **if**(x>1) ans=ans-ans/x;//若x不为1，说明仍有一个质因数没有乘
16. **return** ans;
17. }

**6.3.5 线性求欧拉函数模板**

1. **void** euler(**int** n)
2. {
3. vis[1]=**true**;
4. phi[1]=1;
5. **int** cnt=0;
6. **for**(**int** i=2;i<=n;i++)
7. {
8. **if**(!vis[i]) prime[++cnt]=i,phi[i]=i-1;
9. **for**(**int** j=1;prime[j]\*i<=n;j++)
10. {
11. vis[prime[j]\*i]=**true**;
12. **if**(i%prime[j]==0)
13. {
14. phi[i\*prime[j]]=phi[i]\*prime[j];
15. **break**;
16. }
17. phi[i\*prime[j]]=phi[i]\*(prime[j]-1);
18. }
19. }
20. }

**6.3.6 经典例题**

**6.3.6.1 HDU2824 The Euler function**

**题目描述**

The Euler function phi is an important kind of function in number theory, (n) represents the amount of the numbers which are smaller than n and coprime to n, and this function has a lot of beautiful characteristics. Here comes a very easy question: suppose you are given a, b, try to calculate (a)+ (a+1)+....+ (b)

**输入格式**

There are several test cases. Each line has two integers a, b (2<a<b<3000000).

**输出格式**

Output the result of (a)+ (a+1)+....+ (b)

**输入样例**

3 100

**输出样例**

3042

**思路：**线性求欧拉函数模板题

**题解：**

1. #include<bits/stdc++.h>
2. **using** **namespace** std;
3. **typedef** **long** **long** ll;
4. **const** **int** N=3e6+10;
5. **int** phi[N];
6. **bool** vis[N];
7. **int** prime[N];
8. **int** sum[N];
9. **void** eular\_()//欧拉筛模板
10. {
11. memset(vis,**true**,**sizeof**(vis));
12. memset(phi,0,**sizeof**(phi));
13. **int** cnt=1;
14. vis[1]=**false**;
15. phi[1]=1;//特判1,gcd(1,1)=1
16. **for**(**int** i=2;i<=N-5;i++)
17. {
18. **if**(vis[i])
19. {
20. prime[cnt++]=i;
21. phi[i]=i-1;
22. }
23. **for**(**int** j=1;j<cnt&&prime[j]\*i<=N-5;j++)
24. {
25. vis[prime[j]\*i]=**false**;
26. **if**(i%prime[j]==0)
27. {
28. phi[prime[j]\*i]=phi[i]\*prime[j];
29. **break**;
30. }
31. phi[prime[j]\*i]=phi[i]\*phi[prime[j]];
32. //若i为p[j]的倍数，则i=k\*p[j]，在和p[j+1]相乘后得出的x=i\*p[j+1]=p[j]\*k\*p[j+1]
33. //则在i=k\*p[j+1]时，由于j从小到大，必定经过之前的p[j]，会重复计算
34. }
35. }
36. }
37. **int** main()
38. {
39. **int** a,b;
40. eular\_();
41. **while**(cin>>a>>b)
42. {
43. ll ans=0;
44. **for**(**int** i=a;i<=b;i++) ans+=phi[i];
45. cout<<ans<<endl;
46. }
47. **return** 0;
48. }

**6.3.6.2 HDU 2588 GCD**

**题目描述**

The greatest common divisor GCD(a,b) of two positive integers a and b,sometimes written (a,b),is the largest divisor common to a and b,For example,(1,2)=1,(12,18)=6.

(a,b) can be easily found by the Euclidean algorithm. Now Carp is considering a little more difficult problem:

Given integers N and M, how many integer X satisfies 1<=X<=N and (X,N)>=M.

**输入格式**

The first line of input is an integer T(T<=100) representing the number of test cases. The following T lines each contains two numbers N and M (2<=N<=1000000000, 1<=M<=N), representing a test case.

**输出格式**

For each test case,output the answer on a single line.

**输入样例**

3

1 1

10 2

10000 72

**输出样例**

1

6

260

**思路：**当M为1 时，易知答案为N，当M>1时，可以确定满足条件的数一定时N大于M的约数，如果x不能整除N，则gcd（x,N）=1<M，我们知道N的约数（设为x）和N的最大公约数为x，那是否本题只有N的约数满足条件呢？显然不是，数x\*k（k为满足条件的某些数）与N的约数同样为x，倒退一下，当gcd（N，x\*k）=x时，有gcd（N/x，k）=1，即k满足与N/x互质时满足N与x\*k的最大公约数为x，于是本题就变成了就1-N/x内，与N/x互质的数的个数，用欧拉函数求即可

**题解：**

1. #include<bits/stdc++.h>
2. **using** **namespace** std;
3. **typedef** **long** **long** ll;
4. **const** **int** N=1e5;
5. ll eular(ll x)
6. {
7. ll ans=x;
8. **for**(**int** i=2;i\*i<=x;i++)
9. {
10. **if**(x%i==0)
11. {
12. ans=ans-ans/i;
13. }
14. **while**(x%i==0)
15. {
16. x/=i;
17. }
18. }
19. **if**(x>1) ans=ans-ans/x;
20. **return** ans;
21. }
22. **int** main()
23. {
24. **int** t;
25. cin>>t;
26. **while**(t--)
27. {
28. ll ans=0;
29. ll n,m;
30. cin>>n>>m;
31. **int** i;
32. **for**(i=1;i\*i<n;i++)
33. {
34. **if**(n%i==0)
35. {
36. **if**(i>=m)
37. {
38. ans+=eular(n/i);
39. }
40. **if**(n/i>=m)
41. {
42. ans+=eular(i);
43. }
44. }
45. }
46. **if**(i\*i==n&&i>=m) ans+=eular(i);
47. cout<<ans<<endl;
48. }
49. **return** 0;
50. }

**6.3.6.3 HDU phi**

**题目描述**

给出若干个正整数n，请你求出最小的m，使得φ(m)≥n。

**输入格式**

本题有多组输入。

第一行一个正整数T表示数据组数

接下来T行每行一个正整数n

数据保证1≤T≤104,1≤n≤106。

**输出格式**

共T行，每行一个数代表对应的答案

**输入样例**

5

1

2

3

4

5

**输出样例**

1

3

5

5

7

**题解：**

1. #include<bits/stdc++.h>
2. **using** **namespace** std;
3. **typedef** **long** **long** ll;
4. **const** ll N=5e6+10;
5. **int** phi[N-5];
6. **bool** vis[N-5];
7. **int** prime[N-5];
8. **struct** node
9. {
10. **int** pos;
11. **int** val;
12. **friend** **bool** operator < (**const** node x,**const** node y)//对输入进行处理，优先处理n小的输入并记录
13. {
14. **return** x.val<y.val;
15. }
16. }node[10005];
17. **int** ans[10005];
18. **int** t;
19. **void** euler()
20. {
21. **int** now=1;
22. memset(vis,**false**,**sizeof**(vis));
23. memset(phi,0,**sizeof**(phi));
24. vis[1]=**true**;
25. phi[1]=1;
26. **int** cnt=0;
27. **for**(**int** i=1;i<=N-10;i++)
28. {
29. **if**(!vis[i])
30. {
31. prime[++cnt]=i;
32. phi[i]=i-1;
33. }
34. **for**(**int** j=1;j<=cnt&&prime[j]\*i<=N-10;j++)
35. {
36. vis[prime[j]\*i]=**true**;
37. **if**(i%prime[j]==0)
38. {
39. phi[i\*prime[j]]=phi[i]\*prime[j];
40. **break**;
41. }
42. phi[i\*prime[j]]=phi[i]\*phi[prime[j]-1];
43. }
44. **while**(phi[i]>=node[now].val&now<=t)
45. {
46. ans[node[now].pos]=i;
47. now++;
48. }
49. **if**(now>t) **return** ;
50. }
51. }
52. **int** main()
53. {
54. euler();
55. cin>>t;
56. **for**(**int** i=1;i<=t;i++)
57. {
58. cin>>node[i].val;
59. node[i].pos=i;
60. }
61. sort(node+1,node+t+1);
62. euler();
63. **for**(**int** i=1;i<=t;i++)
64. cout<<ans[i]<<endl;
65. **return** 0;
66. }