**Ngày 23: Bài toán tổng ước, đếm ước của số n**

**Áp dụng công thức**

**n=p1a1p2a2…pkak**

*for(int i=2;i<=sqrt(n);i++)*

*if(n%i==0)*

*{*

*p[++k]=i;*

*a[k]=0;*

*while(n%i==0)*

*{*

*a[k]++;*

*n/=i;*

*}*

*}*

*if(n>1)*

*{*

*p[++k]=n;*

*a[k]=1;*

*}*

**d=(a1+1)(a2+1)…(ak+1); // Số lượng ước của n**

*int\_fast64\_t demuoc(int\_fast64\_t n)*

*{*

*int\_fast64\_t k=0,ans=1, p[100], a[100];*

*for(int i=2;i<=sqrt(n);i++)*

*if(n%i==0)*

*{*

*p[++k]=i;*

*a[k]=0;*

*while(n%i==0)*

*{*

*a[k]++;*

*n/=i;*

*}*

*}*

*if(n>1)*

*{*

*p[++k]=n;*

*a[k]=1;*

*}*

*for(int i=1;i<=k;i++) ans\*=(a[i]+1); return ans;*

*}*

**S= // Tổng các ước của n**

*int\_fast64\_t tonguoc(int\_fast64\_t n)*

*{*

*int\_fast64\_t k=0,ans=1, p[100], a[100];*

*for(int i=2;i<=sqrt(n);i++)*

*if(n%i==0)*

*{*

*p[++k]=i;*

*a[k]=0;*

*while(n%i==0)*

*{*

*a[k]++;*

*n/=i;*

*}*

*}*

*if(n>1)*

*{*

*p[++k]=n;*

*a[k]=1;*

*}*

*for(int i=1;i<=k;i++)*

*ans\*=(((powl(p[i],a[i]+1)))-1)/(p[i]-1); return ans;*

*}*

***//*** *Lưu ý dùng* ***powl(a,n)*** *cho độ chính xác*

**Cách 2: Tính tổng ước của n cho tối ưu của n<106**

*void tonguoc()*

*{*

for (int\_fast64\_t i = 1; i <= 1000000; ++ i) {

for (int\_fast64\_t j = i; j <= 1000000; j += i)

{

sum[j] += i;

}

*}*

**Cách 2: Đếm các ước của ncho tối ưu của n<106**

int\_fast64\_t d[100000]={0}

*void demuoc()*

*{*

for (int\_fast64\_t i = 1; i <= 1000000; ++ i) {

for (int\_fast64\_t j = i; j <= 1000000; j += i)

{

d[j] ++;

}

*}*

**Bài 3. (3,0 điểm) Tổng ước chẵn**

Cho số nguyên dương ước nguyên dương của là số thỏa mãn: chia hết cho

**Yêu cầu:** Hãy lập trình đếm số các số nguyên dương trong đoạn có tổng các ước là một số chẵn.

**Dữ liệu vào:**

* Dòng đầu ghi số nguyên dương
* dòng tiếp theo mỗi dòng ghi hai số nguyên

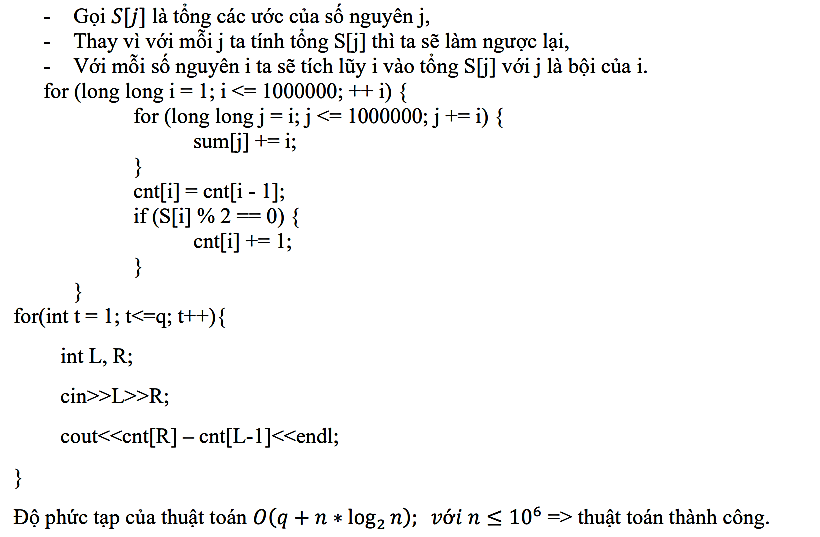
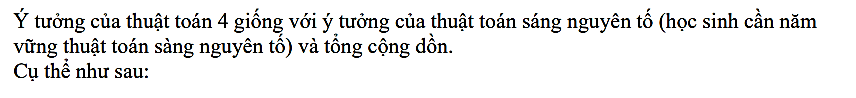
**Kết quả:**

* Ghi ra dòng, mỗi dòng ghi ra số lượng số nguyên dương trong đoạn thỏa mãn có tổng các ước là một số chẵn.

**Ví dụ:**

|  |  |
| --- | --- |
| **input** | **out** |
| 1  1 5 | 2 |
| 2  1 5  3 10 | 2  4 |

* **Ràng buộc 1:** có 25% số test của bài ứng với 25% số điểm của bài có
* **Ràng buộc 2:** có 25% số test của bài ứng với 25% số điểm của bài có
* **Ràng buộc 3:** có 25% số test của bài ứng với 25% số điểm của bài có
* **Ràng buộc 4:** có 25% số test của bài ứng với 25% số điểm của bài có

******

**1. Dấu hiệu chia hết: s là sâu**

Cho hết cho 2: s[s.size()-1]%2=0

Cho hết cho 3: (s0+s1+…+ s.size()-1)%3

Cho hết cho 4: hai chữ tận cùng chia hết cho 4

Cho hết cho 5: Chữ số tận cùng là 0 hoặc 5

Cho hết cho 6 chia hết cho 2 và 3

Chia hết cho 7: (..(((s0\*3)+s1)\*3+s2)\*3+…+s[s.size()-1]\*3)))..)+ s.size()-1)%7=0

Chia hết 8: 3 Chữ số cuối cùng chia hết cho 8

Chia hết cho 9: (s0+s1+…+ s.size()-1)%3

**2. Tìm số dư**

*a. Số nhỏ hơn 10 chữ số*

Tìm số dư của a cho b

Có n=a/b

R là số dư cần tìm: R=a-n.b

*b. Số lớn hơn 10 chữ số*

a=a1a2a3a4a5a6a7a8a9a10a11a12..an

Tìm phần dư của a1a2a3a4a5a6a7a8a9a10 với b được r1

a= r1 a11a12 a13a14 a15a16 a17a18 a19a20 a21a22..an

Tìm phần dư của r1 a11a12 a13a14 a15a16 a17a18 a19 với b được r2

a= r2 a20a21..an làm tương tự đến khi dc rn

**3. Tìm số dư của tổng, tích lớn**

**(a+b)%m=(a%m+b%m)%m**

**(a-b)%m=(a%m-b%m)%m**

**(a.b)%m=(a%m.b%m)%m**

**Đề 2022: Tìm số dư S=a+2a+..+na cho 109+7**

**// S=a(1+2+..+n)**

*#include<bits/stdc++.h>*

*using namespace std;*

*int main()*

*{*

*int\_fast64\_t a,n,ans,m=1000000007;*

*cin>>a>>n;*

*if(n%2==0)*

*{ans=((a%m)\*((n/2)%m)\*((n+1)%m))%m;*

*cout<<ans;*

*}*

*else*

*{*

*ans=((a%m)\*(n%m)\*(((n+1)/2)%m))%m;*

*cout<<ans;*

*}*

*}*

**4. Tìm số dư an cho b**

**(an )%b = ?**

**k=n%(b-1)**

**x=a%b**

**(an )%b = (xk)%b**

**5. Tìm chữ số tận cùng của an**

Nếu a có tận cùng là 0;1;5;6 thì an cũng có tận cùng là: 0;1;5;6

Nếu a tận cùng bằng 9: n chẵn: tận cùng bằng 1, n lẻ tận cùng bằng 9

Nếu a tận cùng bằng 4: n chẵn: tận cùng bằng 6, n lẻ tận cùng bằng 4

Nếu a có tận cùng là 2;3;7,8:

Lấy n chia cho 4: n=4k+r với r={0;1;2;3}

*#include <bits/stdc++.h>*

*using namespace std;*

*int\_fast64\_t tclt(string s,int\_fast64\_t k)*

*{*

*int\_fast64\_t n;*

*n=s[s.size()-1]-'0';*

*if( n==0 || n==5 || n==6 || n==1 && k!=0) return n; //0,1,5,6 dung voi moi TH*

*if( n==4 || n==9) //4,9 xet so mu chan hoac le*

*{*

*if(k%2==1) return n;*

*if(k%2==0)*

*{*

*if(n==4) return 6;*

*return 1;*

*}*

*}*

*//con lai 2,3,7,8*

*if( k%4==1 ) return n;*

*if( k%4==2 )*

*{*

*if( n==2 || n==8 ) return 4;*

*if( n==3 || n==7 ) return 9;*

*}*

*if( k%4==3 )*

*{*

*if( n==3 ) return 7;*

*if( n==7 ) return 3;*

*if( n==2 ) return 8;*

*if( n==8 ) return 2;*

*}*

*if( k%4==0 )*

*{*

*if( n==3 || n==7 ) return 1;*

*if( n==2 || n==8 ) return 6;*

*}*

*}*

*int main()*

*{*

*string s;*

*cin>>s;*

*int\_fast64\_t k;*

*cin>>k;*

*cout<<tclt(s,k)<<endl;*

*}*