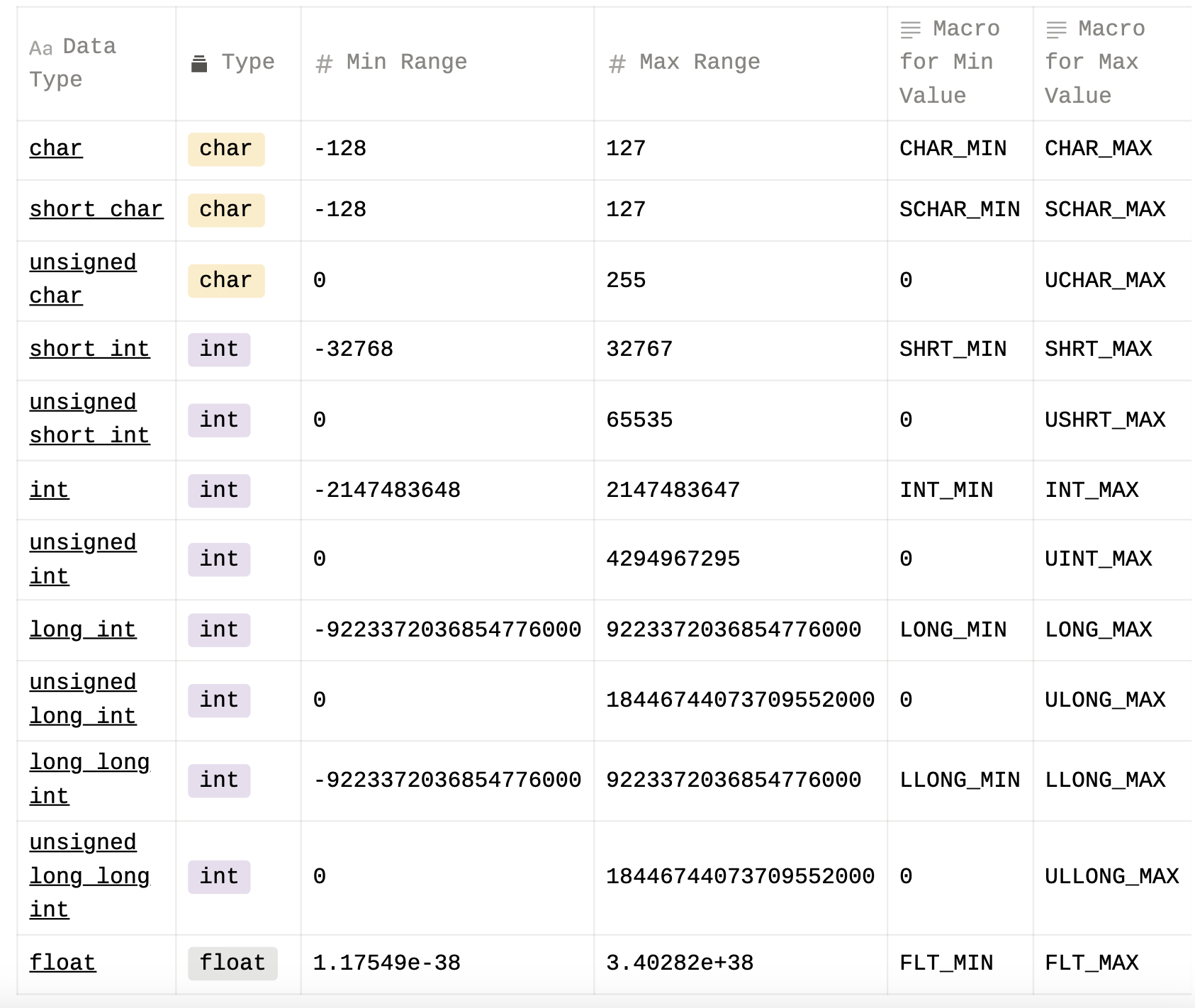
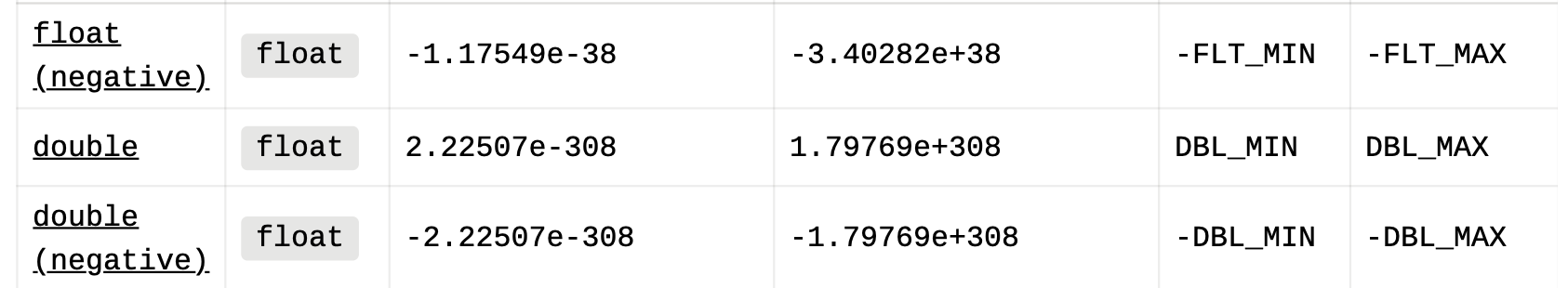
**CHUYÊN ĐỀ 1. KIẾN THỨC CƠ BẢN**

**1.Kiểu dữ liệu**

**** ****

**2. Dữ liệu có cấu trúc**

**A. Mảng**

**a)Mảng một chiều Kích thước: 106**

**1. Nhập mảng**

cin>>n;

for( i=1;i<=n;i++) cin>>A[i];

**2. Xuất mảng**

for( i=1;i<=n;i++) cout<<A[i]<<“ “;

**3. Các hàm , thủ tục trên mảng**

**x=\*max\_element(a+start,a+1+end);**

Giá trị lớn nhất của mảng a từ start đến end

**y=max\_element(a+start,a+1+end)-a;**

Vị trí của Giá trị lớn nhất trong mảng a từ start đến end

**x=\*min\_element(a a+start,a+1+end);**

Giá trị nhỏ nhất của mảng a từ start đến end

**y=min\_element(a+start,a+1+end)-a**;

Vị trí của Giá trị nhỏ nhất trong mảng a từ start đến end

**sort(a+start,a+1+end):** Sắp xếp a tăng dần từ start đến end

*bool comp(int x,int y)*

*{*

*return (x>y);*

*}*

**sort(a+start,a+1+end,comp);//**Sắp xếp a giảm dần từ start đến end

**binary\_seach(a+start,a+1+end,val);** Tìm kiếm nhị phân val trong mảng sắp xếp có:true, không:false

**\*lower\_bound(a+start,a+1+end,val);//**Tìm giá trị đầu tiên lớn hơn hoặc bằng Val trong mảng a đã sắp xếp

**lower\_bound(a+start,a+1+end,val)-a;//**Vị trí đầu tiên lớn hơn hoặc bằng Val trong mảng a đã sx

**\*upper\_bound(a+start,a+1+end,val);//**Tìm giá trị đầu tiên lớn hơn Val trong mảng a đã sắp xếp

**upper\_bound(a+start,a+1+end,val)-a;//**Vị trí đầu tiên lớn hơn Val trong mảng a đã sắp xếp

**4. Mảng thống kê**

cin>>n;

memset(d,0,sizeof(d));

for( i=1;i<=n;i++)

{

cin>>A[i]; d[a[i]]++;

}

**5. Mảng cộng dồn**

cin>>n;

memset(d,0,sizeof(d));

for( i=1;i<=n;i++)

{

cin>>A[i];

if(**điều\_kiện**)d[i]=d[i-1]+1;else d[i]=d[i-1];

}

**6. Mảng tổng**

cin>>n;

memset(s,0,sizeof(d));

for( i=1;i<=n;i++)

{

cin>>A[i]; s[i]=d[i-1]+a[i];

}

**b)Nếu kích thước mảng> 106 nên dùng map**

Khai baó:

**map<int,int> d;**

cin>>n;

for( i=1;i<=n;i++)

{

cin>>A[i]; d[a[i]]++;

}

c) Nếu dữ liệu kiểu căp nên dùng pair

Khai báo:

pair<int,int>a; // Phần tử cặp

pair<int,int>a[10000];// mảng cặp

Nhập 1 cặp

cin>>a.first>>a.second;

Nhập mảng cặp

cin>>n;

for( i=1;i<=n;i++)

{

cin>>a[i].first>>a.[i]second;

**}**

**Lưu ý: Hàm sort sắp xếp cặp theo firsst**

**B. XÂU**

**1. Nhập xâu**

Nhập 1 xâu có dấu cách: **getline(cin,s);**

Nhập nhiều xâu: **cin>>s**

2.Xuất xâu:

**cout<<S;**

**3. Các hàm trên xâu**

**S.size()** :Chiều dài xâu

**S.find(T**) : Tìm vị trí xuất hiện đầu tiên xâu T trong S nếu không trả -1

**S.rfind(T)** : Tìm vị trí xuất hiện cuối cùng xâu T trong S nếu không trả -1

**S.substr(vt,n):** Sao chép xâu s từ vị trí vt, n phần tử

**3. Các thủ tục trên xâu**

**S.c\_str():** trả về 1 mảng ký tự biểu diễn s

**S.erase(vt,n):** Xóa xâu S từ vị trí vt xóa n phần tử

**T.insert(vt,S):** Chèn xâu s vào t tại vị trí vt

**atoll(S.c\_str()):** Chuyển xâu s thành số.

**itoa(x,k,10) :** chuyển số thành xâu

int x; char k[33]

|  |  |  |
| --- | --- | --- |
| Chuyển số | Chuyển xâu | Tách số |

****

**CHUYÊN ĐỀ 2. XỬ LÝ SỐ**

**BÀI TOÁN TỔNG ƯỚC, ĐẾM ƯỚC CỦA SỐ N**

**A. ƯỚC**

**1. Áp dụng công thức**

**n=p1a1p2a2…pkak**

***a. Số lượng ước của n***

**d=(a1+1)(a2+1)…(ak+1);**

***b. Tổng các ước của n***

**S=**

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

*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;*

*}*

**2. Dùng mảng lưu trữ**

***a. Tính tổng ước của n <1000000 dùng mảng tổng***

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

*for ( j = i; j <= 1000000; j += i) sum[j] += i;*

***b. Đếm các ước của n <1000000 dùng mảng đếm***

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

*for ( j = i; j <= 1000000; j += i) d[j] ++;*

***Lưu ý: dùng mảng lưu trữ phù hợp với những bài toán lặp gọi thao tác nhiều lần***

***Ví dụ:***

**Bài 3. (3,0 điểm) Tổng ước chẵn**

Cho số nguyên dương 𝑛, ước nguyên dương của 𝑛 là số 𝑖 = 1, 2, … , 𝑛 thỏa mãn: 𝑛 chia

hết cho 𝑖.

**Yêu cầu:** Hãy lập trình đếm số các số nguyên dương trong đoạn [𝐿, 𝑅] có tổng các ước là một

số chẵn.

**Dữ liệu vào:**

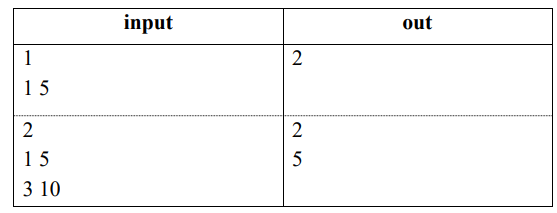
• Dòng đầu ghi số nguyên dương 𝑞 (1 ≤ 𝑞 ≤ 106);

• 𝑞 dòng tiếp theo mỗi dòng ghi hai số nguyên 𝐿, 𝑅 (1 ≤ 𝐿 ≤ 𝑅 ≤ 106)

**Kết quả:**

• Ghi ra 𝑞 dòng, mỗi dòng ghi ra số lượng số nguyên dương trong đoạn [𝐿, 𝑅] thỏa mãn có tổng các ước là một số chẵn.

**Ví dụ:**

****

• Ràng buộc 1: có 25% số test của bài ứng với 25% số điểm của bài có 𝑞 = 1, 𝑛 ≤ 1000;

• Ràng buộc 2: có 25% số test của bài ứng với 25% số điểm của bài có 𝑞, 𝑛 ≤ 1000;

• Ràng buộc 3: có 25% số test của bài ứng với 25% số điểm của bài có 𝑛 ≤ 5000;

• Ràng buộc 4: có 25% số test của bài ứng với 25% số điểm của bài có 𝑞, 𝑛 ≤ 106

Ý tưởng của thuật toán 4 giống với ý tưởng của thuật toán sáng nguyên tố (học sinh cần nắm vững ***thuật toán sàng nguyên tố***) và ***tổng cộng dồn.***

**Cụ thể như sau:**

- Gọi 𝑆[𝑗] là tổng các ước của số nguyên j,

- Thay vì với mỗi j ta tính tổng S[j] thì ta sẽ làm ngược lại,

- Với mỗi số nguyên i ta sẽ tích lũy i vào tổng S[j] với j là bội của i.

**for (long long i = 1; i <= 1000000; ++ i) {**

**for (long long j = i; j <= 1000000; j += i) {**

**sum[j] += i;**

**}**

**cnt[i] = cnt[i - 1];**

**if (S[i] % 2 == 0) {**

**cnt[i] += 1;**

**}**

}

**for(int t = 1; t<=q; t++){**

**int L, R;**

**cin>>L>>R;**

**cout<<cnt[R] – cnt[L-1]<<endl;**

}

*Độ phức tạp của thuật toán 𝑂(𝑞 + 𝑛 ∗ log2 𝑛); 𝑣ớ𝑖 𝑛 ≤ 106 => thuật toán thành công*

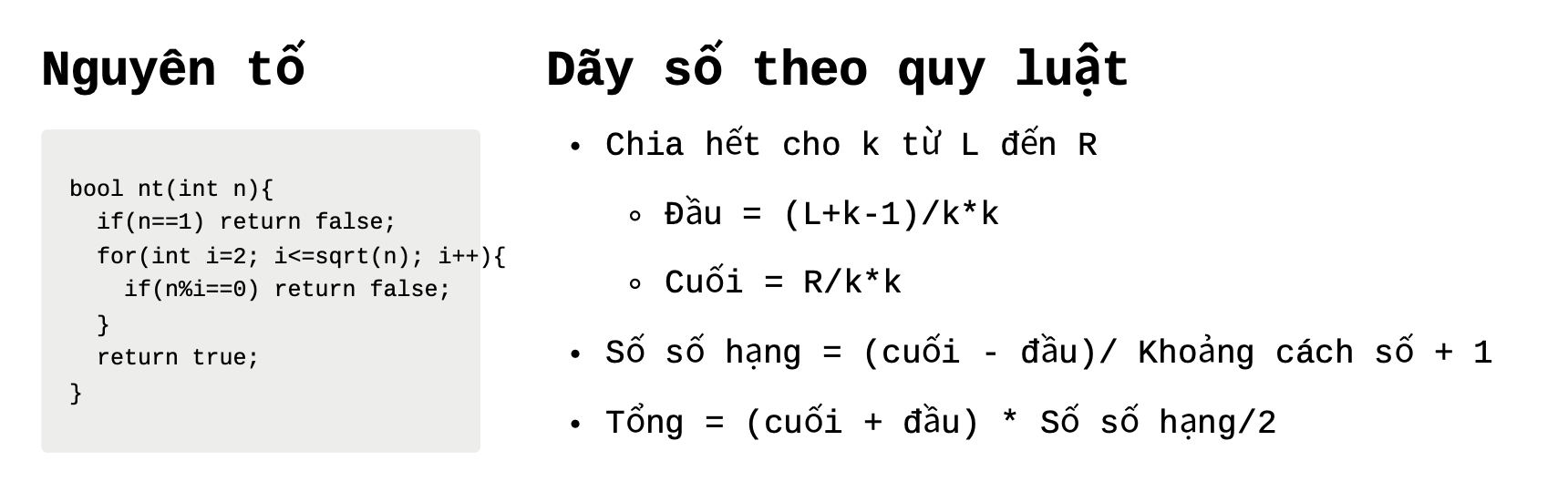
**B. BỘI**

Đếm các bội của a trong phạm vi từ L đến R

x=(L-1)/a

y=R/a

ans=y-x;

****

**C. TẠO SÀNG NGUYÊN TỐ:**

memset(ngto,true,100000);

ngto[0]=0;

ngto[1]=0;

for(i=2;i<=sqrt(32000);i++)

if(ngto[i])

for(int j=i\*i;j<=32000;j+=i) ngto[j]=false;

**CHUYÊN ĐỀ 2. CHIA HẾT, ĐỒNG DƯ**

**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;*

*}*

**CHUYÊN ĐỀ 3. DÃY CON LIÊN TIẾP CÓ TỔNG THOẢ ĐIỀU KIỆN(=<> S)**

1. **Dãy con liên tiếp có tổng bằng S**

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

*using namespace std;*

*long long p=1;*

*int n,i,S, a[100];int sum[100];*

*int main()*

*{*

*sum[0]=0;*

*cin>>n>>S;*

*for(i=1;i<=n;i++)*

*{*

*cin>>a[i];*

*sum[i]=sum[i-1]+a[i];*

*}*

*int res=0;*

*sort(sum,sum+n+1);*

*for(i=0;i<=n;i++)*

*{*

*if(binary\_seach(sum,i,n,sum[i]+S)) res++;*

*}*

*cout<<res;*

*}*

**2. Số dãy con liên tiếp có tổng lớn hơn hoặc bằng S**

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

*using namespace std;*

*long long p=1;*

*int n,i,S, a[100];int\_fast64\_t sum[100];*

*int main()*

*{*

*sum[0]=0;*

*cin>>n>>S;*

*for(i=1;i<=n;i++)*

*{*

*cin>>a[i];*

*sum[i]=sum[i-1]+a[i];*

*}*

*int res=0;*

*for(i=0;i<=n;i++)*

*{*

*int t=lower\_bound(sum,sum+n+1,sum[i]+S)-sum;*

*if(t<n+1) res+=n-t+1;*

*}*

*cout<<res;*

*}*

**3.Số dãy con liên tiếp có tổng lớn hơn S và nhỏ hơn s**

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

*using namespace std;*

*long long p=1;*

*int n,i,S,s, a[100];int\_fast64\_t sum[100];*

*bool comp(int a, int b)*

*{*

*return a <= b;*

*}*

*int main()*

*{*

*sum[0]=0;*

*cin>>n>>s>>S;*

*for(i=1;i<=n;i++)*

*{*

*cin>>a[i];*

*sum[i]=sum[i-1]+a[i];*

*}*

*int res=0;*

*sort(sum,sum+n+1);*

*for(i=0;i<n;i++)*

*{*

*int t1=upper\_bound(sum,sum+n+1,sum[i]+s,comp)-sum;*

*int t2=lower\_bound(sum,sum+n+1,sum[i]+S)-sum;*

*if(sum[t2]!=S)t2=t2-1;*

*cout<<"i="<<i<<" t1="<<t1<<" t2="<<t2<<endl;*

*if(t2>t1) res+=t2-t1+1;*

*if(t1==t2)res++;*

*cout<<" res"<<res<<endl;*

*}*

*cout<<res;*

*}*

**Dãy con dài nhất có tổng bằng 0**

**Bài 5**

**CHUYÊN ĐỀ 5. CHUYÊN ĐỀ XÂU**

**1. Nhập xâu**

Nhập 1 xâu có dấu cách: **getline(cin,s);**

Nhập nhiều xâu: **cin>>s**

2.Xuất xâu:

**cout<<S;**

**3. Các hàm trên xâu**

**S.size()** :Chiều dài xâu

**S.find(T**) : Tìm vị trí xuất hiện đầu tiên xâu T trong S nếu không trả -1

**S.rfind(T)** : Tìm vị trí xuất hiện cuối cùng xâu T trong S nếu không trả -1

**S.substr(vt,n):** Sao chép xâu s từ vị trí vt, n phần tử

**3. Các thủ tục trên xâu**

**S.c\_str():** trả về 1 mảng ký tự biểu diễn s

**S.erase(vt,n):** Xóa xâu S từ vị trí vt xóa n phần tử

**T.insert(vt,S):** Chèn xâu s vào t tại vị trí vt

**atoll(S.c\_str()):** Chuyển xâu s thành số.

**itoa(x,k,10) :** chuyển số thành xâu

int x; char k[33]

**KỸ THUẬT LOANG ĐỐI XỨNG DÀI NHẤT**

for(i=0;i<s.size();i++)

{

if(s[i]==s[i+1])

{

int\_fast64\_t j=0;

while(s[i-j]==s[i+1+j]&&i+j<s.size()&&i-j>=0)j++;

j\*=2;

if(ans<j)ans=j;

}

if(s[i]==s[i+2])

{

int\_fast64\_t j=0;

while(s[i-j]==s[i+2+j]&&i+j<s.size()&&i-j>=0)j++;

j=j\*2+1;

if(ans<j)ans=j;

}

}

**KỸ THUẬT LOANG ĐẾM SỐ XÂU CON ĐỐI XỨNG**

for(i=0;i<s.size();i++)

{

if(s[i]==s[i+1])

{

int\_fast64\_t j=0;

while(s[i-j]==s[i+1+j]&&i+j<s.size()&&i-j>=0){j++;ans++;}

}

if(s[i]==s[i+2])

{

int\_fast64\_t j=0;

while(s[i-j]==s[i+2+j]&&i+j<s.size()&&i-j>=0){j++;ans++;}

}

}

cout<<ans+s.size();

**Bài 8**

**Bài 9**

**Bài 11**

**CHUYÊN ĐỀ 6.. MẢNG 1 CHIỀU**