CHUYÊN ĐỀ 1. KIẾN THỨC CƠ BẢN

1.Kiểu dữ liệu

Data Type	Type	Min Range	Max Range	Macro for Min Value	Macro for Max Value
char	char	-128	127	CHAR_MIN	CHAR_MAX
short char	char	-128	127	SCHAR_MIN	SCHAR_MAX
unsigned char	char	0	255	0	UCHAR_MAX
short int	int	-32768	32767	SHRT_MIN	SHRT_MAX
unsigned short int	int	0	65535	0	USHRT_MAX
int	int	-2147483648	2147483647	INT_MIN	INT_MAX
unsigned int	int	0	4294967295	0	UINT_MAX
long int	int	-9223372036854776000	9223372036854776000	LONG_MIN	LONG_MAX
unsigned long int	int	0	18446744073709552000	0	ULONG_MAX
long long int	int	-9223372036854776000	9223372036854776000	LLONG_MIN	LLONG_MAX
unsigned long long int	int	0	18446744073709552000	0	ULLONG_MAX
float	float	1.17549e-38	3.40282e+38	FLT_MIN	FLT_MAX
float (negative)	float	-1.17549e-38	-3.40282e+38	-FLT_MIN	-FLT_MAX
double	float	2.22507e-308	1.79769e+308	DBL_MIN	DBL_MAX
double (negative)	float	-2.22507e-308	-1.79769e+308	-DBL_MIN	-DBL_MAX

2. Dữ liệu có cấu trúc

A. Mảng

a)Mảng một chiều Kích thước: 10^6

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> 10^6 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

`atoll(s.c_str());`	`int x; char k[33];`
	`itoa(x, k, 10);`
	
	`tostring(s);`
		`while(n>0){`
		`  X = n % 10;`
		`  n /= 10;`
		`}`

CHUYÊN ĐỀ 2. XỬ LÝ SỐ
BÀI TOÁN TỔNG ƯỚC, ĐẾM ƯỚC CỦA SỐ N
A. ƯỚC
1. Áp dụng công thức
$n=p1^{a1}p2^{a2}\dots pk^{ak}$
a. Số lượng ước của n

$d=(a_1+1)(a_2+1)\dots(a_k+1);$

b. Tổng các ước của n

$$S=\frac{p1^{a1+1}-1}{p_1-1}.\frac{p2^{a2+1}-1}{p_2-1}\dots\frac{p1^{ak+1}-1}{p_k-1}$$

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=$(a_1+1)(a_2+1)\ldots(a_k+1)$; // 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=$\frac{(p1^{a1+1}-1}{p_1-1}.\frac{(p2^{a2+1}-1}{p_2-1}....\frac{(p1^{ak+1}-1}{p_k-1}$ // 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 $i = 1, 2, \dots, n$ 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 ($1 \le q \le 10^6$);

• q dòng tiếp theo mỗi dòng ghi hai số nguyên L, R ($1 \le L \le R \le 10^6$)

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:

input	out
1 1 5	2
2 1 5 3 10	2 5

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

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

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

• Ràng buộc 4: có 25% số test của bài ứng với 25% số điểm của bài có $q, n \leq 10^6$

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 $n \leq 106$ => 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;

Dãy số theo quy luật

- Chia hết cho k từ L đến R
  - Đầu = (L+k-1)/k*k
  - Cuối = R/k*k
- Số số hạng = (cuối - đầu)/ Khoảng cách số + 1
- Tổng = (cuối + đầu) * Số số hạng/2

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

$a=a_1a_2a_3a_4a_5a_6a_7a_8a_9a_{10}a_{11}a_{12}..a_n$

Tìm phần dư của $a_1a_2a_3a_4a_5a_6a_7a_8a_9a_{10}$ với b được r_1

a= r_1 $a_{11}a_{12}$ $a_{13}a_{14}$ $a_{15}a_{16}$ $a_{17}a_{18}$ $a_{19}a_{20}$ $a_{21}a_{22}..a_n$

Tìm phần dư của r_1 $a_{11}a_{12}$ $a_{13}a_{14}$ $a_{15}a_{16}$ $a_{17}a_{18}$ a_{19} với b được r_2

a= r_2 $a_{20}a_{21}..a_n$ làm tương tự đến khi dc r_n

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 a^n CHO b

(a^n)%b = ?

k=n%(b-1)

x=a%b

(a^n)%b = (x^k)%b

5. TÌM CHỮ SỐ TẬN CÙNG CỦA a^n

- **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();
```

CHUYÊN ĐỀ 6.. MẢNG 1 CHIỀU