

TÀI LIỆU LẬP TRÌNH C++ CƠ BẢN

BUỔI 4:

- 1/ Mảng một chiều
- 2/ Bài tập cơ bản về mảng 1 chiều.

I. MẢNG MỘT CHIỀU

1. Biến mảng

Chúng ta xét ví dụ: Viết chương trình nhập vào 6 số nguyên và tính tổng của 6 số nguyên đó.

Nhận xét:

- Ta phải khai báo 6 biến để lưu trữ giá trị của 6 số, và 1 biến để tính tổng, khai báo như sau:

```
int a, b, c, d, e, g, tong;
```

- Ta viết lệnh nhập giá trị cũng khá dài:

```
cout << "Số thứ nhất: "; cin >> a;
cout << "Số thứ hai: "; cin >> b;
cout << "Số thứ ba: "; cin >> c;
cout << "Số thứ tư: "; cin >> d;
cout << "Số thứ năm: "; cin >> e;
cout << "Số thứ sáu: "; cin >> g;
```

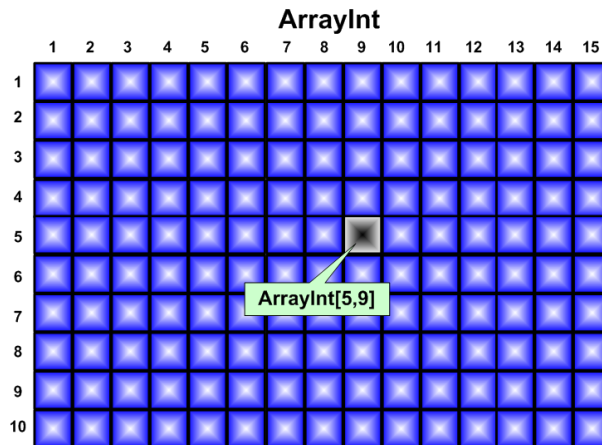
- Ta tính tổng như sau:

```
tong = a + b + c + d + e + g;
```

Và sẽ ra sao nếu ta cần làm việc với 100 số hoặc nhiều hơn nữa?

Nhưng rõ ràng, nếu xét kỹ ta sẽ thấy: Các biến đều cùng một kiểu dữ liệu. Như vậy thay vì phải khai báo nhiều biến riêng lẻ, ta sẽ sử dụng biến mảng để thay thế cho các biến riêng lẻ đó.

Biến mảng: Là biến có dữ liệu kiểu mảng. Đó là kiểu dữ liệu gồm một dãy hữu hạn nhiều ô nhớ liên tiếp nhau tạo thành một mảng (như một bảng, gồm nhiều hàng và nhiều cột). Mỗi ô nhớ được gọi là một phần tử của mảng. Các phần tử này có cùng kiểu dữ liệu và được đánh thứ tự. Mỗi phần tử của biến mảng chứa được một giá trị và được truy xuất thông qua vị trí của phần tử đó trong mảng.



2. Mảng 1 chiều

Là một dãy các phần tử cùng kiểu dữ liệu, mỗi phần tử được đánh dấu bằng một vị trí (chỉ số) xác định.

Ví dụ: Mảng a gồm 6 phần tử: 5 3 6 8 1 4

| | | | | | | |
|-----------|---|---|---|---|---|---|
| Giá trị → | 5 | 3 | 6 | 8 | 1 | 4 |
| Vị trí → | 0 | 1 | 2 | 3 | 4 | 5 |

Ta có:

- Phần tử $a[0]$ có giá trị là 5
- Tổng 2 phần tử $a[2] + a[3] = 6 + 8 = 14$
- Phần tử có giá trị lớn nhất là $a[3]$ (giá trị lớn nhất là 8 nằm ở vị trí thứ 3)
- Các phần tử có giá trị chẵn là $a[2]$, $a[3]$, $a[5]$

3. Khai báo mảng 1 chiều

<kiểu dữ liệu> <tên mảng> [<số lượng phần tử>];

Ví dụ:

- Mảng a có tối đa 100 phần tử kiểu số nguyên:

int a[100];

- Mảng b và c có tối đa 250 phần tử kiểu số thực:

float b[250];

float c[250];

- Mảng ds có tối đa 1000 phần tử kiểu xâu:

string ds[1000];

Lưu ý: Phần tử đầu tiên mặc định có chỉ số là 0, chẳng hạn như khai báo mảng a có 100 phần tử như ở trên thì phần tử đầu tiên là $a[0]$, và phần tử cuối cùng sẽ là $a[99]$.

4. Cách nhập/xuất các phần tử của mảng 1 chiều

Một mảng sẽ luôn luôn có một biến số nguyên để lưu trữ số lượng phần tử hiện tại của mảng (số lượng hiện tại không phải lúc nào cũng bằng chỉ số cuối do trong quá trình làm việc, số lượng phần tử của mảng có thể thay đổi). Ví dụ đề bài sau đây sẽ không cho chúng ta biết chính xác mảng có bao nhiêu phần tử:

Viết chương trình nhập vào một mảng a có n phần tử kiểu số nguyên ($n < 10^6$) và cho biết tổng của các phần tử trong mảng.

Rõ ràng đề bài tổng quát số lượng phần tử là n và n tối đa có thể là 10^6 phần tử. Do đó ta phải khai báo như sau:

int a[1000006];

Tại sao phải là 1.000.006 mà không phải là 1.000.000. Đừng hỏi tôi, hãy hỏi Google :D

Nhưng trong quá trình làm việc, người dùng có thể nhập 10, 20 phần tử tùy theo tính chất công việc. Do đó ta sẽ phải khai báo thêm một biến n để ghi nhớ số phần tử hiện tại của mảng. Và do đó khai báo để làm việc với mảng phải là:

int a[1000006]; int n;

Để nhập giá trị cho các phần tử của mảng, với số lượng phần tử $n=6$, ta sẽ nhập lần lượt bằng lệnh như sau:

```
cout << "Gia tri cua phan tu thu 1: "; cin >> a[1];
cout << "Gia tri cua phan tu thu 2: "; cin >> a[2];
cout << "Gia tri cua phan tu thu 3: "; cin >> a[3];
cout << "Gia tri cua phan tu thu 4: "; cin >> a[4];
cout << "Gia tri cua phan tu thu 5: "; cin >> a[5];
cout << "Gia tri cua phan tu thu 6: "; cin >> a[6];
```

Để thấy rằng ta có thể thay bằng một vòng lặp For

```
for (int i=0; i<n; i++)
{
    cout << "Gia tri cua phan tu thu " << i << ": ";
    cin >> a[i];
}
```

Tương tự, đối với việc xuất giá trị của phần tử ra màn hình, ta có thể sử dụng vòng lặp như sau:

for (int i=0; i<n; i++) cout << a[i] << " ";

II. BÀI TẬP CƠ BẢN VỀ MẢNG 1 CHIỀU

1. Chương trình tính tổng và trung bình các phần tử của mảng

Cài đặt:

```
Nhập mảng n phần tử a[i].
Khai báo và gán biến tổng S=0.
For i từ 1 đến n
    S=S+a[i];
tb = S/n;
Thông báo tổng và trung bình.
```

Lưu ý:

- Kết quả của phép chia là giá trị ở kiểu số thực, do đó biến trung bình phải khai báo kiểu số thực.

- Khi in giá trị số thực ra màn hình, ta cần định dạng số chữ số thập phân được in ra. Cú pháp in lấy 2 chữ số thập phân được viết như sau:

```
float tb=(float) s/n;
cout << "Trung binh cua mang la: " ;
cout << fixed<< setprecision (2) << tb << '\n';
```

- Chương trình:

```
#define c 1000006
int a[c];
int main()
{
    //Nhap mang
    int n;
    cin >>n;
    for (int i=1; i<=n; i++) cin >>a[i];
    //Tinh tong
    int s=0;
    for (int i=1; i<=n; i++) s+=a[i];
    //Tinh trung binh
    float tb=(float) s/n;
    cout << "Trung binh cua mang la: " ;
    cout << fixed<< setprecision (2) << tb << '\n';
    return 0;
}
```


2. Chương trình tìm phần tử lớn nhất trong mảng

Ý tưởng: Khai báo một biến max, cho max là phần tử đầu tiên, sau đó duyệt qua từng phần tử còn lại, phần tử nào lớn hơn max thì gán lại max bằng phần tử đó.

Cài đặt:

Nhập mảng n phần tử a[i].

Khai báo và gán biến lớn nhất có giá trị là phần tử đầu tiên $\text{max} = a[1]$.

For i từ 2 đến n //Phần tử đầu tiên đã xét rồi

Nếu $\text{max} < a[i]$ thì gán $\text{max} = a[i]$;

Thông báo giá trị lớn nhất (max).

```
int maxx=a[1];
for (int i=2; i<=n; i++) if (a[i]>maxx) maxx=a[i];
cout<<"Phan tu lon nhat: " <<maxx;
```

3. Tìm kiếm một giá trị trong mảng

Cài đặt:

Nhập phần tử cho mảng

Nhập vào một số x

Khai báo một biến v và gán bằng -1 (nghĩa là chưa tìm thấy)

Duyệt qua các phần tử a[i]

Nếu $a[i] = x$ thì gán $v = i$ (với i là vị trí xuất hiện của x)

Kiểm tra biến v, nếu $v \geq 0$ thì thông báo đã tìm thấy phần tử x trong mảng, ngược lại thông báo không tìm thấy.

```
#define c 1000006
int a[c];
int main()
{
    int n, x, v=-1;
    cin >>n;
    for (int i=1; i<=n; i++) cin >>a[i];
    cin>>x;
    for (int i=1; i<=n; i++) if (a[i]==x) v=i;
    if (v>=0) cout <<"Yes\n";
    else cout<<"No\n";
    return 0;
}
```

Nếu muốn dừng lại khi tìm thấy X trong mảng:

```
int v=-1;
for (int i=1; i<=n; i++)
{
    if (a[i]==x)
    {
        v=i;
        cout <<"Yes\n";
        break;
    }
}
if (v<0) cout<<"No\n";
```

4. Đếm số lượng phần tử dương

- Cài đặt:

Khai báo một biến đếm và gán bằng 0.

Duyệt qua các phần tử $a[i]$

Nếu $a[i]>0$ thì tăng đếm lên 1 đơn vị $dem++$

In ra giá trị tìm được.

- Chương trình:

```
int dem=0;
for (int i=1; i<=n; i++)
    if (a[i]>0) dem++;
cout<< "Co " << dem << " so duong " ;
```

5. Chương trình thêm, xóa phần tử ở cuối mảng

- Cài đặt thêm phần tử vào cuối mảng:

Tăng giá trị n lên 1 đơn vị (thêm 1 phần tử).

Nhập giá trị cần thêm vào mảng.

Gán $a[n] = x$;

- Chương trình:

```
n++;
int x;
cout << "Gia tri can them vao mang: "; cin >> x;
a[n]=x ;
```

Để xóa phần tử cuối mảng, ta chỉ cần giảm n đi 1 đơn vị.

6. Chương trình xóa phần tử ở vị trí k trong mảng

Cài đặt:

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|----|---|
| 5 | 3 | 6 | 8 | 1 | 4 | 9 | 7 | 10 | 2 |
|---|---|---|---|---|---|---|---|----|---|

k

Đem các phần tử phía sau k lên phía trước 1 vị trí:

(For i từ k đến n-1

$a[i] = a[i+1];$)

| | | | | | | | | | |
|---|---|---|---|---|---|---|----|---|---|
| 5 | 3 | 6 | 8 | 1 | 9 | 7 | 10 | 2 | 2 |
|---|---|---|---|---|---|---|----|---|---|

k

Giảm n đi 1 đơn vị $n=n-1$;

Để thêm phần tử vào một vị trí k, ta thực hiện tương tự:

Tăng n lên 1 đơn vị $n=n+1$;

Đem các phần tử từ vị trí k ra phía sau 1 vị trí:

(For i từ n ngược về k+1

$a[i] = a[i-1];$)

Nhập x và gán $a[k] = x$;

III. BÀI TẬP

1. Viết chương trình nhập vào một mảng số nguyên a gồm n phần tử ($n < 10^6$), giá trị của mỗi phần tử là khác nhau và nằm trong khoảng $-32000 < a_i < 32000$. Thực hiện các yêu cầu sau:

A) Thông báo các phần tử ra màn hình.

B) Tính tổng và trung bình các phần tử của mảng.

C) Tìm giá trị và vị trí của phần tử nhỏ nhất trong mảng

D) Đếm xem trong mảng có bao nhiêu số dương chẵn.

E) Xét xem một mảng có phải là mảng tăng dần hay không, nếu không phải mảng tăng dần thì thực hiện sắp xếp lại mảng.

- Một mảng tăng dần nghĩa là ta luôn có $a[i] \leq a[i+1]$

Gán ok=1 (mảng là mảng tăng dần)

For i:=1 to n-1 do

Nếu $a[i] > a[i+1]$ thì ok=0;

Nếu ok=0 thì mảng không phải tăng dần

- Cách sắp xếp mảng theo kiểu tăng dần

For i =1 to n-1

For k=i+1 to n

Nếu $a[i] > a[k]$ thì

tam=a[i];

a[i] =a[k];

a[k] =tam;

| | |
|----------------------------|--|
| 6 | Cac phan tu cua mang: |
| -345 234 -5678 4321 11 789 | -345 234 -5678 4321 13 789 |
| | Trung binh cua mang la: 111 |
| | Trong mang co 1 so duong chan |
| | Phan tu nho nhat la -5678 nam o vi tri thu 3 |
| | Mang khong phai mang tang dan |
| | -5678 -345 11 234 789 4321 |

2. SẮP XẾP VÀ XÓA TRÙNG

Viết chương trình nhập mảng gồm N phần tử ($n < 10^6$) và thực hiện sắp xếp mảng tăng dần, sau đó xóa các phần tử trùng nhau trong mảng.

| Mang.inp | Mang.out |
|---------------------|---------------------|
| 10 | 2 4 4 4 4 5 7 8 8 8 |
| 4 7 5 4 8 4 8 2 4 8 | 2 4 5 7 8 |

3. Viết chương trình nhập vào một mảng số nguyên gồm n phần tử, sau đó nhập vào một số x và tìm xem x có nằm trong mảng hay không, nếu x đã tồn tại trong mảng thì cho biết các vị trí của x, nếu x không tồn tại trong mảng thì thêm x vào cuối mảng.

Hướng dẫn:

$v := n + 1;$

For từ 1 đến n

Nếu $a[i] = x$ thì gán $v := i;$

Nếu $v < n$ thì in ra đã tồn tại và vị trí v

Ngược lại thì tăng n lên 1 và cho $a[n] = x;$

Dữ liệu vào:

- Dòng đầu tiên gồm 2 số N và X
- Dòng thứ 2 là N số nguyên cách nhau 1 khoảng trắng

Dữ liệu ra: Vị trí của X trong mảng.

Ví dụ

| Timkiem.inp | Timkiem.out | Giải thích |
|--------------------|-------------|--|
| 6 8 5 3 6 8 1 4 | 4 | Giá trị 8 đã tồn tại và nằm ở vị trí thứ 4 |
| 6 9 5 3 6 8 1 4 | 7 | Giá trị 9 không tồn tại trong mảng nên phải thêm 9 vào mảng (vị trí 7 là vị trí sau khi thêm 9 vào mảng) |

4. MẢNG NGUYÊN TỐ

Viết chương trình nhập mảng có N phần tử và in ra số lượng và các số nguyên tố có trong mảng

| snt.in | snt.out |
|--------------------------------------|---------------------|
| 10 79 11 68 31 82 49 51 67 29 108 | 5 79 11 31 67 29 |

5. PHẦN TỬ LỚN NHÌ

Viết chương trình nhập vào một mảng a gồm n phần tử có giá trị khác nhau. Tìm phần tử có giá trị lớn thứ 2 trong mảng và vị trí của phần tử đó.

Hướng dẫn:

Khai báo biến lớn nhất max và biến lớn thứ 2 mid.

Cho $\text{max} := a[1]$;

Nếu $a[2] > \text{max}$ thì $\text{Max} := a[2]$; $\text{mid} := a[1]$;

Ngược lại thì $\text{mid} := a[2]$;

Nếu $a[i] > \text{max}$ thì $\text{max} = a[i]$; $\text{mid} = \text{max}$;

ngược lại nếu $a[i] > \text{mid}$ thì $\text{mid} = a[i]$;

Dữ liệu vào:

- Dòng đầu tiên là số N
- Dòng thứ 2 là N số nguyên cách nhau 1 khoảng trắng

Dữ liệu ra:

- Dòng 1: Giá trị lớn thứ nhì trong mảng
- Dòng 2: Vị trí của phần tử lớn thứ nhì

| Seconmax.inp | Seconmax.out |
|-----------------------|--------------|
| 10 | 9 |
| 5 10 6 8 1 4 9 7 10 2 | 7 |

6. MẢNG HAI CHỮ SỐ

Viết chương trình nhập vào mảng a có n phần tử, mỗi phần tử là một số có 2 chữ số. Hãy thực hiện các yêu cầu sau:

A) In ra các phần tử có phần chục không lớn hơn phần đơn vị.

B) In ra các phần tử thỏa mãn tính chất $a[i-1] < a[i] < a[i+1]$

Hướng dẫn: For i := 2 to n-1 do

C) In ra các phần tử của mảng nhưng bỏ qua những phần tử có giá trị nhỏ hơn phần tử trước nó.

Ví dụ:

| Twodig.inp | Twodig.out |
|-------------------------------|-------------------|
| 10 | 35 13 66 79 27 49 |
| 35 13 66 79 81 54 42 90 27 49 | 66 79 |
| | 35 66 79 81 90 |

7. CẶP SỐ

Viết chương trình nhập hai số nguyên N, K và dãy số A_1, A_2, \dots, A_n

Biết rằng: $0 < N < 255$; $0 < K, A_i < 65535$). Hãy tìm vị trí (i, j) của tất cả các cặp số (A_i, A_j) và số cặp (A_i, A_j) với $i < j$ thỏa mãn $A_i + A_j = K$.

Word;

- Dữ liệu vào: ghi vào tệp `D:\CAPSO.INP`

- Kết quả: ghi ra tệp `D:\CAPSO.OUT` (đưa ra số lượng và chỉ số của các cặp)

| DOI.INP | DOI.OUT |
|---------------|---------|
| 5 35 | 1 3 |
| 12 3 23 32 23 | 1 5 |
| | 2 4 |
| | 3 |

8. CẶP SỐ NGUYÊN TỐ

Cho một dãy gồm N số nguyên A_1, A_2, \dots, A_N . Hãy đếm xem trong dãy đã cho có bao nhiêu cặp số giống nhau và kiểm tra cặp số giống nhau đó có phải là số nguyên tố hay không.

Dữ liệu vào từ tập tin văn bản `CapSo.inp` gồm:

- Dòng đầu tiên là số N ($2 < N < 100$)
- Dòng tiếp theo là dãy gồm N số nguyên A_1, A_2, \dots, A_N

Kết quả xuất ra tập tin văn bản `CapSo.out` gồm:

- Dòng đầu tiên là số lượng cặp số giống nhau.
- Các dòng tiếp theo gồm: chỉ số của cặp số giống nhau, trong đó chỉ số bé đứng trước chỉ số lớn đứng sau. Tiếp theo là giá trị và kiểm tra số nguyên tố của cặp số giống nhau đó.

| CapSo.inp | CapSo.out |
|-------------------|-------------------------------|
| 9 | 6 |
| 3 6 4 3 2 4 2 4 6 | 1-4 : 3 là số nguyên tố |
| | 2-9 : 6 không là số nguyên tố |
| | 3-6 : 4 không là số nguyên tố |
| | 3-8 : 4 không là số nguyên tố |
| | 5-7 : 2 là số nguyên tố |
| | 6-8 : 4 không là số nguyên tố |

9. KHOẢNG CÁCH

Cho một số nguyên dương n và dãy A gồm n số nguyên dương A_1, A_2, \dots, A_n , mỗi phần tử A_i là tồn tại duy nhất trong dãy số ($1 \leq A_i \leq 10^9$ với $1 \leq i \leq n \leq 255$). Tìm trong dãy số các cặp số kế cận A_i và A_{i+1} mà có tổng các chữ số của chúng là nhỏ nhất. Giá trị nhỏ nhất đó là bao nhiêu? Có bao nhiêu cặp số thỏa mãn? Sắp xếp dãy số tăng dần, tìm khoảng cách giữa 2 phần tử của cặp số có tổng các chữ số nhỏ nhất trong dãy đã sắp xếp. (Chỉ cần tìm khoảng cách cặp có tổng chữ số nhỏ nhất tìm được đầu tiên trong dãy chưa sắp xếp).

Với dữ liệu vào từ tệp tin D:\DAYSO.INP

- + Dòng đầu tiên là số nguyên n
- + Dòng thứ 2 là dãy số nguyên dương A gồm A_1, A_2, \dots, A_n

Với dữ liệu ra ghi vào tệp D:\KETQUA.OUT

- + Dòng thứ nhất ghi giá trị tổng chữ số nhỏ nhất
- + Dòng thứ 2 ghi các cặp số có tổng chữ số nhỏ nhất
- + Dòng thứ 3 ghi số cặp có tổng chữ số nhỏ nhất đếm được
- + Dòng thứ tư ghi dãy số đã được sắp xếp
- + Dòng thứ năm ghi khoảng cách của cặp số có tổng các chữ số nhỏ nhất ở vị trí đầu tiên trong dãy ban đầu.

| D:\DAYSO.INP | GIẢI THÍCH |
|---|---|
| 10 10 4 6 1 5 34 2 3 12 20 | $n=10$ 10 phần tử của dãy số |
| D:\KETQUA.OUT | GIẢI THÍCH |
| 5 10 va 4; 2 va 3; 12 va 20 3 1 2 3 4 5 6 10 12 20 34 3 | Giá trị tổng chữ số nhỏ nhất là 5 Các cặp số có tổng chữ số nhỏ nhất Có 3 cặp có tổng chữ số nhỏ nhất Dãy số sau khi sắp xếp Cặp số tìm được đầu tiên là 10 và 4: 10 ở vị trí thứ 7, 4 ở vị trí thứ 4. Khoảng cách giữa 10 và 4 = $7 - 4 = 3$ |

10. ĐẾM CẶP

Cho dãy gồm N số nguyên dương a_1, a_2, \dots, a_n ($N \leq 10^6, a_i \leq 10^6$). Đếm số lượng cặp (hai phần tử) có tổng là số chẵn.

| demcap.inp | demcap.out |
|----------------|--|
| 5 1 4 5 3 2 | 4 (gồm các cặp $\{1,5\}, \{1,3\}, \{5,3\}, \{4,2\}$, |

11. MẬT MÃ

Nhân ngày sinh nhật, Minh được các bạn tặng quà chúc mừng, riêng phần quà của lớp trưởng thì Minh không mở được. Hôm sau, Minh nhận được thông tin từ lớp trưởng là cần có mật mã để mở phần quà. Mật mã là dãy liên tiếp được kết hợp từ số lượng phần tử của dãy số C sau đó là số lượng phần tử là số chẵn của dãy số C và cuối cùng là số lượng phần tử là số lẻ của dãy số C.

Biết rằng, dãy số C được tạo ra bằng cách tìm các phần tử xuất hiện đồng thời cả trong dãy số A gồm $a_1, a_2, a_3, \dots, a_n$ ($a_i < 10^5, 2 < n < 100$) và dãy số B gồm $b_1, b_2, b_3, \dots, b_m$ ($b_i < 10^5, 2 < m < 100$). Mỗi phần tử trong dãy C là duy nhất, không lặp lại và được sắp xếp theo thứ tự tăng dần. Bạn hãy giúp Minh tìm mật mã trên.

Dữ liệu vào từ tập tin văn bản *MatMa.inp* gồm:

- Dòng 1: chứa hai số n, m ($2 < n < 100, 2 < m < 100$).
- Dòng 2: dãy số nguyên A gồm $a_1, a_2, a_3, \dots, a_n$, mỗi phần tử cách nhau một khoảng trắng.
- Dòng 3: dãy số nguyên B gồm $b_1, b_2, b_3, \dots, b_m$, mỗi phần tử cách nhau một khoảng trắng.

Kết quả xuất ra tập tin văn bản *MatMa.out* gồm:

- Dòng 1: dãy số C được sắp xếp tăng dần, mỗi phần tử cách nhau một khoảng trắng.
- Dòng 2: số phần tử của C.
- Dòng 3: số phần tử là số chẵn của C.
- Dòng 4: số phần tử là số lẻ của C.
- Dòng Mật mã: Là các số dòng 2,3,4 viết liền nhau.

| MatMa.inp | MatMa.out |
|--------------------|--------------------|
| 7 14 | C: -1 3 12 |
| 7 3 12 -1 -14 3 -1 | Số phần tử: |
| 8 -1 3 12 3 5 | Số phần tử chẵn: 1 |
| | Số phần tử lẻ: 2 |
| | Mật mã: 312 |

12. TÌM LẶP

Cho dãy số $a_1, a_2, a_3, \dots, a_n$ ($a_i < 10^5$, $2 < n < 100$) và một số k . Hãy tìm ra các phần tử xuất hiện từ k lần trở lên trong dãy số trên. Cho biết có bao nhiêu phần tử như vậy và liệt kê các phần tử đó.

Dữ liệu vào từ tập tin văn bản **TimLap.inp** gồm:

- Dòng thứ nhất chứa số n ($2 < n < 100$) và k .
- Dòng tiếp theo chứa dãy gồm n số nguyên $a_1, a_2, a_3, \dots, a_n$.

Kết quả xuất ra tập tin văn bản **TimLap.out** gồm:

- Dòng thứ nhất là số lượng các phần tử xuất hiện từ k lần trở lên trong dãy số.
- Dòng tiếp theo chứa các phần tử xuất hiện từ hai lần trở lên trong dãy số (nếu có), mỗi phần tử cách nhau một khoảng trắng.

| TimLap.inp | TimLap.out |
|---------------------|------------|
| 10 2 | 3 |
| 6 9 3 2 5 9 2 6 9 2 | 6 9 2 |

13. BÁN BÁNH MÌ

Trong mùa dịch bệnh Covid-19 đang diễn biến phức tạp, Bình chủ một cửa hàng sản xuất và bán bánh mì đã chuyển từ hình thức kinh doanh thông thường sang hình thức kinh doanh online và giao hàng tận nơi, mỗi ngày cửa hàng của Bình sản xuất T ($1 \leq T \leq 50000$) ổ bánh mì. Đây là một hiệu bánh nổi tiếng, mỗi ngày có n ($1 \leq n \leq 100$) khách hàng đặt mua, mỗi khách hàng đặt mua a_i ổ bánh mì (với $1 \leq i \leq n$; $1 \leq a_i \leq 50000$) và số lượng bánh mà cửa hàng sản xuất ra không đủ bán. Để bán hết bánh đạt được doanh thu, vừa phòng chống dịch bệnh (hạn chế tiếp xúc nhiều người), Bình cần phải chọn những khách hàng nào để bán sao cho số lượng khách hàng mua được bánh là ít nhất?

Em hãy viết chương trình để giúp Bình giải quyết vấn đề trên, đồng thời thông báo mỗi ngày lượng bánh mì có đủ để bán không? Nếu đủ thì thông báo đủ ngược lại thông báo số lượng bánh còn thiếu và số người không mua được bánh.

- Với dữ liệu đầu vào từ file `d:\banhmi.inp` như sau: Hàng thứ nhất là số T, N ; hàng thứ hai: a_1, a_2, \dots, a_n .

- Dữ liệu đầu ra ghi vào file `d:\banhmi.out` như sau: Hàng thứ nhất ghi số lượng khách hàng mua được bánh và số bánh mì cửa hàng không cung cấp đủ cho khách; hàng thứ hai: số lượng khách hàng không mua được bánh (do cửa hàng hết bánh) và tổng số bánh mì mà họ đặt mua.

| D:\banhmi.inp | D:\banhmi.out |
|-------------------|---------------|
| 60 6 | 3 15 |
| 10 30 20 17 18 25 | 3 45 |

14. DÃY SỐ HẠNH PHÚC

Dãy số tự nhiên $a_1, a_2, a_3, \dots, a_n$ ($a_i < 10^5, 2 \leq n \leq 30$) được gọi là hạnh phúc nếu thoả mãn các điều kiện sau:

- Dãy trên là một dãy giảm dần.
- Với mọi i ($2 \leq i \leq n$) a_i hoặc là số nguyên tố, hoặc phải là ước của một trong các số a_1, a_2, \dots, a_{i-1} .

Dựa vào gợi ý trên em hãy viết chương trình kiểm tra một dãy số có phải là dãy hạnh phúc không? Nếu không thì biến đổi dãy số này (sắp xếp trước và xóa phần tử nếu cần) xem có được dãy hạnh phúc hay không?

Với dữ liệu vào từ tệp tin D:\DSHP.INP

+ Dòng đầu tiên là số nguyên n

+ Dòng thứ hai là dãy số nguyên dương a gồm a_1, a_2, \dots, a_n

Với dữ liệu ra ghi vào tệp D:\DSHP.OUT theo hướng dẫn sau:

| DSHP.INP | DSHP.OUT |
|----------------------------|---------------------------------------|
| 7 24 23 17 13 11 8 3 | La day so hanh phuc |
| 8 4 13 17 16 28 7 15 26 | 5 28 17 13 7 4 la day so hanh phuc |
| 5 30 28 16 4 18 | Khong tim duoc day so hanh phuc |

15. SỐ DƯƠNG LIÊN TIẾP

Viết chương trình nhập vào mảng a có n phần tử ($n \leq 10^3, -10^9 < a_i < 10^9$). Cho biết

- A) Số lượng các số hạng dương liên tiếp nhiều nhất.
- B) Tổng các số hạng dương liên tiếp lớn nhất.
- C) Tổng tìm được ở câu B có phải là tổng của các số ở câu A?

| PartPI.inp | PartPI.out |
|---|---------------|
| 20 1 3 5 -4 -5 -6 -2 -4 6 9 5 -5 -8 -7 9 1 2 2 -3 2 | 4 20 NO |

16. Viết chương trình nhập vào mảng a có n phần tử ($n \leq 10^3$), mỗi phần tử a_i thỏa mãn $0 < a_i \leq 10$). In ra:

A) Số lần xuất hiện của các phần tử trong mảng.

Hướng dẫn:

Khai báo mảng B gồm 10 phần tử, cho mỗi phần tử $=0$

For từ 1 đến n

$B[a[i]] = B[a[i]] + 1;$

B) Các phần tử xuất hiện nhiều nhất trong mảng và số lần xuất hiện của nó.

Hướng dẫn:

- Tạo một mảng C với số phần tử k . Khai báo biến max cho biết số lần xuất hiện nhiều nhất.

- Khi có một $a[i]$ có số lần xuất hiện bằng với max thì tăng k lên một đơn vị và gán $C[k] = a[i]$. Nếu số lần xuất hiện của $a[i]$ lớn hơn max thì gán lại $k=1$ và cho $C[k] = a[i]$.

C) Xét xem mảng đã cho có phải là một mảng đồng đều hay không.

Một mảng được gọi là đồng đều khi số lượng mỗi phần tử không trùng nhau của mảng chênh lệch nhau không quá $n/10$ phần tử. Chẳng hạn mảng gồm các phần tử 5 6 7 5 6 8 5 6 7 8, mảng này có $n=10$ phần tử và số lần xuất hiện như sau:

5: 3

6: 3

7: 2

8: 2

Với $n/10=1$, ta thấy số lượng các phần tử lệch nhau không quá 1, nên mảng là đồng đều.

| Dongdeu.inp | Dongdeu.out |
|-----------------|-------------------------------------|
| 8 | 1: 2 |
| 1 2 2 1 3 9 3 2 | 2: 3 |
| | 3: 2 |
| | 9: 1 |
| | Cac phan tu xuat hien nhieu nhat: 2 |
| | So lan xuat hien: 3 |
| | Mang khong dong deu. |

17. GIAO CỦA 2 TẬP HỢP

Cho mảng A và B đều gồm N ($N \leq 10^5$) phần tử nguyên dương ($A_i, B_i \leq 10^9$). Yêu cầu bạn hãy tạo ra một mảng C, gồm các số có ở mảng A và phải có ở mảng B, sao cho với mỗi số ở mảng C chỉ xuất hiện duy nhất một lần.
Dữ liệu vào :

Dòng đầu : Một số nguyên duy nhất N

Dòng hai : Gồm N số nguyên dương của mảng A cách nhau bởi dấu cách.

Dòng ba : Gồm N số nguyên dương của mảng B cách nhau bởi dấu cách.

Dữ liệu ra :

Dòng đầu: Độ dài của mảng C

Dòng hai: Các phần tử của mảng C

| Giao.inp | Giao.out |
|-----------|----------|
| 5 | 2 |
| 1 5 3 7 9 | 1 3 |
| 1 1 3 2 4 | |

18. DÂY TUYẾN TÍNH

Một dãy tuyến tính là 1 bộ 3 có thứ tự (s_1, s_2, s_3) thỏa mãn $s_2 - s_1 = s_3 - s_2$

Ví dụ: (1,2,3), (2,4,6), (14,21,28) là các dãy tuyến tính

Cho trước S ($3 \leq S \leq 30$) số nguyên tăng thuộc đoạn [1,100], đếm số các dãy tuyến tính

Dữ liệu: Cho trong tập tin văn bản LSEQ.INP, dòng đầu là số nguyên S, dòng tiếp theo là S số nguyên, cách nhau ít nhất 1 khoảng trắng

Kết quả: Cho trong tập tin văn bản LSEQ.OUT, chứa số nguyên duy nhất chỉ số dãy tuyến tính tìm được.

| LSEQ.INP | LSEQ.OUT | Giải thích |
|--------------------|----------|--|
| 7 1 2 3 4 6 8 9 | 5 | 5 dãy tuyến tính tìm được là: 1 2 3 2 3 4 2 4 6 3 6 9 4 6 8 |

19. HÀNG XÓM

Nhập dãy N số nguyên và một số nguyên k. Xét xem trong dãy có k số dương đứng cạnh nhau hay không?

Dữ liệu nhập: gồm 2 số N và k

Dữ liệu xuất: Trả lời có hay không.

| hangxom.inp | hangxom.out |
|-------------------------------------|-------------|
| 12 3 -1 2 3 4 -2 -4 4 5 -3 4 5 6 | YES |

20. TỔNG 2 SỐ

Nhập dãy N số nguyên. Tìm các số của dãy bằng tổng 2 số khác trong dãy.

| Tong2so.inp | Tong2so.out |
|--------------|-------------|
| 4 6 9 4 2 | 6 |

21. DÃY KHÁCH

Cho dãy số nguyên. Gọi M và m lần lượt là các giá trị lớn nhất và bé nhất của dãy số nguyên. Hãy đưa ra các số nguyên theo thứ tự tăng dần thuộc đoạn [m.M] mà không xuất hiện trong dãy.

Dữ liệu nhập: Cho trong file MINMAX.INP gồm 2 dòng;

- Dòng 1: Số phần tử của dãy n.
- Dòng 2: Dãy số nguyên, giữa các số cách nhau 1 khoảng trắng.

Kết quả: Ghi ra file MINMAX.OUT 1 dòng duy nhất là dãy tìm được.

Ví dụ:

| MINMAX.INP | MINMAX.OUT |
|----------------|------------|
| 5 9 1 5 3 6 | 2 4 7 8 |

Câu 2: Số có ước nhiều nhất

Cho dãy có N số nguyên ($0 < N < 10^{12}$). Viết chương trình tìm số có số ước nhiều nhất trong các số đó.

Dữ liệu vào: file UOCSO.INP

Dòng đầu tiên ghi số nguyên dương N ;

Dòng tiếp theo ghi N số nguyên dương, mỗi số cách nhau một khoảng trống.

Kết quả: ghi file UOCSO.OUT

Dòng đầu tiên ghi số tìm được, nếu có nhiều số có cùng ước số nhiều nhất thì ghi mỗi số trên một hàng theo thứ tự từ số nhỏ đến số lớn, nếu không tìm được số nào thì ghi chữ KHÔNG.

Ví dụ:

| UOCSO.INP | UOCSO.OUT |
|--------------------------|-----------|
| 10 | 20 |
| 7 2 28 5 8 4 20 10 12 15 | |

BÀI 3. CHẠY TIẾP SỨC

Trong kỳ thi hội khỏe phù đồng cấp huyện vừa qua có n vận động viên tham gia chạy ngắn, các vận động viên được đánh số báo danh từ 1 đến n . Kết quả thời gian chạy của vận động viên thứ i là t_i (miligiây). Để chuẩn bị cho môn chạy tiếp sức trong kỳ thi hội khỏe phù đồng cấp tỉnh, ban huấn luyện muốn chọn ra 4 vận động viên chạy nhanh nhất trong số n vận động viên này.

Yêu cầu: Hãy cho biết số báo danh của 4 vận động viên được chọn. Biết rằng không có hai vận động viên nào có cùng thành tích.

Dữ liệu vào: Cho từ tệp văn bản có tên BL3.INP có dạng như sau:

- Dòng đầu tiên ghi số nguyên n ($4 \leq n \leq 1000$)

- n dòng tiếp theo, dòng thứ i ghi số nguyên dương t_i là thời gian chạy của vận động viên thứ i ($t_i \leq 30000, i=1..n$)

Kết quả: Ghi ra tệp văn bản có tên BL3.OUT gồm một dòng ghi 4 số nguyên tương ứng là 4 số báo danh của 4 vận động viên được chọn (không cần ghi theo thứ tự). Các số trên cùng một dòng ghi cách nhau ít nhất một dấu cách.

Ví dụ:

| BL3.INP | BL3.OUT |
|---------|---------|
| 7 | 2 3 5 7 |
| 13000 | |
| 9800 | |
| 10200 | |
| 11500 | |
| 10060 | |
| 12050 | |
| 11200 | |

BÀI 4. (5 điểm) DÃY TĂNG

Một dãy số a_1, a_2, \dots, a_n được gọi là dãy tăng nếu như nó thỏa điều kiện:

$$a_1 < a_2 < \dots < a_n$$

Yêu cầu: Cho dãy số nguyên a_1, a_2, \dots, a_n . Hãy cho biết dãy số này có phải là dãy tăng hay không? Nếu không phải dãy tăng thì hãy cho biết chỉ số k nhỏ nhất của phần tử làm cho dãy không có tính chất của dãy tăng.

Dữ liệu vào: Cho từ tệp văn bản có tên **BL4.INP** có dạng như sau:

- Dòng đầu tiên ghi số nguyên n ($1 < n \leq 1000$)
- Dòng thứ hai ghi dãy n số nguyên a_1, a_2, \dots, a_n ($-1000 \leq a_i \leq 1000, i=1..n$)

Kết quả: Ghi ra tệp văn bản có tên **BL4.OUT** gồm một số nguyên duy nhất k . Trường hợp dãy đã cho là dãy tăng thì ghi số 0.

Ví dụ 1:

| BL4.INP | BL4.OUT |
|----------------------|---------|
| 6 2 7 10 18 25 41 | 0 |

Ví dụ 2:

| BL4.INP | BL4.OUT |
|---------------------------|---------|
| 8 3 6 10 7 15 20 18 12 | 4 |

BÀI 3. (7 điểm) XẾP HÀNG HÓA

Tại một bến cảng, các công nhân đang bốc dỡ các kiện hàng từ tàu biển lên các xe container. Các kiện hàng phải được bốc dỡ **lần lượt** từ kiện hàng thứ nhất đến kiện hàng cuối cùng. Mỗi kiện hàng khi bốc lên phải được đặt ngay vào xe container đang chờ sẵn để vận chuyển đi, xe này đầy thì đến lượt xe kế tiếp. Mỗi xe container chỉ có thể chở hàng hóa có tải trọng không quá **M**.

Yêu cầu: Hãy cho biết cần ít nhất bao nhiêu chuyến xe để vận chuyển hết hàng hóa trên tàu.

Dữ liệu vào: Cho từ tệp văn bản có tên **BL3.INP** có dạng như sau:

- Dòng đầu tiên ghi hai số nguyên **n** và **M** ($1 \leq n \leq 10^3$, $1 \leq M \leq 10^6$) tương ứng là số kiện hàng trên tàu và tải trọng tối đa của xe container.
- Dòng thứ hai ghi n số nguyên **a₁, a₂, ..., a_n** ($1 \leq a_i \leq M$) trong đó **a_i** tương ứng là trọng lượng của kiện hàng thứ **i** ($i=1..n$).

Kết quả: Ghi ra tệp văn bản có tên **BL3.OUT** gồm một dòng ghi một số nguyên là số chuyến xe ít nhất.

Các số trên cùng một dòng ghi cách nhau ít nhất một dấu cách.

Ví dụ:

| BL4.INP | BL4.OUT |
|-------------------------------------|----------------|
| 10 15 <u>7 6 8 8 7 5 2 4 3 9</u> | 5 |

Giải thích: Cần 5 chuyến xe

- Chuyến 1 chở kiện hàng 1 và 2 → trọng lượng 13
- Chuyến 2 chở kiện hàng 3 → trọng lượng 8
- Chuyến 3 chở kiện hàng 4 và 5 → trọng lượng 15
- Chuyến 4 chở kiện hàng 6, 7, 8, 9 → trọng lượng 14
- Chuyến 5 chở kiện hàng 10 → trọng lượng 9

Câu 4 (4,0 điểm)

Cho mảng số nguyên A gồm các phần tử $a_1, a_2, a_3, \dots, a_m$ với $(0 \leq a_i \leq 9; 1 \leq m \leq 1000)$; mảng B gồm các phần tử $b_1, b_2, b_3, \dots, b_n$ với $(0 \leq b_i \leq 9; 1 \leq n \leq 1000)$, các phần tử của mảng A và B có thể giống nhau. Sử dụng ngôn ngữ lập trình Pascal viết chương trình thực hiện:

a. Tìm số nguyên C nhỏ nhất được ghép từ tất cả các phần tử của mảng A. Biết số có số không (0) đứng đầu không được gọi là số nguyên C. Nếu mảng A chỉ gồm toàn số 0 thì $C=0$.

b. Tìm số nguyên D lớn nhất được ghép từ tất cả các phần tử khác nhau lấy từ mảng A và B (các chữ số của D khác nhau).

* **Dữ liệu vào:** Đọc từ tệp văn bản CAU4.INP

- Dòng 1: số phần của mảng A;
- Dòng 2: số phần tử của mảng B;
- Dòng 3: giá trị các phần tử của mảng A mỗi phần tử cách nhau một kí tự trống
- Dòng 4: giá trị các phần tử của mảng B mỗi phần tử cách nhau một kí tự trống

* **Kết quả:** Ghi ra tệp văn bản CAU4.OUT với cấu trúc như sau:

- Dòng 1: số nguyên C
- Dòng 2: số nguyên D

| CAU4.INP | CAU4.OUT |
|-----------------------|-----------|
| 8 | 20034557 |
| 11 | 987654320 |
| 2 5 7 4 0 5 3 0 | |
| 2 3 3 9 8 6 5 7 9 5 5 | |

Bài 2: (7,0 điểm) CÁC THÍ SINH THÔNG MINH

Cuộc thi “Các thí sinh thông minh” là một cuộc thi rất nổi tiếng trên truyền hình. Để chọn các thí sinh thông minh nhất, cuộc thi có nhiều vòng thi, thang điểm mỗi vòng thi là 20 điểm. Năm nay có n thí sinh dự thi, hiện tại đang chuẩn bị bước vào vòng thi cuối cùng và tổng điểm số của từng thí sinh ở các vòng thi trước lần lượt là a_1, a_2, \dots, a_n . Tổng điểm chung cuộc của các thí sinh sẽ bằng điểm số của họ đạt được trong vòng thi này cộng với điểm số đã có ở các vòng thi trước.

Yêu cầu: Cho trước tổng số điểm của từng thí sinh ở các vòng thi trước. Hãy tính xem có bao nhiêu thí sinh có khả năng vô địch. Biết rằng thí sinh vô địch là thí sinh đạt tổng số điểm cao nhất của cuộc thi.

Dữ liệu vào: Cho từ tệp TM.INP gồm hai dòng:

- Dòng thứ nhất ghi số nguyên n là số thí sinh dự thi ($1 < n \leq 10^5$)
- Dòng thứ hai ghi n số nguyên a_1, a_2, \dots, a_n là tổng điểm số ở các vòng thi trước của từng thí sinh, thí sinh thứ i có tổng điểm số là a_i ($1 < a_i < 10^5$, $i=1..n$).

Các số ghi trên cùng một dòng cách nhau một kí tự trắng.

Kết quả: Ghi vào tệp văn bản TM.OUT gồm một dòng ghi số nguyên là số lượng thí sinh có khả năng vô địch.

Ví dụ:

| TM.INP | TM.OUT |
|------------------|--------|
| 4 40 45 15 45 | 3 |

Bài 3: (8,0 điểm) XÉT HẠNG

Vào dịp tổng kết năm học, nhà trường tổ chức khen thưởng cho các học sinh đạt hạng Nhất, Nhì, Ba trong toàn trường với những phần thưởng đặc biệt và muốn biết điểm số có phá kỉ lục điểm của năm học trước hay không. Số học sinh được các lớp đề nghị khen thưởng rất nhiều nên nhà trường cần xét chọn lại.

Việc xét hạng theo nguyên tắc: Các học sinh bằng điểm thì đồng hạng và không có thứ hạng kế tiếp. Chẳng hạn: Có 3 học sinh đồng hạng Nhất thì không có học sinh hạng Nhì, hạng Ba. Có 2 học sinh đồng hạng Nhì thì không có học sinh hạng Ba.

- **Yêu cầu:** Cho trước kỉ lục điểm năm học trước là K , số lượng học sinh đề nghị khen thưởng là N , các điểm số tương ứng là: a_1, a_2, \dots, a_N .

a) Xét xem điểm số của học sinh có phá kỉ lục điểm năm học trước hay không? Nếu phá kỉ lục điểm thì tính số điểm vượt kỉ lục, nếu không phá kỉ lục điểm thì ghi số 0.

b) Thống kê số lượng học sinh đạt hạng Nhất, số lượng học sinh đạt hạng Nhì, số lượng học sinh đạt hạng Ba và điểm số tương ứng của thứ hạng đó.

- **Dữ liệu vào:** Từ tệp văn bản XETHANG.INP gồm 2 dòng:

+ Dòng đầu tiên ghi số nguyên N và số thực K ($3 \leq N \leq 100, 0 < K \leq 10.0$).

+ Dòng tiếp theo ghi các số thực a_1, a_2, \dots, a_N ($0 < a_i \leq 10.0, 1 \leq i \leq N$).

Các số trên một dòng cách nhau một kí tự trắng.

- **Kết quả:** Ghi vào tệp văn bản XETHANG.OUT gồm:

+ Dòng thứ nhất ghi kết quả của câu a.

+ Dòng thứ hai ghi số lượng học sinh đạt hạng Nhất và điểm số của hạng Nhất.

+ Dòng thứ ba ghi số lượng học sinh đạt hạng Nhì và điểm số của hạng Nhì.

+ Dòng thứ tư ghi số lượng học sinh đạt hạng Ba và điểm số của hạng Ba.

Các số trên cùng một dòng cách nhau một kí tự trắng, các điểm số lấy một chữ số thập phân.

- **Ví dụ:**

| XETHANG.INP | XETHANG.OUT |
|---------------------|-------------|
| 5 9.8 | 0.1 |
| 9.6 9.8 9.6 9.8 9.9 | 1 9.9 |
| | 2 9.8 |
| | 0 |

| XETHANG.INP | XETHANG.OUT |
|-------------------------|-------------|
| 6 9.9 | 0 |
| 9.6 9.8 9.6 9.8 9.6 9.5 | 2 9.8 |
| | 0 |
| | 3 9.6 |