Chương 5
Mảng dữ liệu

- Khai báo và sử dụng
- Truyền mảng cho hàm
- Kỹ thuật xử lý cơ bản

ĐẶT VẤN ĐỀ

khai báo 3 số nguyên:

```
int sn1, sn2, sn3;
```

khai báo 10 số nguyên:

```
int sn1, sn2, sn3, sn4, sn5, sn6, sn7, sn8, sn9, sn10;
```

khai báo 51 số nguyên, thêm 51 số nguyên nữa: ...
(quản lý số tín chỉ đã đăng ký, stc đã đạt của 51 sv vlth)

=> khai báo 2 dãy số nguyên n số

Giải pháp:
(trong hầu hết các ngôn ngữ lập trình cấp cao)
sử dụng khai báo dãy (mảng)

===>mảng là gì? khai báo sử dụng ra sao?

KHÁI NIỆM

Mảng là kiểu dữ liệu có cấu trúc
chứa tập các phần tử có cùng kiểu.

//mỗi phần tử mảng thỏa tính chất của một biến của kiểu được khai báo:
+ Có thể thay đổi giá trị (gán lại)
+ Có thể dùng trong biểu thức.

Mảng một chiều biểu diễn dãy của nhiều phần tử cùng kiểu.

==========

Thảo luận

Cú pháp khai báo mảng?
Cú pháp truy xuất các phần tử trong mảng?

KHÁI NIỆM

Khái báo mảng

```
1  int mang1d[10]; //khai báo 10 phần tử kiểu nguyên
2
3  Phân biệt các cách sử dụng '[1_so_nguyen]' đối với mảng.
4
5  int a[10]; //khai báo mảng a có 10 phần tử
6  int x,y;
7
8  a[0];// truy xuất phần tử “0” (thứ nhất) của mảng a
9  a[4];// truy xuất phần tử “4” (thứ năm) của mảng a
10 a[9];// truy xuất phần tử “9” (cuối) của mảng a
1  ...
2  x = 2; y =3;
3  a[x+y];//truy xuất phần tử “giá trị biểu thức” của mảng a
4  //giá trị biểu thức x+y là 5 => truy xuất phần tử “5” của mảng a
5
6  ghi chú: với 0<= i <  số phần tử của mảng a thì
7  a[i]; truy xuất phần tử “i”của mảng a
8
9
20
```

KHÁI NIỆM

Khái báo mảng

```
1   #define ssvk13 51
2
3   //int tcdk[51]; //khai báo 51 phần tử kiểu nguyên
4   //int tctl[51];
5   int tcdk[ssvk13]; //khai báo 51 phần tử kiểu nguyên
6   int tctl[ssvk13];
7
8
9
10
1
2
3
4
5
6
7
8
9
20
```

KHÁI NIỆM

Khởi tạo giá trị cho mảng

```
1
2
3
4
5
6
7
8
9
10
1
2
3
4
5
6
7
8
9
20
```

```
int manga[5] = {1, 4, 6, 3, 2};
```

Chú giải

Khai báo **manga** có 5 phần tử

Khởi gán giá trị cho 5 phần tử lần lượt là

```
manga[0] = 1
manga[1] = 4
manga[2] = 6
manga[3] = 3
manga[4] = 2
```

KHÁI NIỆM

Khởi tạo giá trị cho mảng

```
1   //      int manga[5] = {1, 4, 6, 3, 2};
2           int mangb[5] = {9, 6, 1};
3
4
5
6
7
8
9
10
1
2
3
4
5
6
7
8
9
20
```

Chú giải

Khai báo **mangb** có 5 phần tử

Khởi gán giá trị cho 5 phần tử lần lượt là

mangb[0] = 9
mangb[1] = 6
mangb[2] = 1
mangb[3] = 0
mangb[4] = 0

KHÁI NIỆM

Khởi tạo giá trị cho mảng

```
1   //          int manga[5] = {1, 4, 6, 3, 2};
2   //          int mangb[5] = {9, 6, 1};
3               int mangc[5] = {0};
4
5
6   Chú giải
7
8
9       Khai báo mangc có 5 phần tử
10
1       Khởi gán giá trị cho 5 phần tử lần lượt là
2
3
4               mangc[0] = 0
5               mangc[1] = 0
6               mangc[2] = 0
7               mangc[3] = 0
8               mangc[4] = 0
9
20
```

KHÁI NIỆM

Khởi tạo giá trị cho mảng

```
1   //      int manga[5] = {1, 4, 6, 3, 2};
2   //      int mangb[5] = {9, 6, 1};
3   //      int mangc[5] = {0};
4           int mangd[] = {4, 19, 10};
5
6
7
8
9
10
1
2
3
4
5
6
7
8
9
20
```

Chú giải

Khai báo **mangd**

Khởi gán giá trị cho 3 phần tử lần lượt là

mangd[0] = 4
mangd[1] = 19
mangd[2] = 10

Số phần tử của **mangd** được tự động cập nhật là 3

MINH HỌA

/*mang.c*/

```
1    #include <stdio.h>
2
3    void main()
4    {          #define ssvk13 51
5               int tctl[ssvk13];
6               int tcdk[ssvk13] = {88,88,88,50};//cac phan tu khac bang 0
7               int dotuoi[] = {20,21,22};
8               int i, tam;
9
10              tctl[0] = 84; tctl[1] = 85; tctl[2] = 88; tctl[3] = 44;
1     //cac phan tu khac co the nhan gia tri nguyen bat ky
2               tam = sizeof(tctl)/sizeof(tctl[0]);
3               printf("\n do dai toi da cua mang tctl la: %d\n",tam);
4               tam = sizeof(dotuoi)/sizeof(dotuoi[0]);
5               printf("\n do dai toi da cua mang dotuoi la: %d\n",tam);
6    for (i=0;i<10;i++){
7                         printf("tcdk[%d] = %d\n",i,tcdk[i]);
8                         printf("tctl[%d] = %d\n",i,tctl[i]);
9               }
20   }
```

MINH HỌA

/*mang.c*/

```
1     do dai toi da cua mang tctl la: 51
2
3     do dai toi da cua mang dotuoi la: 3
4     tcdk[0] = 88
5     tctl[0] = 84
6     tcdk[1] = 88
7     tctl[1] = 85
8     tcdk[2] = 88
9     tctl[2] = 88
10    tcdk[3] = 50
1     tctl[3] = 44
2     tcdk[4] = 0
3     tctl[4] = 1
4     tcdk[5] = 0
5     tctl[5] = 32767
6     tcdk[6] = 0
7     tctl[6] = -1390276248
8     tcdk[7] = 0
9     tctl[7] = 54
20    tcdk[8] = 0
21    tctl[8] = 0
22    tcdk[9] = 0
23    tctl[9] = 0
```

TRUYỀN MẢNG CHO HÀM

```
1   //khai bao thu vien
2   void nhapmang(int a[51], int *n);//tham kiểu mảng có tối đa 51 phần tử
3   void xuatmang(int a[], int n);//có thể bỏ chỉ số "số phần tử tối đa"
4
5   void main()
6   {
7           int a[51], n;
8   // các xử lý bên trong hàm có thể làm thay đổi nội dung của mảng
9           nhapmang(a, &n);
10          xuatmang(a, n);
1   }
2   //dinh nghia ham nhapmang(int a[51], int *n) {...}
3   //dinh nghia ham xuatmang(int a[], int n) {...}
4
5
6
7
8
9
20
```

TRUYỀN MẢNG CHO HÀM

```
1   //khai bao thu vien
2   void nhapmang(int a[51], int *n);//tham kiểu mảng có tối đa 51 phần tử
3   void xuatmang(int a[], int n);//có thể bỏ chỉ số "số phần tử tối đa"
4
5
6
7
8   Trong thực tế, số lượng phần tử cần dùng không cố định. Do đó, khi viết
9   chương trình, người lập trình cần ước lượng số phần tử tối đa đủ lớn ứng với
10  nhu cầu tính toán, xử lý đang xét.
1
2
3   Số phần tử thực sự n của mảng không được lớn hơn số phần tử tối đa N
4
5
6
7
8
9
20
```

SV LÀM THEO NHÓM

Cài đặt chương trình nhập vào 1 mảng, thực hiện

1
2
3
4
5
6
7
8
9
10
1
2
3
4
5
6
7
8
9
20

+ Xuất số các số nguyên tố
+ Xuất số các số chẵn, lẻ
+ Sắp xếp không giảm, xuất mảng sau khi sắp xếp
+ Sắp xếp không tăng, xuất mảng sau khi sắp xếp

SV LÀM THEO NHÓM

Cài đặt chương trình nhập vào 1 mảng, thực hiện

+ Xuất số các số nguyên tố
+ Xuất số các số chẵn, lẻ
+ Sắp xếp không giảm, xuất mảng sau khi sắp xếp
+ Sắp xếp không tăng, xuất mảng sau khi sắp xếp

Chia nhóm như sau:

0, nhập xuất mảng, chép mảng (gv)
1, a) hàm kiem so nguyên tố ?
   b) hoan vi 2 số
2, a) sosanh 2 số là tăng, giảm, bằng ?
   b) 1 số là chẵn, lẻ ?
3, a) đếm số chẵn, lẻ trong mảng,
   b) đếm số nguyên tố trong mảng, có bao nhiêu hay ko có?
4, Sắp xếp mảng không giảm
5, Sắp xếp mảng không tăng
6, Cài đặt chương trình

GIỚI THIỆU

Mảng 2D biểu diễn cho một ma trận ***m* x *n***

Trong đó, các phần tử có cùng kiểu.

m: biểu diễn số dòng,

n: biểu diễn số cột,

KHAI BÁO

==========

<kiểu_cơ_sở> <tên_biến>[<N1>][<N2>];

==========

N1,N2:Số lượng phần tử mỗi chiều

ví dụ

```
int mang2d1[5][5];
int mang2d2[2][5];
int mang2d3[5][2];
```

==========

Truy xuất

mang2d1[0][0]; mang2d1[1][4]; mang2d2[0][2]; mang2d2[1][4];
mang2d2[0][0]; mang2d3[1][1]; mang2d3[4][1]; mang2d3[0][0];

KHAI BÁO

==========

typedef <kiểu_cơ_sở> <tên_kiểu_mới>[<N1>][<N2>];

==========

N1,N2:Số lượng phần tử mỗi chiều

ví dụ

```
typedef int mang5x5[5][5];
typedef int mang10x5[10][5];
mang5x5 a; mang10x5 b;
```

==========

Truy xuất

a[0][0]; a[1][4]; a[4][4];
b[0][0]; b[1][2]; b[9][4];

GÁN DỮ LIỆU

```
1   #include <stdio.h>
2   #define N 5
3   typedef int mang2d[N][N];
4
5   void main()
6   {       mang2d a, b;
7           int i, j;
8
9           for (i=0;i<5;i++) {
10              for (j=0;j<5;j++) {
1                   a[i][j] = i*i + j*j;
2               }
3           }
4   //Gan 2 mang: bat buoc gan tung phan tu.
5           for (i=0;i<5;i++) {
6               for (j=0;j<5;j++) {
7                   b[i][j] = a[i][j];
8             }
9           }
20  }
```

ĐỐI SỐ CỦA HÀM

Địa chỉ phần tử đầu tiên của mảng được truyền cho hàm

Mảng có thể thay đổi sau khi truyền cho hàm

Khai báo đối số mảng như khái báo mảng

==========

```
#include <stdio.h>
#define N 5
typedef int mang2d[N][N];
void nhapmang2d(mang2d a);

void main()
{
    mang2d b;
    nhapmang2d(b);
}

void nhapmang2d(mang2d a) {
//cac lenh tuong ung
}
```

ĐỐI SỐ CỦA HÀM

Địa chỉ phần tử đầu tiên của mảng được truyền cho hàm

Mảng có thể thay đổi sau khi truyền cho hàm

Khai báo đối số mảng như khái báo mảng

```
==================
#include <stdio.h>
#define N 5
typedef int mang2d[N][N];
void nhapmang2d(mang2d a);

void main()
{
    mang2d b;
    nhapmang2d(b);
}

void nhapmang2d(int a[N][N]) {
//cac lenh tuong ung
}
```

ĐỐI SỐ CỦA HÀM

Địa chỉ phần tử đầu tiên của mảng được truyền cho hàm

```
1   Mảng có thể thay đổi sau khi truyền cho hàm
2
3   Khai báo đối số mảng như khái báo mảng
4
5   Có thể bỏ chỉ số phần tử thứ 2
6   ================
7   #include <stdio.h>
8   #define N 5
9   typedef int mang2d[N][N];
10  //void nhapmang2d(mang2d a);
1   void nhapmang2d(int a[][N]);
2   void main()
3   {
4       mang2d b;
5       nhapmang2d(b);
6   }
7
8   void nhapmang2d(int a[N][N]) {
9   //cac lenh tuong ung
20  }
```

ĐỐI SỐ CỦA HÀM

Địa chỉ phần tử đầu tiên của mảng được truyền cho hàm

Mảng có thể thay đổi sau khi truyền cho hàm

Khai báo đối số mảng như khái báo mảng

Có thể bỏ chỉ số phần tử thứ 2

==========

```
#include <stdio.h>
#define N 5
typedef int mang2d[N][N];
//void nhapmang2d(mang2d a);
void nhapmang2d(int a[][N]);
void main()
{
    mang2d b;
    nhapmang2d(b);
}

void nhapmang2d(mang2d a) {
//cac lenh tuong ung
}
```

ĐỐI SỐ CỦA HÀM

Viết chương trình xử lý ma trận A(mxn) theo từng yêu cầu sau

1) Chép dòng k vào dòng h

```
void chepDong(int A[][N], int m, int n, int k, int h);
```

2) Chép cột k vào cột h

```
void chepcot(int A[][N], int m, int n, int k, int h);
```

3) Hoán đổi dòng k và h

```
void hoandoiDong(int A[][N], int m, int n, int k, int h);
```

4) Hoán đổi cột k và h

```
void hoandoicot(int A[][N], int m, int n, int k, int h);
```

==========

Trong đó, m ≤ M, n ≤ N