

NỘI DUNG CHÍNH BUỔI 12

CHUỖI

(1) Khái niệm

- Chuỗi được xem như một mảng 1 chiều gồm các phần tử kiểu **char**.
- Ký hiệu kết thúc chuỗi được quy ước là ký tự `'\0'` (ký tự có mã ASCII là 0) đặt ở cuối chuỗi
=> Một mảng ký tự gồm n phần tử sẽ lưu được tối đa n - 1 ký tự.

(2) Khai báo chuỗi

- Cú pháp: **char** <tên_chuỗi>[<kích_thước>];

VD: `char s[10];` //Khai báo một chuỗi s chứa tối đa 9 ký tự.

(3) Khởi tạo chuỗi

- Khi khởi tạo chuỗi có thể chỉ ra hoặc không chỉ ra kích thước chuỗi.
- Có thể dùng một biến con trỏ kiểu **char** để khởi tạo một chuỗi.

VD1: Chỉ ra kích thước khi khởi tạo chuỗi: `char s[10]="Hello";`

VD2: Không chỉ ra kích thước khi khởi tạo chuỗi: `char s[]="Hello";`

VD3: Dùng biến con trỏ kiểu char để khởi tạo chuỗi: `char *p="Hello";`

(4) Hàm nhập chuỗi

- Cú pháp: **gets**(tên_chuỗi);

VD: `gets(s);` //Nhập 1 chuỗi ký tự s có thể có khoảng trắng

- Chú ý: Hàm **scanf** có để lại ký tự `'\n'` trên **stdin**, ký tự này sẽ làm trôi hàm **gets** (nếu có) sau đó. Để hàm **gets** không bị trôi thì phải khử ký tự `'\n'` trong stdin trước khi gọi hàm **gets** bằng cách dùng hàm **fflush(stdin)** hoặc **flushall()** để làm sạch **stdin**.

```
void main()
{
    int tuoi;
    char ten[31];
    printf("Nhap tuoi:");
    scanf("%d", &tuoi);
    fflush(stdin); //Làm sạch stdin
    printf("Nhap ten: ");
    gets(ten);
}
```

//Nếu nhập tên trước tuổi thì không cần làm sạch stdin

```
void main()
{
    int tuoi;
    char ten[31];
    printf("Nhap ten: ");
    gets(ten);
    printf("Nhap tuoi:");
    scanf("%d", &tuoi);
}
```

(5) Hàm xuất chuỗi

- Cú pháp: **puts**("văn bản" / tên_chuỗi);

VD: `puts("KTLC C"); //In chuỗi "KTLC C" ra màn hình`
`puts(s); //In chuỗi s ra màn hình`

(6) Một số hàm thao tác chuỗi (trong thư viện **string.h)**

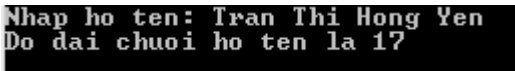
- strlen(s);** => Trả về độ dài chuỗi s.
- strcat(s1, s2);** => Ghép s2 vào s1 và trả về con trỏ đến s1.
- strcpy(s1, s2);** => Chép s2 đè lên s1 và trả về con trỏ đến s1.
- strcmp(s1, s2);** => So sánh s1 và s2. Hàm trả về giá trị âm nếu s1 đứng trước s2 trong bảng mã ASCII, giá trị 0 nếu s1 trùng s2 và giá trị dương nếu s1 đứng sau s2 trong bảng mã ASCII.
- stricmp(s1, s2);** => Hàm có chức năng tương tự như **strcmp** nhưng không phân biệt chữ hoa với chữ thường.
- strupr(s)** => IN HOA các ký tự trong chuỗi s
- strlwr(s)** => in thường các ký tự trong chuỗi s
- strchr(s, c);** => Tìm sự xuất hiện đầu tiên của ký tự c trong s. Nếu tìm thấy hàm trả về con trỏ đến ký tự tìm được trong s, nếu không hàm trả về giá trị NULL.
- strstr(s1, s2);** => Tìm sự xuất hiện đầu tiên của s2 trong s1. Nếu tìm thấy hàm trả về con trỏ đến chuỗi con tìm được trong s1, nếu không hàm trả về giá trị NULL.

VD1: Viết CT tính độ dài chuỗi họ tên do người dùng nhập vào.

```
#include<stdio.h>
#include<conio.h>
#include<string.h>

int DoDaiChuoi(char s[])
{
    int i=0;
    while(s[i]!='\0')
        i++;
    return i;
}

void main()
{
    char hoten[50];
    printf("Nhap ho ten: ");
    gets(hoten);
    printf("Do dai chuoi ho ten la %d", DoDaiChuoi(hoten));
    _getch();
}
```

Kết quả: 

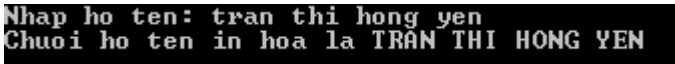
VD2: Viết CT in hoa chuỗi họ tên do người dùng nhập vào.

```
#include<stdio.h>
#include<conio.h>
#include<string.h>

int DoDaiChuoi(char s[])
{
    int i=0;
    while(s[i]!='\0')
        i++;
    return i;
}

void INHOA(char s[])
{
    int ls=DoDaiChuoi(s);
    for(int i=0;i<ls;i++)
        if(s[i]>='a'&&s[i]<='z')
            s[i]-=32;
}

void main()
{
    char hoten[50];
    printf("Nhap ho ten: ");
    gets(hoten);
    INHOA(hoten);
    printf("Chuoi ho ten in hoa la %s",hoten);
    _getch();
}
```

Kết quả: 

VD3: Viết CT so sánh 2 chuỗi họ tên do người dùng nhập vào.

```
#include<stdio.h>
#include<conio.h>
#include<string.h>

void SoSanhChuoi(char s1[],char s2[])
{
    if(strcmp(s1,s2)<0)
        printf("%s dung truoc %s",s1,s2);
    else if(strcmp(s1,s2)==0)
        printf("%s trung %s",s1,s2);
    else
        printf("%s dung sau %s",s1,s2);
}
```

```
void main()
{
    char hoten1[50],hoten2[50];
    printf("Nhap ho ten 1: ");
    gets(hoten1);
    printf("Nhap ho ten 2: ");
    gets(hoten2);
    SoSanhChuoai(hoten1,hoten2);
    _getch();
}
```

Kết quả:

```
Nhap ho ten 1: vo ngoc tan
Nhap ho ten 2: tran thi hong yen
vo ngoc tan dung sau tran thi hong yen
```

```
Nhap ho ten 1: vo ngoc tan
Nhap ho ten 2: UO NGOC TAN
vo ngoc tan dung sau UO NGOC TAN
```

Thay lệnh **strcmp** thành lệnh **stricmp** với CT trên để cho kết quả so sánh chính xác hơn (không phân biệt HOA / thường):

```
Nhap ho ten 1: vo ngoc tan
Nhap ho ten 2: UO NGOC TAN
vo ngoc tan trung UO NGOC TAN
```

MẢNG 2 CHIỀU

(1) Khái niệm

- Mảng hai chiều thực chất là mảng một chiều trong đó mỗi phần tử của mảng lại là mảng một chiều, và được truy xuất bởi hai chỉ số dòng và cột.

(2) Khai báo mảng 2 chiều

- Cú pháp: **<Kiểu> <tên mảng>[<Kích thước dòng>][<Kích thước cột>];**
- Mỗi phần tử của mảng được truy nhập thông qua tên mảng cùng với hai chỉ số: chỉ số dòng (bắt đầu từ 0 đến <Kích thước dòng> - 1) và chỉ số cột (bắt đầu từ 0 đến <Kích thước cột> - 1).

VD: `int a[2][3]; /* Khai báo mảng hai chiều gồm 6 phần tử kiểu int, bao gồm:`

```
a[0][0]    a[0][1]    a[0][2]
a[1][0]    a[1][1]    a[1][2]    */
```

(3) Khởi tạo mảng 2 chiều

VD:

```
//Khởi tạo mảng 2 chiều có 2 dòng và 3 cột
int a[2][3]={11, 12, 13},{21, 22, 23}};
//Khởi tạo có thể không cần chỉ ra kích thước dòng
double b[][3]={6, 7.8, 8},{3, 12.6, 4},{6.5, 20, 7}};
```

(4) Nhập / xuất mảng 2 chiều

```

#include "stdio.h"
#include "conio.h"

#define MAXROW 10
#define MAXCOL 10

void main()
{
    int a[MAXROW][MAXCOL], d, c;

    //Nhập mảng 2 chiều
    do {
        printf("Nhập số dòng và số cột: ");
        scanf("%d%d", &d, &c);
    } while (d < 1 || d > MAXROW || c < 1 || c > MAXCOL);
    for (int i = 0; i < d; i++)
        for (int j = 0; j < c; j++)
        {
            printf("a[%d][%d]=", i, j);
            scanf("%d", &a[i][j]);
        }

    //Xuất mảng 2 chiều
    for (int i = 0; i < d; i++)
    {
        for (int j = 0; j < c; j++)
            printf("%3d", a[i][j]);
        printf("\n");
    }

    _getch();
}

```

(5) Nhập / xuất mảng 2 chiều

- Tên mảng hai chiều cũng là một hằng địa chỉ và nó chính là địa chỉ phần tử đầu tiên của mảng.
- Khi dùng tên mảng làm tham số thực truyền cho hàm thì thực chất là địa chỉ phần tử đầu tiên của mảng được truyền cho hàm và như vậy tham số hình thức tương ứng trong định nghĩa hàm phải viết dưới dạng con trỏ.

```

voidNhapMang2cConTro(int (*a)[MAXCOL], int *sd, int *sc)
voidXuatMang2cConTro(int (*a)[MAXCOL], int sd, int sc)

```

- Tham số hình thức tương ứng với tham số thực là tên mảng hai chiều cũng có thể viết như sau:

```

voidNhapMang2c(int a[][MAXCOL], int &sd, int &sc)
voidXuatMang2c(int a[][MAXCOL], int sd, int sc)

```

```
#include <stdio.h>
#include <conio.h>

#define MAXROW 10
#define MAXCOL 10

void NhapMang2c(int a[][MAXCOL],int &sd,int &sc)
{
    do {
        printf("Nhap so dong va so cot:");
        scanf("%d%d",&sd,&sc);
    } while(sd<1||sd>MAXROW||sc<1||sc>MAXCOL);
    for(int i = 0; i < sd; i++)
        for(int j= 0; j < sc; j++)
        {
            printf("a[%d][%d]=", i, j);
            scanf("%d", &a[i][j]);
        }
}

void NhapMang2cConTro(int (*a) [MAXCOL],int *sd,int *sc)
{
    do {
        printf("Nhap so dong va so cot:");
        scanf("%d%d",&(*sd),&(*sc)); //scanf("%d%d",sd,sc);
    } while(*sd<1||*sd>MAXROW||*sc<1||*sc>MAXCOL);
    for(int i = 0; i < *sd; i++)
        for(int j= 0; j < *sc; j++)
        {
            printf("a[%d][%d]=", i, j);
            scanf("%d", *(a+i)+j);
        }
}

void XuatMang2c(int a[][MAXCOL],int sd,int sc)
{
    for(int i = 0; i < sd; i++)
    {
        for(int j = 0; j < sc; j++)
            printf("%3d", a[i][j]);
        printf("\n");
    }
}

void XuatMang2cConTro(int (*a) [MAXCOL],int sd,int sc)
{
    for(int i = 0; i < sd; i++)
    {
        for(int j = 0; j < sc; j++)
            printf("%3d", *(*a+i)+j);
        printf("\n");
    }
}
```

```

void main()
{
    int a[MAXROW][MAXCOL], d, c;
    NhapMang2c(a, d, c);
    XuatMang2c(a, d, c);
    _getch();
}

```

(6) Các bài toán cơ bản trên mảng 2 chiều

- **Duyệt hết các phần tử của mảng:**

a. Duyệt theo từng dòng từ trên xuống dưới, với mỗi dòng duyệt từ trái sang phải:

```

for(int i = 0; i < sd; i++)
    for(int j = 0; j < sc; j++)
    {
        //Xử lý a[i][j]
        ...
    }

```

b. Duyệt theo từng cột từ trái sang phải, với mỗi cột duyệt từ trên xuống dưới:

```

for(int j = 0; j < sc; j++)
    for(int i = 0; i < sd; i++)
    {
        //Xử lý a[i][j]
        ...
    }

```

- **Duyệt các phần tử nằm trên cùng một dòng hay trên cùng một cột:**

a. Duyệt các phần tử trên dòng có chỉ số k ($0 \leq k < sd$):

```

for(int j = 0; j < sc; j++)
{
    <Xử lý a[k][j]>
}

```

b. Duyệt các phần tử trên cột có chỉ số k ($0 \leq k < sc$):

```

for(int i = 0; i < sd; i++)
{
    <Xử lý a[i][k]>
}

```

VD1: Viết hàm tính tổng giá trị các phần tử của mảng 2 chiều.

```

void TinhTongGiaTriCacPhanTu(int (*a)[MAXCOL], int sd, int sc)
{
    int tong=0;
    for(int i = 0; i < sd; i++)
        for(int j = 0; j < sc; j++)
            tong+=*(*(a+i)+j); //tong+=a[i][j];
    printf("\nTong gia tri cac phan tu la %d\n", tong);
}

```

```

void main()
{
    int a[MAXROW][MAXCOL], d, c;
    NhapMang2c(a, d, c);
    XuatMang2c(a, d, c);
    TinhTongGiaTriCacPhanTu(a, d, c);
    _getch();
}

```

Kết quả:

```

Nhap so dong va so cot: 2 3
a[0][0]=1
a[0][1]=2
a[0][2]=3
a[1][0]=4
a[1][1]=5
a[1][2]=6
  1  2  3
  4  5  6

Tong gia tri cac phan tu la 21

```

VD2: Viết hàm tính tổng giá trị các phần tử trên dòng k của mảng 2 chiều.

```

void TinhTongGiaTriCacPTuTrenDongk(int a[][MAXCOL], int sd, int sc)
{
    int dong, tong=0;
    do{
        printf("Nhap so dong: ");
        scanf("%d", &dong);
    }while(dong<0 || dong>=sd);
    for(int j=0; j<sc; j++)
        tong+=*(a+dong)+j; //tong+=a[dong][j];
    printf("\nTong gia tri cac phan tu tren dong %d la
           %d\n", dong, tong);
}

```

```

void main()
{
    int a[MAXROW][MAXCOL], d, c;
    NhapMang2c(a, d, c);
    XuatMang2c(a, d, c);
    TinhTongGiaTriCacPTuTrenDongk(a, d, c);
    _getch();
}

```

Kết quả:

```

Nhap so dong va so cot: 2 3
a[0][0]=1
a[0][1]=2
a[0][2]=3
a[1][0]=4
a[1][1]=5
a[1][2]=6
  1  2  3
  4  5  6
Nhap so dong: 1
Tong gia tri cac phan tu tren dong 1 la 15

```


VD3: Viết hàm sắp xếp tăng dần các phần tử trên cột k của mảng 2 chiều.

```

void TinhTongGiaTriCacPTuTrenDongk(int a[][MAXCOL], int sd, int sc)
{
    int dong, tong=0;
    do{
        printf("Nhap so dong: ");
        scanf("%d", &dong);
    }while(dong<0 || dong>=sd);
    for(int j=0; j<sc; j++)
        tong+=*(a+dong)+j); //tong+=a[dong][j];
    printf("\nTong gia tri cac phan tu tren dong %d la
           %d\n", dong, tong);
}

void main()
{
    int a[MAXROW][MAXCOL], d, c;
    NhapMang2c(a, d, c);
    XuatMang2c(a, d, c);
    TinhTongGiaTriCacPTuTrenDongk(a, d, c);
    _getch();
}

```

Kết quả:

```

Nhap so dong va so cot: 3 3
a[0][0]=0
a[0][1]=-7
a[0][2]=12
a[1][0]=-5
a[1][1]=8
a[1][2]=1
a[2][0]=20
a[2][1]=6
a[2][2]=-9
 0 -7 12
-5 8 1
20 6 -9
Nhap so cot can sap xep: 2
Mang 2 chieu sap xep tang dan theo cot 2 la:
 0 -7 -9
-5 8 1
20 6 12

```

(7) Mảng vuông (ma trận vuông)• **Tính chất:**

Mảng vuông là mảng hai chiều có số dòng = số cột = n, n gọi là cấp ma trận.

• **Đặc tính của ma trận vuông:**

- a. Các phần tử nằm trên đường chéo chính là $a[i][i]$ với $0 \leq i < n$



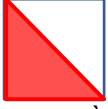
- b. Các phần tử nằm trên đường chéo phụ là $a[i][n-1-i]$ với $0 \leq i < n$



- c. Các phần tử nằm trong **nửa mảng vuông phía trên đường chéo chính** là **$a[i][j]$** với **$i \leq j$**



- d. Các phần tử nằm trong **nửa mảng vuông phía dưới đường chéo chính** là **$a[i][j]$** với **$i \geq j$**



- e. Các phần tử nằm trong **nửa mảng vuông phía trên đường chéo phụ** là **$a[i][j]$** với **$i + j \leq n - 1$**



- f. Các phần tử nằm trong **nửa mảng vuông phía dưới đường chéo phụ** là **$a[i][j]$** với **$i + j \geq n - 1$**



• **Duyệt mảng vuông:**

- a. Duyệt các phần tử nằm trên ĐCC



```
for(int i = 0; i < n; i++)
    <Xử lý a[i][i]>
```

- b. Duyệt các phần tử nằm trên ĐCP



```
for(int i = 0; i < n; i++)
    <Xử lý a[i][n - 1 - i]>
```

- c. Duyệt các phần tử nằm trong nửa mảng vuông phía trên ĐCC



```
for(int i = 0; i < n; i++)
    for(int j = 0; j < n; j++)
        if(i <= j)
        {
            <Xử lý a[i][j]>
        }
```

- d. Duyệt các phần tử nằm trong nửa mảng vuông phía dưới ĐCC

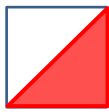


```
for(int i = 0; i < n; i++)
    for(int j = 0; j < n; j++)
        if(i >= j)
        {
            <Xử lý a[i][j]>
        }
```

- e. Duyệt các phần tử nằm trong nửa mảng vuông phía trên ĐCP



```
for(int i = 0; i < n; i++)
    for(int j = 0; j < n; j++)
        if(i + j <= n - 1)
            <Xử lý a[i][j]>
```

f. Duyệt các phần tử nằm trong nửa mảng vuông phía dưới ĐCP

```
for(int i = 0; i < n; i++)
    for(int j = 0; j < n; j++)
        if(i + j >= n - 1)
            <Xử lý a[i][j]>
```

VD: Viết CT nhập / xuất ma trận vuông, tìm phần tử lớn nhất trên đường chéo chính và kiểm tra ma trận vuông có tồn tại phần tử âm ở nửa dưới đường chéo phụ không?

```
#include <stdio.h>
#include <conio.h>

#define Cap_n 10

void NhapMangVuong(int a[][Cap_n], int &n)
{
    do {
        printf("Nhap cap ma tran: ");
        scanf("%d", &n);
    } while(n < 1 || n > Cap_n);
    for(int i = 0; i < n; i++)
        for(int j = 0; j < n; j++)
        {
            printf("a[%d][%d]=", i, j);
            scanf("%d", &a[i][j]);
        }
}

void XuatMangVuong(int a[][Cap_n], int n)
{
    for(int i = 0; i < n; i++)
    {
        for(int j = 0; j < n; j++)
            printf("%3d", a[i][j]);
        printf("\n");
    }
}

void TimMaxDCC(int a[][Cap_n], int n)
{
    int max = a[0][0], vtmax = 0;
    for(int i = 1; i < n; i++)
        if(a[i][i] > max)
        {
            max = a[i][i];
            vtmax = i;
        }
    printf("Phan tu lon nhat tren DCC la a[%d][%d] = %d\n", vtmax, vtmax, max);
}
```

```
int KiemTraTonTaiPhanTuAmDuoiDCP(int a[][Cap_n],int n)
{
    int am=0;
    for(int i = 0; i < n; i++)
        for(int j = 0; j < n; j++)
            if(i + j >= n - 1)
                if(a[i][j]<0)
                {
                    am=1;
                    break;
                }
    return am;
}

void main()
{
    int a[Cap_n][Cap_n],n;
   NhapMangVuong(a,n);
   XuatMangVuong(a,n);
    TimMaxDCC(a,n);
    if(KiemTraTonTaiPhanTuAmDuoiDCP(a,n)==1)
        printf("Ton tai phan tu am o nua duoi DCP\n");
    else
        printf("Khong ton tai phan tu am o nua duoi DCP\n");
    _getch();
}
```