

# Tin Học Cơ Sở 4: Mạng và Chuỗi



**Vũ Thị Hồng Nhạn**

(vthnhan@vnu.edu.vn)

Human Machine Interaction Lab.

Vietnam National Univ., Hanoi

# Nội dung

- ❖ Khai báo mảng, chuỗi
- ❖ Nhập & xuất mảng, chuỗi
- ❖ Khởi tạo mảng, chuỗi
- ❖ Một số kỹ thuật thao tác trên mảng, chuỗi
- ❖ Dùng mảng làm tham số cho hàm

# Mảng

- ❖ Là một tập hợp các phần tử có cùng dữ liệu
- ❖ Ví dụ: muốn lưu 3 số thực nhập từ bàn phím để tính trung bình

Lúc trước

float a, b, c

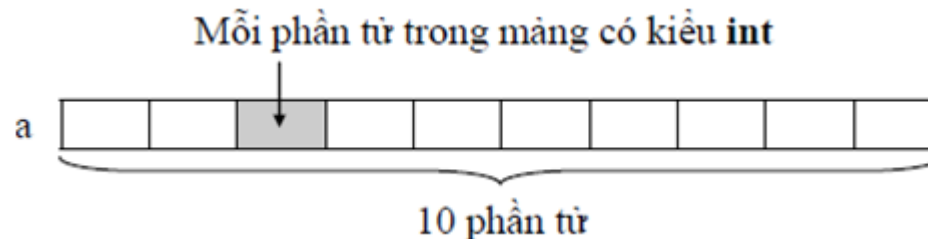
Nhập vào ba số và gán cho a,b,c

Tính trung bình  $(a+b+c)/3$

# Mảng 1 chiều

## Khai báo mảng

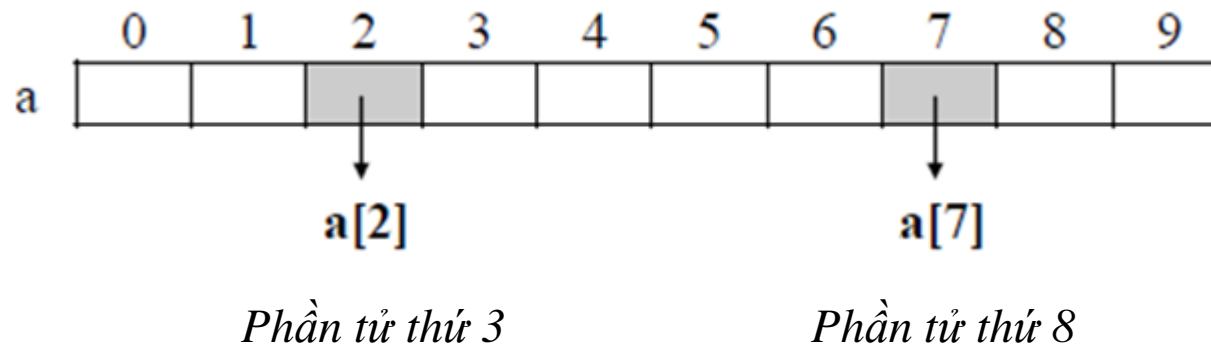
- ❖ Khai báo một mảng, sử dụng `[]`
  - Vd. Tạo một mảng với 3 phần tử số nguyên
  - `int a[3]`
- ❖ Không thể thay đổi kích cỡ của mảng
- ❖ Số nằm giữa cặp dấu `[]` phải là **hằng số** (constant)
- ❖ Có thể khởi tạo giá trị cho các phần tử trong hàm
  - `int a[10] = {2, -1, 6, 2, 8, 0, 3, 5, 1, 3};`
  - Cách tốt nhất để trình biên dịch tự tính kích thước: `int a[] = {2, -1, 6};`



# Mảng 1 chiều

## Tham chiếu đến từng phần tử

- ❖ Sau khi khai báo, mỗi phần tử trong mảng đều có một **chỉ số để tham chiếu**
  - $n$ : kích cỡ của mảng
  - Chỉ số từ  $0$  tới  $n-1$



# Mảng 1 chiều

## Đọc dữ liệu cho mảng

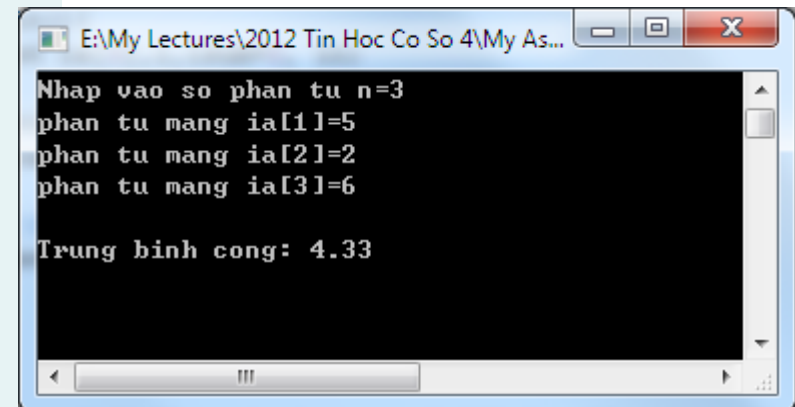
```
for (i=0; i<n; i++){  
    printf("Nhap vao phan tu thu %d:",i+1);  
    scanf("%d", &a[i]);  
}
```

# Mảng 1 chiều

## Ví dụ 1

**Nhập n số nguyên và tính giá trị trung bình**

```
int main(){
    int a[50], i, sum=0, n;
    printf("Nhap vao so phan tu n=");
    scanf("%d", &n);
    //Nhap du lieu vao mang
    for(i=0; i<n; i++){
        printf("phan tu mang a[%d]=", i+1);
        scanf("%d", &a[i]);
        sum +=a[i];
    }
    printf("Trung binh cong: %.2f\n", (float)sum/n);
}
```



```
E:\My Lectures\2012 Tin Hoc Co So 4\My As...
Nhap vao so phan tu n=3
phan tu mang ia[1]=5
phan tu mang ia[2]=2
phan tu mang ia[3]=6

Trung binh cong: 4.33
```

# Mảng 1 chiều

## Nhận xét

❖ Điều gì sẽ xảy ra nếu ta nhập  $n > 50$  ?

→ Yêu cầu nhập

```
do{  
    printf("");  
    scanf();  
}while (n<0 || n>50)
```



# Mảng 1 chiều

## Ví dụ 2

Nhập vào 1 dãy số **dương** rồi in tổng các số **dương** đó, nếu nhập số  $\leq 0$  thì ctr kết thúc

```
#define MAX 4
```

```
int main(){
```

```
float a[MAX], sum=0;
```

```
int i=0, j;
```

```
do{
```

```
    if(i>MAX){
```

```
        printf("Array is full\n");
```

```
        i++;
```

```
        break;
```

```
    }
```

```
    printf("Nhap vao phan tu a[%d]=",i+1);
```

```
    scanf("%f", &a[i]);
```

```
    i++;
```

```
}while(a[i] >0);
```

```
i--;
```

```
for(j=0; j<i; j++) //calculate sum
```

```
    sum +=a[j];
```

```
printf("Sum=%.2f\n", sum);
```

```
}
```

*Kiểm tra số lượng phần*

*tử đã vượt quá giới hạn*

*chưa*

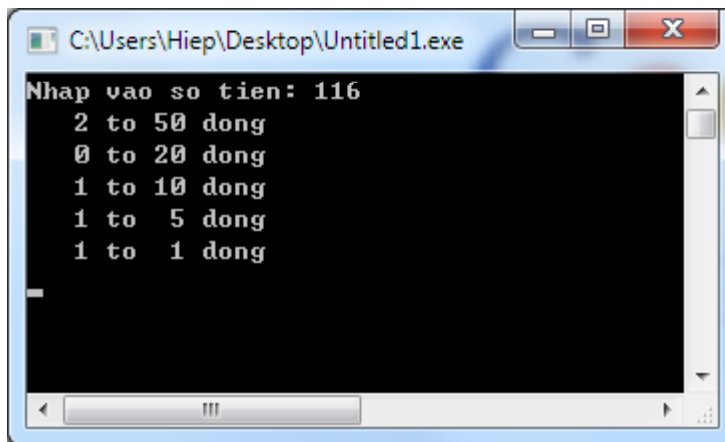
*Để chấm dứt nhập số  
liệu, hãy nhập vào số 0  
hoặc âm*

*Tính & in tổng*

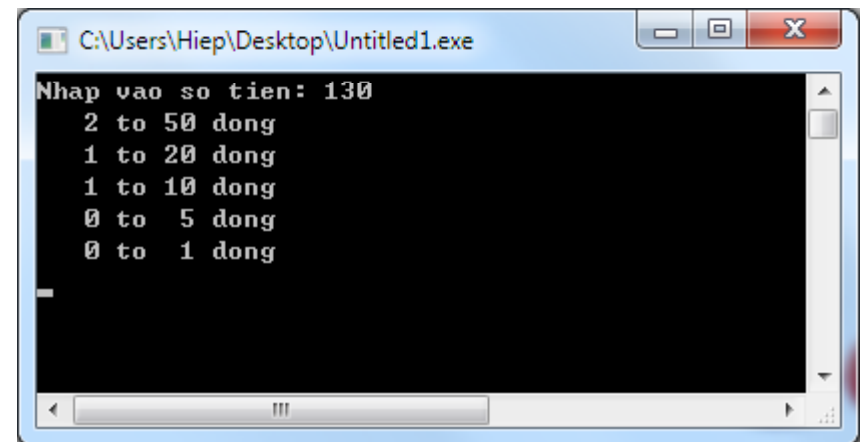
# Mảng 1 chiều

## Ví dụ 3: luyện tập

- ❖ Có 4 loại tiền 1, 5, 10, 20 và 50 đồng. Hãy viết chương trình nhập vào số tiền sau đó cho biết số tiền trên gồm mấy loại tiền, mỗi loại bao nhiêu tờ



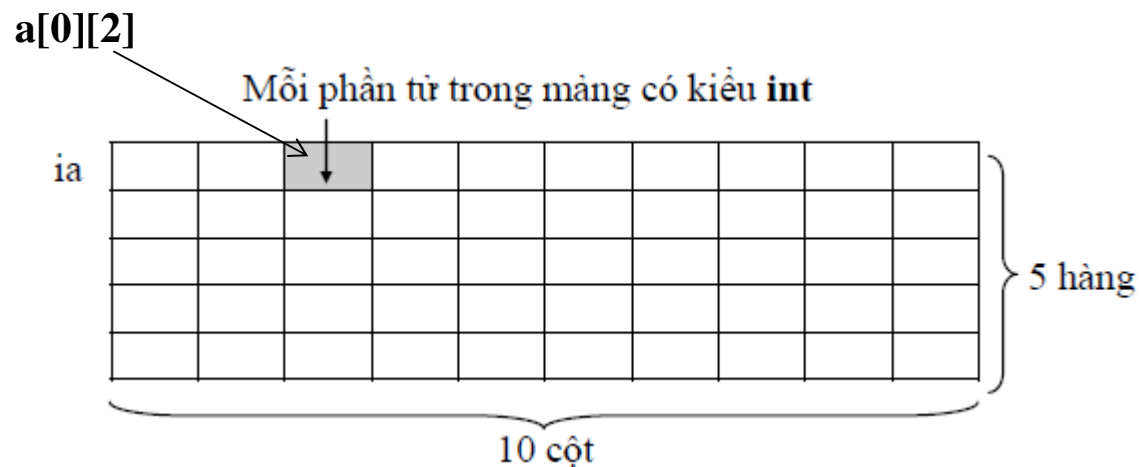
```
C:\Users\Hiep\Desktop\Untitled1.exe
Nhap vao so tien: 116
 2 to 50 dong
 0 to 20 dong
 1 to 10 dong
 1 to 5 dong
 1 to 1 dong
```



```
C:\Users\Hiep\Desktop\Untitled1.exe
Nhap vao so tien: 130
 2 to 50 dong
 1 to 20 dong
 1 to 10 dong
 0 to 5 dong
 0 to 1 dong
```

# Mảng 2 chiều

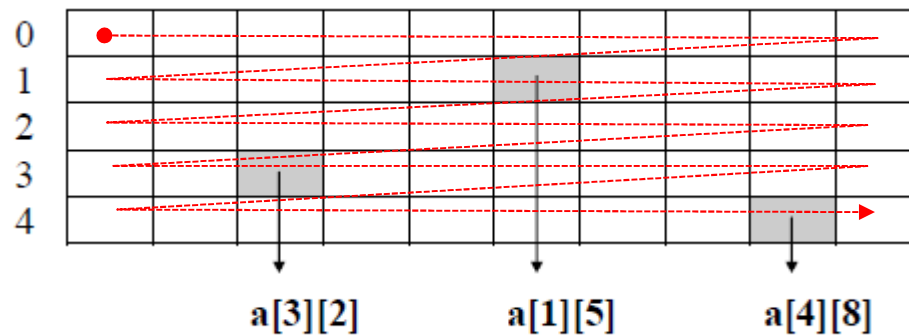
- ❖ Khai báo mảng 2 chiều, ví dụ, `a[5][10]`
  - mỗi phần tử trong mảng 2 chiều đều có 2 chỉ số để tham chiếu, **chỉ số hàng** và **chỉ số cột**
  - `a[chỉ số hàng][chỉ số cột]`



# Mảng 2 chiều

## Nhập dữ liệu

- ❖ Thứ tự nhập dữ liệu vào mảng



```
for(i=0; i<n; i++)  
    for(j=0; j<m; j++){  
        printf("Nhap phan tu a[%d][%d]=", i+1, j+1);  
        scanf("%d", &a[i][j]);  
    }  
}
```

# Mảng 2 chiều

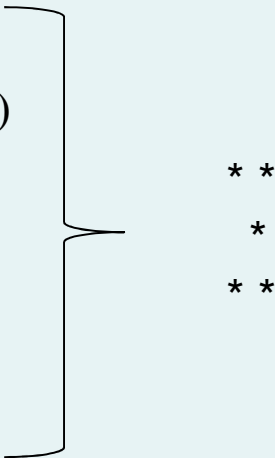
## Ví dụ 4

1. Khởi tạo ma trận với giá trị 0,1, in ma trận với dấu \* cho giá trị 1 và ngược lại để trống

```
#define MAX 3
int M[MAX][MAX]={{ 1, 0, 1},
                  {0, 1, 0},
                  {1, 0, 1}};

int main(){
    int i=0, j=0;

    for(i=0; i<MAX; i++){
        for(j=0; j<MAX; j++)
            if(M[i][j]==1)
                printf("*");
            else printf(" ");
        printf("\n");
    }
```



```
  * *
   *
  * *
```

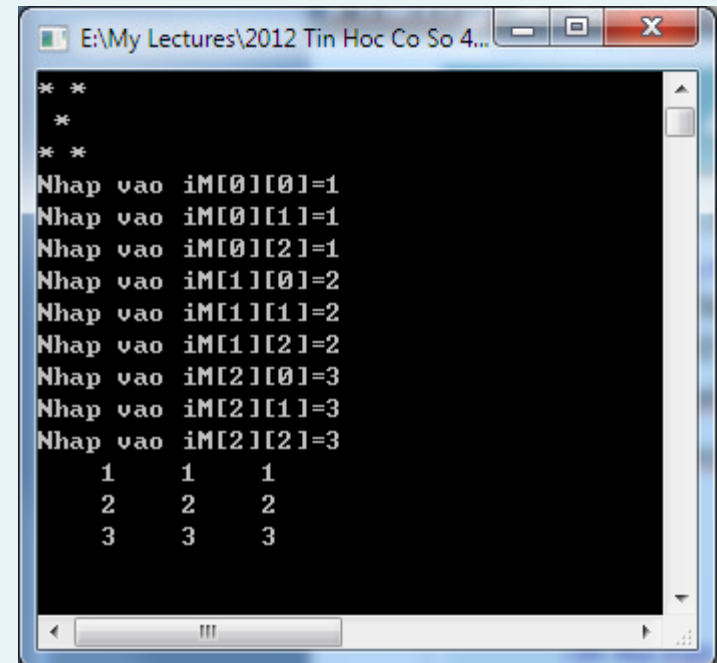
# Mảng 2 chiều

## Ví dụ 4

2. Nhập dữ liệu vào một ma trận nxn và in ra ma trận đó

```
//Nhập giá trị vào ma trận
for(i=0; i<MAX; i++){
    for(j=0; j<MAX; j++){
        printf("Nhập vào M[%d][%d]=",i,j);
        scanf("%d",&M[i][j]);
    }
}

//In ma trận
for(i=0;i<MAX; i++){
    for(j=0;j<MAX;j++){
        printf("%5d",M[i][j]);
    }
    printf("\n");
}
}
```



```
E:\My Lectures\2012 Tin Hoc Co So 4...
** *
**
** *
Nhập vào iM[0][0]=1
Nhập vào iM[0][1]=1
Nhập vào iM[0][2]=1
Nhập vào iM[1][0]=2
Nhập vào iM[1][1]=2
Nhập vào iM[1][2]=2
Nhập vào iM[2][0]=3
Nhập vào iM[2][1]=3
Nhập vào iM[2][2]=3
  1   1   1
  2   2   2
  3   3   3
```

# Ví dụ 5

## Hàm với tham số là mảng 1 chiều

Tìm số lớn nhất trong ma trận bất kỳ, **nhập 0 để thoát** khi nhập liệu

```
#define MAX 3
//Khai báo hàm
int input(int);
int max(int, int);

//Định nghĩa hàm nhập liệu
int input(int a[]){
    int i=0;
    printf("Nhap vao so 0 de ket thuc\n");
    do{
        printf("Nhap vao a[%d]=", i);
        scanf("%d", &a[i]);
        i++;
    }while( a[i] !=0 );
    i--;
    return i; //trả về tổng số lượng phần tử đã nhập vào mảng
}
```

**Khi truyền mảng (có thể 1 hoặc 2 chiều), hàm không tạo bản sao mới**

# Ví dụ 5

## Hàm với tham số là mảng 1 chiều

Tìm số lớn nhất trong ma trận

```
int max(int a[],int in){ //Định nghĩa hàm tìm giá trị lớn nhất
    int max = a[0], i;
    for(i=1; i<n; i++)
        if(a[i] > max)
            max = a[i];
    return max;
}

int main(){
    int a[MAX];
    int max,n;

    n = input(a);      //gọi hàm nhập liệu
    max = max(a,n);    //gọi hàm tìm giá trị lớn nhất
    printf("So lon nhat trong mang: %d", max);
}
```



## Ví dụ 6

### Hàm với tham số là mảng hai chiều

Nhập 2 ma trận vuông cấp n,

```
#define MAX 3

//Khai báo hàm
void input(float);
void output(float);
void add(float,float,float);

int n; //bien toan cuc

void input(float a[][MAX]){
    for(int i=0; i<n; i++){
        for(int j=0; j<n; j++){
            printf("Nhap vao phan tu[%d][%d]=" ,i,j);
            scanf("%f", &a[i][j]);
        }
    }
}
```

Cộng 2 ma trận và in ra kết quả

```
void output(float a[][MAX]){
    for(int i=0;i<n;i++){
        for(int j=0;j<n;j++){
            printf("%5.2f", a[i][j]);
            printf("\n");
        }
    }
}

void add(float a[][MAX], float b[][MAX], float
c[][MAX]){
    for(int i=0; i<n; i++)
        for(int j=0; j<n; j++)
            c[i][j] = a[i][j]+ b[i][j];
}
```

## Ví dụ 6

### Hàm với tham số là mảng hai chiều

Nhập 2 ma trận vuông cấp n, Cộng 2 ma trận và in ra kết quả

```
int main(){
    float a[MAX][MAX], b[MAX][MAX],c[MAX][MAX];

    printf("Nhap vao cap ma tran:");
    scanf("%d", &n);

    printf("Nhap lieu ma tran a\n");
    input(a);

    printf("Nhap lieu ma tran b\n");
    input(b);

    printf("Tong ma tran a+b:\n");
    add(a,b,c);
    output(c);

    getch(); return 0;
}
```

#### ❖ Tham số truyền vào hàm

- mảng 2 chiều dạng **a[][MAX]**
- vì hàm không giành chỗ cho mảng, hàm chỉ cần biết số cột để tham khảo đến các phần tử

## Ví dụ 6

### Giải thích

- ❖ Mảng 2 chiều được khai báo `a[3][3]`

Truyền tham số vào hàm: `a[][3]`,  
để tham khảo đến ptử `a[2][1]`,  
hàm tính như sau:

$$2 * 3 + 1 = 7 \text{ (chỉ số hàng * số cột + chỉ số cột)}$$

`a[3][3]` gồm 9 phần tử được lưu trữ trong bộ nhớ như sau:

