



TRƯỜNG ĐẠI HỌC THỦY LỢI  
Khoa Công nghệ thông tin  
Bộ môn Tin học và KTTT

# **NHẬP MÔN LẬP TRÌNH**

Giảng viên: TS.GVC Nguyễn Quỳnh Diệp

Email: [diepnq@tlu.edu.vn](mailto:diepnq@tlu.edu.vn)

Điện thoại: 0904345673

# Chương 4: Mảng, con trỏ và chuỗi ký tự

## 4.1. Mảng

- Khái niệm
- Khai báo và sử dụng
- Các thao tác thường gặp

## 4.2. Con trỏ

- Khái niệm và cách khai báo
- Toán tử địa chỉ (&), toán tử nội dung (\*)
- Phép toán trên con trỏ
- Con trỏ và mảng

## 4.3. Chuỗi ký tự

- Khái niệm, khai báo và sử dụng
- Các hàm xử lý ký tự và chuỗi ký tự
- Mảng chuỗi ký tự



# 4.1. MẢNG

---

- Bài toán: Nhập điểm thi (số nguyên) môn Nhập môn lập trình cho lớp gồm 50 sinh viên rồi đưa ra số lượng sinh viên phải học lại
- Phương pháp: Điểm của mỗi sinh viên là 1 biến
  - Tên biến là tên sinh viên
    - Ví dụ: `int An, Anh, Binh1, Binh2, Cuong,..... Van, Viet;`
  - Tên biến dạng "*dx*" với *x* là chỉ số thứ tự của SV trong lớp
    - Ví dụ: `int d1, d2, d3,.....,d50;`
- Nhận xét 1: Không hợp lý
  - Có quá nhiều biến (*Điểm thi cho toàn trường.. !?*)
  - Khó khăn cho các thao tác duyệt toàn bộ danh sách
    - Số SV học lại: `if(d1 <5) d++; if(d2 <5) d++; .....if(d50 <5) d++;`
- Nhận xét 2: Các biến có chung ý nghĩa, tính chất

- Trong thực tế, thường gặp các đối tượng có tính chất chung
  - Tháng trong năm
  - Điểm trung bình của sinh viên trong lớp
- Các đối tượng được nhóm lại dưới một tên
- Đối tượng được đặc trưng bởi **tên nhóm** và **thứ tự** trong nhóm
  - Phần tử thứ 4 trong dãy số
- Số thứ tự của đối tượng trong nhóm là **chỉ số phần tử**

- Kiểu mảng là một kiểu dữ liệu gồm:
  - Một số hữu hạn thành phần.
  - Các thành phần có cùng một kiểu: gọi là kiểu cơ sở hay là kiểu thành phần.
- Mỗi phần tử của mảng được xác định thông qua:
  - Tên mảng
  - Chỉ số của phần tử trong mảng

**Kiểu\_dữ\_liệu** Tên\_Mảng[Kích\_thước];

- **Kiểu\_dữ\_liệu**: kiểu của các phần tử trong mảng (nguyên, thực, ký tự, chuỗi, mảng,...)
- **Tên\_Mảng**: tên của mảng khai báo
- **Kích\_thước**: số phần tử **tối đa** trong mảng.

Ví dụ:

int DiemTin[50]; // khai báo mảng tên là DiemTin  
gồm 50 phần tử có kiểu dữ liệu int

float A[10]; //Mảng tên A gồm 10 phần tử kiểu số  
thực

- Các phần tử trong mảng được cấp phát các ô nhớ **kế tiếp nhau** trong bộ nhớ
- Kích thước của mảng bằng kích thước một phần tử nhân với số phần tử

Ví dụ:

`int A[10];`      //Mảng A gồm 10 phần tử nguyên

A[0]	A[1]	A[2]	A[3]	A[4]	A[5]	A[6]	A[7]	A[8]	A[9]
------	------	------	------	------	------	------	------	------	------

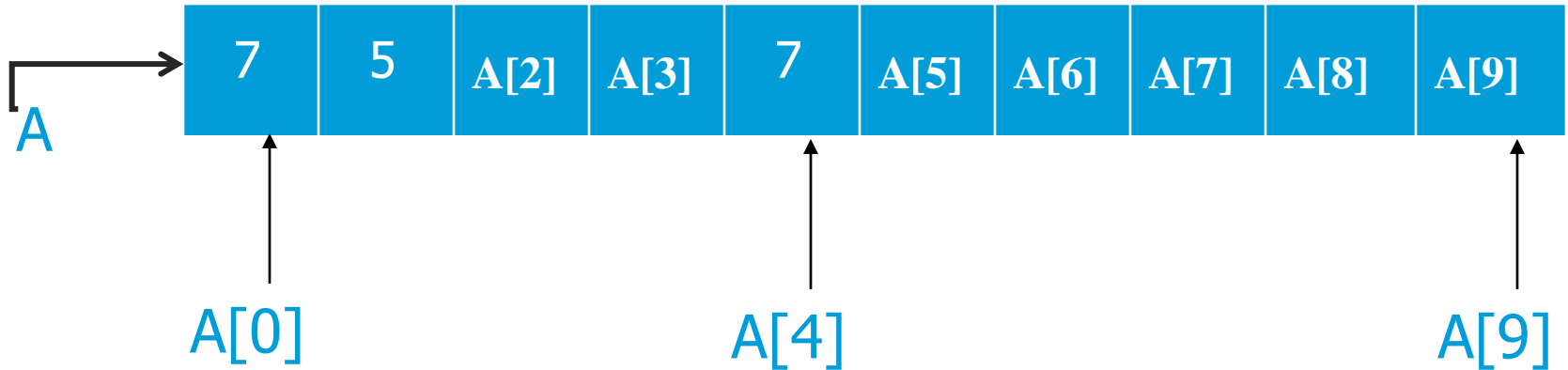
Kích thước của mảng A:  $10 \times 4 = 40$  bytes



- Biến mảng lưu trữ **địa chỉ ô nhớ đầu tiên** trong vùng nhớ được cấp phát
- Ngôn ngữ C đánh chỉ số các phần tử trong mảng **bắt đầu từ 0**
- Các phần tử của mảng được truy nhập thông qua
  - Tên mảng và
  - Chỉ số của phần tử của phần tử trong mảng

`Tên_Mang[Chỉ_số_phần_tử];`

`int A[10];` //Mảng A gồm 10 phần tử nguyên



`A[0] = 7;`

`A[1] = 5;`

`A[4] = 7;`

`int N = A[1] + A[4];` // `N = 12`

```
int A[10], i;  
for(i = 0; i < 10; i++) A[i] = 2 * i;
```

0	2	4	6	8	10	12	14	16	18
---	---	---	---	---	----	----	----	----	----

i : 

**Chú ý:** C không kiểm tra vượt quá giới hạn của mảng khi truy nhập

```
int A[3], B[4], C[3];
```

A[0]	A[1]	A[2]	B[0]	B[1]	B[2]	B[3]	C[0]	C[1]	C[2]
------	------	------	------	------	------	------	------	------	------

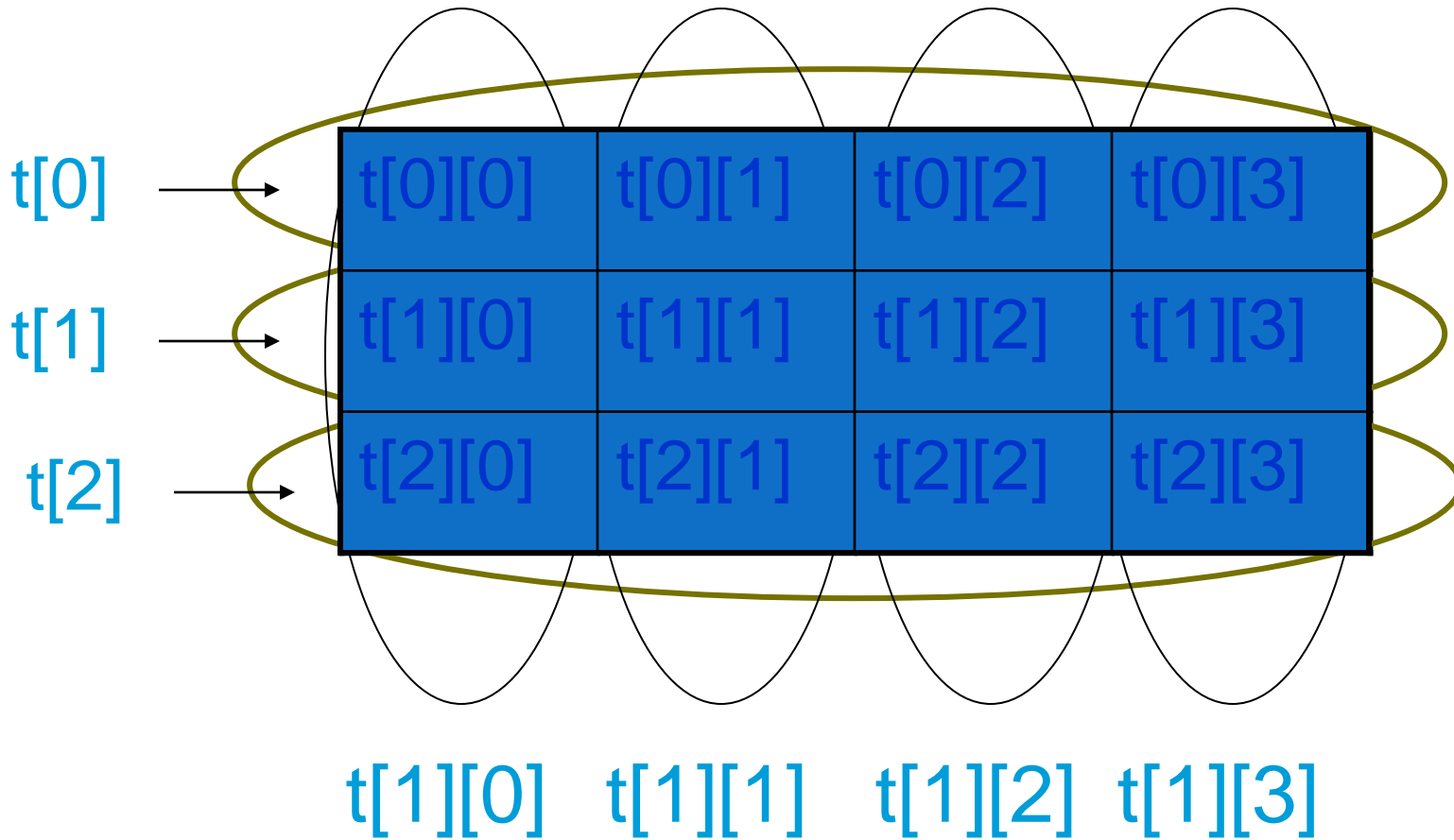
$A[5] \Leftrightarrow B[2] \Leftrightarrow C[-2] \leftarrow$  nếu cấp phát liên tiếp

- Mỗi phần tử của mảng có thể là một mảng → Mảng nhiều chiều

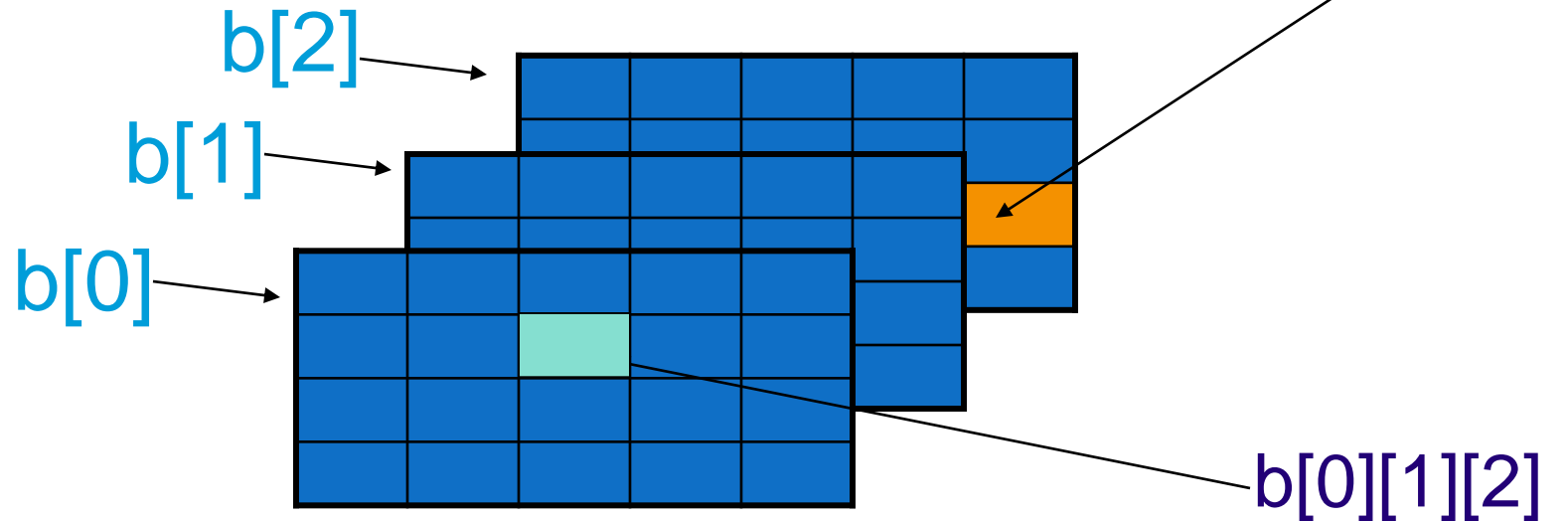
```
Kiểu_dữ_liệu Tên_mảng[Chiều_1] [Chiều_2]... [Chiều_N];
```

- **Kiểu\_dữ\_liệu**: Kiểu của mỗi phần tử trong mảng
- **Chiều\_1, Chiều\_2, ..., Chiều\_N**: Các hằng số nguyên, cho biết kích thước (số phần tử) của mỗi chiều
- Mảng gồm: Chiều\_1 x Chiều\_2 x ... x Chiều\_N phần tử được lưu trữ trong vùng nhớ liên tục. Các phần tử thuộc kiểu **Kiểu\_dữ\_liệu**.

```
int t[3][4];
```



```
int b[3][4][5];
```



- Mảng `b` gồm 3 phần tử `b[0]`, `b[1]`, `b[2]`
- Mỗi phần tử là mảng hai chiều gồm 4 hàng (hàng 0, 1, 2, 3) và 5 cột (0, 1, 2, 3, 4)
- Mỗi phần tử là một số nguyên có dấu 4 byte

Các phần tử của mảng có thể được khởi tạo giá trị ngay khi khai báo

Ví dụ:

```
int a[4] = {1,4,6,2};
```

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

```
int t[3][4] = { {1, 2, 3, 4},{5, 6, 7, 8},{9, 10, 11, 12}};
```

- Số lượng giá trị khởi tạo không được lớn hơn số lượng phần tử trong mảng
  - Nếu số lượng này nhỏ hơn thì các phần tử còn lại được khởi tạo giá trị **0**

```
int A[3][4] = { {1}, {4,5} };
```

```
int A[3][4] = { }; //Tất cả đều mang giá trị 0
```

- Có thể xác định kích thước mảng thông qua số giá trị khởi tạo nếu để trống kích thước mảng

```
int A1 [8] = {2, 4, 6, 8, 10, 12, 14, 16};
```

```
int A2 [] = {2, 4, 6, 8, 10, 12, 14, 16};
```



- Nhập/Xuất dữ liệu cho mảng
  - Mảng 1 chiều, ma trận
- Bài toán đếm
  - Đếm số phần tử
  - Tính toán trên các phần tử..
- Tìm kiếm phần tử
  - Lớn nhất/nhỏ nhất/bất kỳ
- Sắp xếp phần tử trong mảng
  - Theo thứ tự, theo nguyên tắc
- Chèn thêm phần tử, xóa phần tử

## Ví dụ:

```
int A[10];
```

- Nhập dữ liệu cho một phần tử

```
scanf("%d",&A[2]); //phần tử thứ 3 của mảng
```

- Nhập dữ liệu cho cả mảng

- Dùng vòng lặp for

```
for(i = 0; i<10; i++)
```

```
    scanf("%d", &A[i]);
```

- Nên in ra chỉ số phần tử khi nhập

```
printf("A[%d]: ", i);
```

```
scanf("%d", &A[i])
```



# Ví dụ 1



//Nhập vào lượng mưa (mm) trong năm

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

```
#define thang 12
```

```
int main(){
```

```
    int Mua[thang], i;
```

```
    for(i=0; i<thang; i++)
```

```
{
```

```
        printf("Nhap luong mưa tháng %d:", i+1);
```

```
        scanf("%d", &Mua[i]);
```

```
}
```

```
return 0;
```

```
}
```

- Nếu số phần tử của mảng chỉ được biết tại thời điểm thực hiện chương trình (nhưng *biết số phần tử tối đa*) thì:
  - Khai báo mảng với **kích thước tối đa**
  - Sử dụng biến **nguyên** lưu số phần tử thực sự của mảng.

## Ví dụ:

- Nhập vào mảng không quá 100 số thực
  - Khai báo mảng thực A có tối đa 100 phần tử.
  - Nhập số phần tử thực sự của mảng
  - Nhập giá trị cho từng phần tử (dùng for)

```
// Nhập mảng A tối đa 100 phần tử là các số thực
#include<stdio.h>
int main(){
    float A[100];
    int n, i;
    do{
        printf("Cho biet so phan tu cua mang: ");
        scanf("%d",&n);
    }while (n>100||n<=0);
    for(i = 0; i < n; i++)
    {
        printf("A[%d] = ", i);
        scanf("%f",&A[i]);
    }
    return 0;
}
```

## Ví dụ:

```
int A[10];
```

- Hiển thị phần tử thứ 5:

```
printf("%d", A[4]);
```

- Để hiển thị tất cả các phần tử:

```
for(i = 0; i < 10; i++)
```

```
printf("%4d", A[i]);
```

- Các kiểu xuất dữ liệu:

- Hiển thị tất cả/một phần theo dòng/cột..
- Hiển thị từng **k** phần tử trên một dòng...

```
#include <stdio.h>
#define MAX 12
int main(){
    int A[MAX], i;
    for ( i=0; i < MAX; i++ ){
        printf("A[%d]: ", i+1);
        scanf("%d", &A[i]);
    }
    printf("\n");
    for(i=0; i < MAX; i++ )
        printf("%4d", A[i]);
    printf("\n");
    for(i=0; i < MAX; i++ )
        printf("%d\n" , A[i]);
    printf("\n");
    for(i=0; i < MAX; i++ ){
        printf("%4d", A[i]);
        if( (i+1) %4==0) printf("\n");
    }
    return 0;
}
```

```
A[5]: 5
A[6]: 6
A[7]: 7
A[8]: 8
A[9]: 9
A[10]: 10
A[11]: 11
A[12]: 12
```

```
1 2 3 4 5 6 7 8 9 10 11 12
1
2
3
4
5
6
7
8
9
10
11
12
1 2 3 4
5 6 7 8
9 10 11 12
```

## //Nhập và đưa ra màn hình một ma trận

```
Nhap so hang: 3
Nhap so cot: 3

Nhap phan tu A[1,1]: 1
Nhap phan tu A[1,2]: 2
Nhap phan tu A[1,3]: 3
Nhap phan tu A[2,1]: 4
Nhap phan tu A[2,2]: 5
Nhap phan tu A[2,3]: 6
Nhap phan tu A[3,1]: 7
Nhap phan tu A[3,2]: 8
Nhap phan tu A[3,3]: 9
```

MA TRAN DA NHAP

1	2	3
4	5	6
7	8	9

-----  
Process exited after 10.68 second  
Press any key to continue . . .

```
Nhap so hang: 2
Nhap so cot: 3
```

```
Nhap phan tu A[1,1]: 1
Nhap phan tu A[1,2]: 2
Nhap phan tu A[1,3]: 3
Nhap phan tu A[2,1]: 4
Nhap phan tu A[2,2]: 5
Nhap phan tu A[2,3]: 6
```

MA TRAN DA NHAP

1	2	3
4	5	6

-----  
Process exited after 5.943 second  
Press any key to continue . . .



```
#include <stdio.h>
int main(){
    int A[20][20], n, m, i,j;
    printf("Nhap so hang: "); scanf("%d",&n);
    printf("Nhap so cot: "); scanf("%d",&m);
    printf("\n");
    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]);
        }
    printf("\n\nMA TRAN DA NHAP\n\n");
    for(i=0;i < n; i++ ){
        for(j=0;j < m; j++)
            printf("%4d", A[i][j]);
        printf("\n");
    }
    return 0;
}
```

```
#include <stdio.h>
int main(){
    int A[20][20], n, m, i, j;
    printf("Nhap so hang: "); scanf("%d",&n);
    printf("Nhap so cot: "); scanf("%d",&m);
    printf("\n");
    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]);
        }
    printf("\n\nMA TRAN DA NHAP\n\n");
    for(i=0; i < n; i++){
        for(j=0; j < m; j++)
            printf("%4d", A[i][j]);
        printf("\n");
    }
    return 0;
}
```

```
Nhap so hang: 2
Nhap so cot: 3
```

```
Nhap phan tu A[1,1]: 1
Nhap phan tu A[1,2]: 2
Nhap phan tu A[1,3]: 3
Nhap phan tu A[2,1]: 4
Nhap phan tu A[2,2]: 5
Nhap phan tu A[2,3]: 6
```

```
MA TRAN DA NHAP
```

```
1 2 3
4 5 6
```

---

- Duyệt từng phần tử của dãy (có thể dùng vòng lặp **for**)
- Nếu phần tử xét thỏa mãn điều kiện thì:
  - Ghi nhận
- Chuyển sang xem xét phần tử tiếp theo

**Ví dụ:** Đếm số tháng có lượng mưa lớn hơn 50mm

```
int dem = 0;
for(i = 0; i < thang; i++)
    if(Mua[i] > 50)
        dem++;
printf("\nThang mua nhieu hon 50mm: %d", dem);
```

- Nhập mảng, đưa ra trung bình cộng các số chia hết cho 7

```
So phan tu cua mang (n<100) n = 6  
A[0] = 12  
A[1] = 14  
A[2] = 7  
A[3] = 8  
A[4] = 6  
A[5] = 21  
TBC so chia het cho 7: 14.00.  
-----  
Process exited after 15.05 seconds  
Press any key to continue . . .
```

```
#include<stdio.h>
int main(){
    int A[100], n, i, d = 0, S = 0;
    printf("So phan tu cua mang (n<100) n = "); scanf("%d",&n);
    for(i = 0; i < n; i++){
        printf("A[%d] = ", i);
        scanf("%d",&A[i]);
    }
    for(i = 0; i < n; i++)
        if(A[i] %7==0){
            d++;
            S+= A[i];
        }
    if(d > 0)
        printf("TBC so chia het cho 7: %7.2f.",(float)S/d);
    else
        printf("Trong day khong co so chia het cho 7.");
    return 0;
}
```

```
#include<stdio.h>
int main(){
    int A[100], n, i, d = 0, S = 0;
    printf("So phan tu cua mang (n<100) n = "); scanf("%d",&n);
    for(i = 0; i < n; i++){
        printf("A[%d] = ", i);
        scanf("%d",&A[i]);
    }
    for(i = 0; i < n; i++){
        if(A[i] %7==0){
            d++;
            S+= A[i];
        }
    }
    if(d > 0)
        printf("TBC so chia het cho 7: %7.2f.",(float)S/d);
    else
        printf("Trong day khong co so chia het cho 7.");
    return 0;
}
```

```
So phan tu cua mang (n<100) n = 5
A[0] = 14
A[1] = 7
A[2] = 23
A[3] = 28
A[4] = 45
TBC so chia het cho 7: 16.33.
-----
Process exited after 16.44 seconds
Press any key to continue . . .
```

Tìm phần tử lớn nhất (*nhỏ nhất*)

- Giả sử phần tử đó là phần tử đầu tiên
- Lần lượt so sánh với các phần tử còn lại
  - Nếu phần tử mới của dãy lớn hơn  $\Rightarrow$  coi đây là phần tử lớn nhất và tiếp tục so sánh với phần tử kế
  - Nếu không đúng, so sánh tiếp với phần tử kế

Ví dụ:

Tìm tháng có lượng mưa nhiều nhất trong năm?

```
max = Mua[0];
```

```
for(i = 1; i < thang; i++)
```

```
    if(Mua[i] > max)
```

```
        max = Mau[i];
```

```
printf("\n Luong mua nhieu nhat la: %d", max);
```

- Tìm kiếm các phần tử thỏa mãn điều kiện (*giống bài toán đếm*)
  - Dùng for duyệt toàn bộ
  - Nếu cần thiết, dùng thêm mảng ghi lại chỉ số

## Ví dụ:

Đưa ra danh sách các tháng có lượng mưa nhiều hơn 50mm.

```
printf("Thang co luong mua lon hon 500mm")  
for(i = 0; i < thang; i++)  
    if(Mua[i] > 50)  
        printf("\nThang %d", i+1);
```



- Tìm phần tử **đầu tiên** của danh sách thỏa mãn điều kiện
  - Dùng vòng lặp **for** kết hợp với **break**;
  - Dùng vòng lặp **while**

**Ví dụ:** Đưa ra phần tử đầu tiên của mảng có giá trị bằng k.

```
int A[100];  
int N, i, k, f;    //N: số phần tử, k phần tử cần tìm
```

Dùng vòng lặp for:

```
for(i = 0; i < N; i++)  
    if(A[i] == k) break;  
if(i < N) printf("Tim thay tai vi tri %d", i);
```

Dùng vòng lặp while:

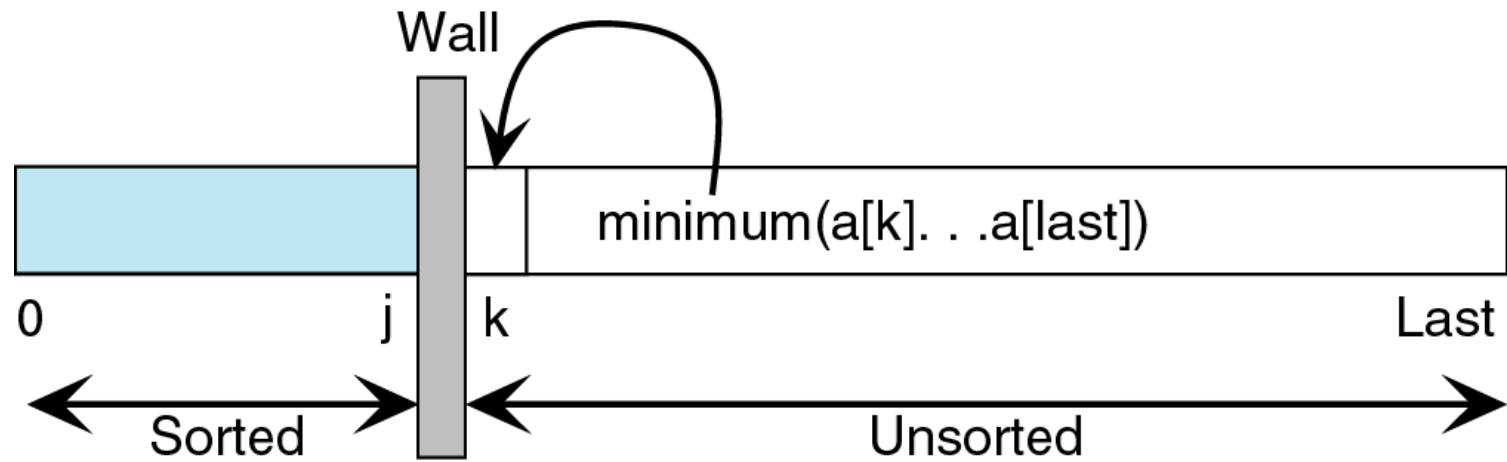
```
i=0; f =0;        //f: found, f = 1  $\Leftrightarrow$  k is found  
while(i < N && f==0){  
    if(A[i] == k)  
        f = 1;  
    else i++;  
}  
if (f==1) printf("Tim thay tai vi tri %d", i);
```

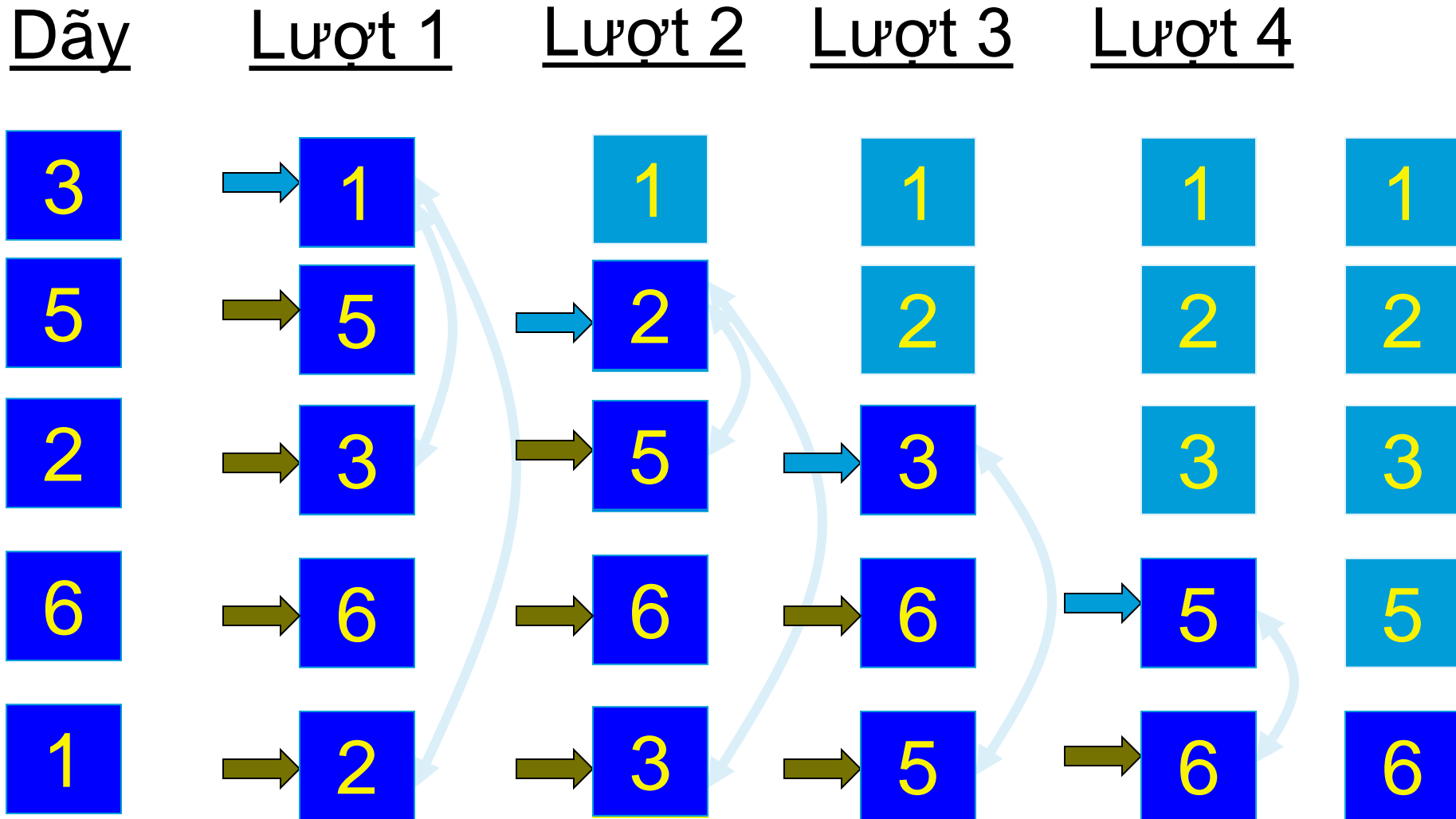
1. Nhập vào dãy  $n$  số ( $n < 100$ ), tính và đưa ra màn hình:
  - Tổng và tích của dãy số
  - Các số chia hết cho 3 và lớn hơn 10
  - Đếm các số có giá trị nằm trong đoạn  $[100, 1000)$
2. Nhập vào một dãy số, tìm số chẵn nhỏ nhất dãy
3. Viết chương trình nhập vào từ bàn phím một dãy số có số phần tử  $< 100$ . Tìm max và đưa ra vị trí những phần tử bằng max.

- Cho mảng phần tử, sắp xếp theo thứ tự tăng/giảm
- Các thuật toán
  - Sắp xếp thêm dần (Insertion sort)
  - Sắp xếp lựa chọn (Selection sort)
  - Sắp xếp nổi bọt (Bubble sort)
  - Sắp xếp vun đống (Heap sort)
  - Sắp xếp nhanh (Quick sort)
  - Sắp xếp trộn (Merge sort)

**Nguyên tắc:** Tại lượt sắp thứ  $k$ , tìm phần tử nhỏ nhất trong số các phần tử chưa được sắp xếp ( $[k..last]$ ) và đổi chỗ cho phần tử thứ  $k$  (có chỉ số  $k-1$ )

- Khi  $k = 1$ , phần tử thứ nhất (chỉ số 0) đúng vị trí
- Khi  $k = 2$ , phần tử thứ hai (chỉ số 1) đúng vị trí...





//Khai báo các biến

int A[100]; //Mảng chứa dữ liệu

int N, i, j, tmp;

//Sắp xếp

for(i = 0; i < N - 1; i++)

for(j = i + 1; j < N; j++)

if(A[i] > A[j])

{

tmp = A[i];

A[i] = A[j];

A[j] = tmp;

}

- Nhập vào từ bàn phím một mảng các số nguyên không quá 100 phần tử
- Hiển thị dãy số vừa nhập
- Sắp xếp dãy theo thứ tự giảm dần
- Hiển thị dãy tại mỗi lượt sắp xếp

```
So phan tu [N<100], N = 5
Hay nhap day so...
A[1] = 1
A[2] = 2
A[3] = 3
A[4] = 4
A[5] = 5

Day vua nhap...
   1   2   3   4   5
Sap xep day theo thuat toan lua chon:
Luot 1 :   5   1   2   3   4
Luot 2 :   5   4   1   2   3
Luot 3 :   5   4   3   1   2
Luot 4 :   5   4   3   2   1
-----
Process exited after 4.969 seconds with
Press any key to continue . . .
```



```

int main(){
    int A[100], N, i, j, t;
    printf("So phan tu [N<100], N = "); scanf("%d",&N);
    printf("Hay nhap day so...\n");
    for(i=0; i < N; i++){
        printf("A[%d] = ",i+1); scanf("%d",&A[i]);
    }
    printf("\nDay vua nhap...\n");
    for(i=0; i < N; i++)
        printf("%4d", A[i]);
    printf("\nSap xep day theo thuat toan lua chon:");
    for(i=0; i < N-1; i++){
        for(j=i+1; j < N; j++){
            if(A[i] < A[j]){
                t = A[i];
                A[i] = A[j];
                A[j] = t;
            }
        }
        printf("\nLuot %d : ",i+1);
        for(j=0;j < N; j++)
            printf("%4d", A[j]);
    }
}

```

So phan tu [N<100], N = 5

Hay nhap day so...

A[1] = 1

A[2] = 2

A[3] = 3

A[4] = 4

A[5] = 5

Day vua nhap...

1 2 3 4 5

Sap xep day theo thuat toan lua chon:

Luot 1 : 5 1 2 3 4

Luot 2 : 5 4 1 2 3

Luot 3 : 5 4 3 1 2

Luot 4 : 5 4 3 2 1

-----

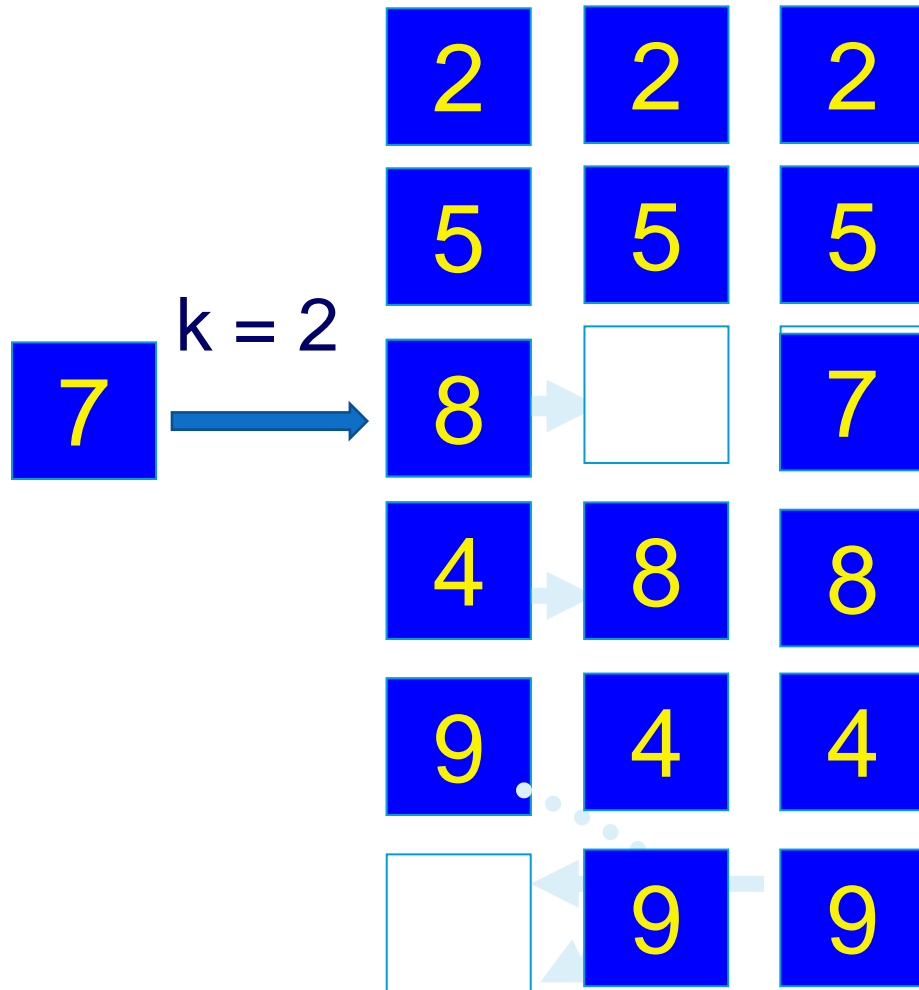
Process exited after 4.969 seconds with

Press any key to continue . . .

```
So phan tu [N<100], N = -12
So phan tu [N<100], N = 120
So phan tu [N<100], N = 5
Hay nhap day so...
A[1] = 1
A[2] = 4
A[3] = 5
A[4] = 7
A[5] = 2

Day vua nhap...
  1   4   5   7   2
Sap xep day theo thuat toan lua chon:
Luot 1 :    7   1   4   5   2
Luot 2 :    7   5   1   4   2
Luot 3 :    7   5   4   1   2
Luot 4 :    7   5   4   2   1
-----
Process exited after 13.97 seconds with return value 4
Press any key to continue . . .
```

```
int A[100], N, i, j, t;
do{
    printf("So phan tu [N<100], N = "); scanf("%d",&N);
}while(N<=0 || N>100);
printf("Hay nhap day so...\n");
for(i=0; i < N; i++){
    printf("A[%d] = ",i+1); scanf("%d",&A[i]);
}
printf("\nDay vua nhap...\n");
for(i=0; i < N; i++)
    printf("%4d", A[i]);
printf("\nSap xep day theo thuat toan lua chon:");
for(i=0; i < N-1; i++){
    for(j=i+1; j < N; j++)
        if(A[i] < A[j]){
            t = A[i];
            A[i] = A[j];
            A[j] = t;
        }
    printf("\nLuot %d : ",i+1);
    for(j=0;j < N; j++)
        printf("%4d", A[j]);
}
```



```
for(i = N; i > k; i--)
```

```
    A[i] = A[i-1];
```

```
A[k] = x;
```

```
N = N + 1;
```

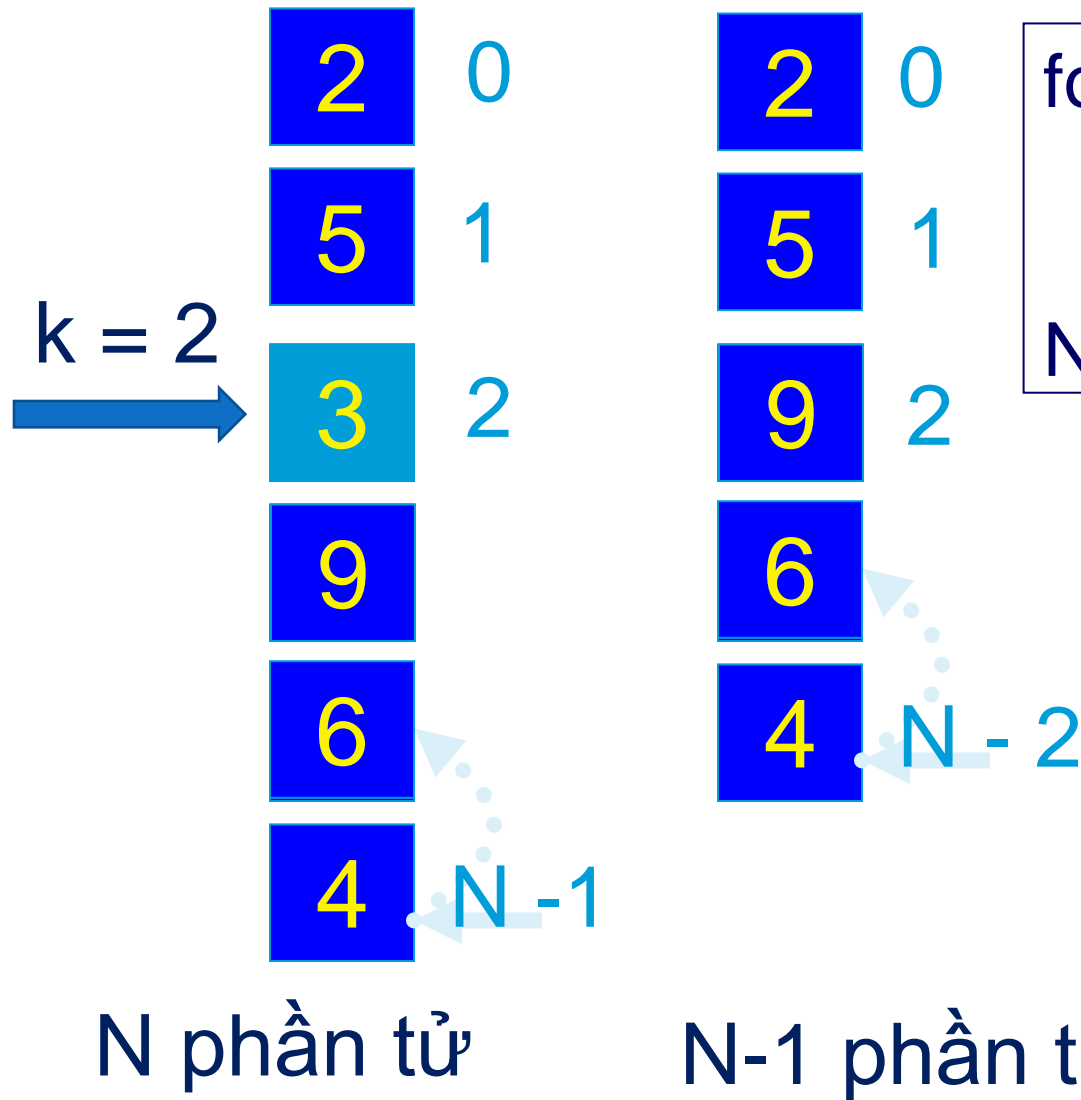
Chú ý:

N = MAX: không chèn được

k >= N: Chèn vào vị trí N

k < 0: Chèn vào vị trí 0

# Xóa phần tử ở vị trí $k$ ( $0 \leq k < N$ )



```
for(i = k+1; i < N; i++)  
    A[i-1] = A[i];  
N = N - 1;
```

1. Nhập vào dãy số có  $n$  phần tử ( $n < 100$ ). Nhập số  $x$  và số  $k$  nguyên. Chèn  $x$  vào vị trí  $k$  của dãy. Nếu  $k \geq n$  thì chèn  $x$  vào vị trí  $n$ .
2. Nhập vào một dãy ( $< 100$  phần tử); xóa đi các phần tử chia hết cho 5 và đưa kết quả ra màn hình.
3. Nhập vào một dãy số ( $< 100$  phần tử) và sắp xếp theo thứ tự tăng dần. Nhập thêm vào một số  $x$  và chèn số  $x$  vào đúng vị trí để dãy vẫn tăng dần.
4. Nhập vào từ bàn phím một dãy số nguyên ( $< 100$  phần tử). Sắp xếp dãy theo nguyên tắc: Bên trên là số chẵn chia hết cho 3. Bên dưới là số lẻ chia hết cho 3. Giữa là các số còn lại. Đưa cả 2 dãy ra màn hình.

# Bài tập 1

```

1  #include<stdio.h>
2  #define MAX 100
3  int main(){
4      int A[MAX], N, i;
5      do{
6          printf("So phan tu N = ");
7          scanf("%d",&N);
8      }while(N<0 || N >=100);
9
10     for(i = 0; i < N; i++){
11         printf("A[%d] = ",i);
12         scanf("%d", &A[i]);
13     }
14     printf("In day da nhap:\n");
15     for(i = 0; i < N; i++)
16         printf("%4d", A[i]);
17     int x, k;
18     printf("\nNhap so x = "); scanf("%d",&x);
19     printf("Nhap so nguyen k = "); scanf("%d",&k);
20     if (N == MAX)
21         printf("Mang day. Khong chen them duoc!");

```

```

22     else{
23         if (k<0) k = 0;
24         if (k>=0 && k<N)
25         {
26             for(i = N; i > k; i--)
27                 A[i] = A[i-1];
28             A[k] = x;
29             N++;
30         }
31         else if(k>=N)
32         {
33             A[N] = x;
34             N++;
35         }
36         printf("In lai day sau khi chen x = %d:\n", x);
37         for(i = 0; i < N; i++)
38             printf("%4d", A[i]);
39     }
40     return 0;
41 }

```

```

So phan tu N = 5
A[0] = 1
A[1] = 2
A[2] = 3
A[3] = 4
A[4] = 5
In day da nhap:
 1  2  3  4  5
Nhap so x = 0
Nhap so nguyen k = -1
In lai day sau khi chen x = 0:
 0  1  2  3  4  5
-----

```

```

So phan tu N = 6
A[0] = 1
A[1] = 2
A[2] = 3
A[3] = 4
A[4] = 5
A[5] = 6
In day da nhap:
 1  2  3  4  5  6
Nhap so x = 0
Nhap so nguyen k = 2
In lai day sau khi chen x = 0:
 1  2  0  3  4  5  6
-----

```

```

So phan tu N = 5
A[0] = 5
A[1] = 4
A[2] = 3
A[3] = 2
A[4] = 1
In day da nhap:
 5  4  3  2  1
Nhap so x = -2
Nhap so nguyen k = 6
In lai day sau khi chen x = -2:
 5  4  3  2  1 -2
-----

```

Nhập vào một dãy gồm  $n$  phần tử ( $0 < n < 100$ ). Xóa đi các phần tử chia hết cho 5 và đưa kết quả ra màn hình.

```
So phan tu N = 101
So phan tu N = 6
A[0] = 5
A[1] = 1
A[2] = 3
A[3] = 10
A[4] = 25
A[5] = 3
Day da nhap:
    5    1    3   10   25    3
Day sau khi xoa:
    1    3    3
-----
```



```
#include<stdio.h>
#define MAX 100
int main(){
    int A[MAX], N, i;
    do{
        printf("So phan tu N = ");
        scanf("%d",&N);
    }while(N<=0 || N >=100);

    for(i = 0; i < N; i++){
        printf("A[%d] = ",i);
        scanf("%d", &A[i]);
    }
    printf("Day da nhap:\n");
    for(i = 0; i < N; i++)
        printf("%4d", A[i]);
```

```
// giu lai cac phan tu khong chia het cho 5
{
    int d = 0;
    for(i = 0; i < N; i++){
        if(A[i] % 5 != 0){
            A[d] = A[i];
            d++;
        }
    }
    N = d;// cap nhat so phan tu sau xoa
}
printf("\nDay sau khi xoa:\n");
for(i = 0; i < N; i++)
    printf("%4d", A[i]);
return 0;
}
```

Nhập vào một dãy số (<100 phần tử) và sắp xếp theo thứ tự tăng dần.

Nhập thêm vào một số x và chèn số mới nhập vào đúng vị trí.

```
So phan tu N = 6
A[0] = 1
A[1] = 6
A[2] = 5
A[3] = 7
A[4] = 8
A[5] = 9
In day da nhap:
1 6 5 7 8 9
In lai day sau khi xep tang dan:
1 5 6 7 8 9
Nhap so x = 4
In lai day sau khi chen x = 4:
1 4 5 6 7 8 9
-----
```

```

1  #include<stdio.h>
2  #define MAX 100
3  int main(){
4      int A[MAX], N, i, j, t;
5      do{
6          printf("So phan tu N = ");
7          scanf("%d",&N);
8      }while(N<0 || N >=100);
9
10     for(i = 0; i < N; i++){
11         printf("A[%d] = ",i);
12         scanf("%d", &A[i]);
13     }
14     printf("Day da nhap:\n");
15     for(i = 0; i < N; i++)
16         printf("%4d", A[i]);
17
18     for(i = 0; i < N-1; i++)
19         for(j = i; j < N; j++)
20             if (A[i]>A[j]){
21                 t = A[i];
22                 A[i] = A[j];
23                 A[j] = t;
24             }
25     printf("\nDay xep tang dan:\n");
26     for(i = 0; i < N; i++)
27         printf("%4d", A[i]);

```

```

29     int x;
30     printf("\nNhap so x = "); scanf("%d",&x);
31     if (N == MAX)
32         printf("Mang day!");
33     else{
34         i = N;
35         while((i > 0) &&(A[i-1] > x)) {
36             A[i] = A[i-1];
37             i--;
38         }
39         A[i] = x;
40         N++;
41         printf("Day sau khi chen x = %d:\n", x);
42         for(i = 0; i < N; i++)
43             printf("%4d", A[i]);
44     }
45     return 0;
46 }

```

```

So phan tu N = 6
A[0] = 1
A[1] = 6
A[2] = 5
A[3] = 7
A[4] = 8
A[5] = 9
In day da nhap:
1  6  5  7  8  9
In lai day sau khi xep tang dan:
1  5  6  7  8  9
Nhap so x = 4
In lai day sau khi chen x = 4:
1  4  5  6  7  8  9
-----

```

Nhập vào từ bàn phím một dãy số nguyên (<100 phần tử).  
Sắp xếp dãy theo nguyên tắc: Bên trên là số chẵn chia hết cho 3. Bên dưới là số lẻ chia hết cho 3. Giữa là các số còn lại. Đưa cả 2 dãy ra màn hình.

```
So phan tu N = 10
```

```
A[0] = 1  
A[1] = 2  
A[2] = 3  
A[3] = 4  
A[4] = 5  
A[5] = 6  
A[6] = 9  
A[7] = 12  
A[8] = 15  
A[9] = 4
```

```
In day da nhap:
```

```
1 2 3 4 5 6 9 12 15 4
```

```
In lai day da sap theo yeu cau:
```

```
6 12 4 5 1 2 4 3 15 9
```

```
-----
```

```

1  #include<stdio.h>
2  int main(){
3      int A[100], N, i;
4      do{
5          printf("So phan tu N = ");
6          scanf("%d",&N);
7      }while(N<0 || N >=100);
8
9      for(i = 0; i < N; i++){
10         printf("A[%d] = ",i);
11         scanf("%d", &A[i]);
12     }
13     printf("\nIn day da nhap:\n");
14     for(i = 0; i < N; i++)
15         printf("%4d", A[i]);
16     //Sap xep so chan chia het cho 3 len truoc
17     int d = 0, t;
18     for(i = 0; i < N; i++)
19         if(A[i]%6==0){
20             t = A[i];
21             A[i] = A[d];
22             A[d] = t;
23             d++;
24         }

```

```

25     //Sap xep so le chia het cho 3 o cuoi day
26     for(i = d; i < N; i++)
27         if(A[i]%3 != 0){
28             t = A[i];
29             A[i] = A[d];
30             A[d] = t;
31             d++;
32         }
33     printf("\nIn lai day da sap theo yeu cau:\n");
34     for(i = 0; i < N; i++)
35         printf("%4d", A[i]);
36     return 0;
37 }

```

So phan tu N = 10

```

A[0] = 1
A[1] = 2
A[2] = 3
A[3] = 4
A[4] = 5
A[5] = 6
A[6] = 9
A[7] = 12
A[8] = 15
A[9] = 4

```

In day da nhap:

```

1  2  3  4  5  6  9 12 15  4
6 12  4  5  1  2  4  3 15  9
-----

```

1. Viết chương trình nhập vào một ma trận vuông, các phần tử nguyên, sau đó
  - Đưa ra ma trận tam giác dưới
  - Đưa ra ma trận tam giác trên
2. Nhập M, N ( $M, N < 30$ ) và một ma trận  $M \times N$ . Đưa ma trận ra màn hình
  - Tìm hàng/cột có tổng các phần tử lớn nhất
  - Tìm số lớn nhất/nhỏ nhất và vị trí trong ma trận
  - Đưa ra ma trận S cùng kích thước thỏa mãn

$$s_{i,j} = \begin{cases} 1 & \text{nếu } u_{i,j} > 0 \\ 0 & \text{nếu } u_{i,j} = 0 \\ -1 & \text{nếu } u_{i,j} < 0 \end{cases}$$

Viết chương trình nhập vào một ma trận vuông, các phần tử nguyên, sau đó

- Đưa ra ma trận tam giác dưới
- Đưa ra ma trận tam giác trên

```
Nhap kích thước: 3
Nhap phan tu [1,1] = 1
Nhap phan tu [1,2] = 2
Nhap phan tu [1,3] = 3
Nhap phan tu [2,1] = 4
Nhap phan tu [2,2] = 5
Nhap phan tu [2,3] = 6
Nhap phan tu [3,1] = 7
Nhap phan tu [3,2] = 6
Nhap phan tu [3,3] = 5
```

```
MA TRẬN DÃ NHẬP
1 2 3
4 5 6
7 6 5
```

```
MA TRẬN TAM GIÁC TRÊN
1 2 3
  5 6
    5
```

```
MA TRẬN TAM GIÁC DƯỚI
1
4 5
7 6 5
```

---

```
1  #include <stdio.h>
2  int main(){
3      int A[20][20], N, i, j;
4      printf("Nhap kích thước: "); scanf("%d",&N);
5      printf("\n");
6      for (i=0; i < N; i++ )
7          for(j=0; j < N; j++) {
8              printf("Nhap phan tu [%d,%d] = ", i+1,j+1);
9              scanf("%d", &A[i][j]);
10         }
11     printf("\nMA TRAN DA NHAP\n");
12     for(i=0;i < N; i++){
13         for(j=0;j < N; j++)
14             printf("%4d",A[i][j]);
15         printf("\n");
16     }
```

```
17     printf("\nMA TRAN TAM GIAC TREN\n");
18     for (i=0;i < N; i++ ){
19         for(j=0;j < N; j++)
20             if(j >= i)
21                 printf("%4d",A[i][j]);
22             else
23                 printf("%4c",32);
24         printf("\n");
25     }
26     printf("\n MA TRAN TAM GIAC DUOI\n");
27     for (i=0;i<N;i++){
28         for(j=0;j <= i;j++)
29             printf("%4d",A[i][j]);
30         printf("\n");
31     }
32     return 0;
33 }
```



Viết chương trình nhập vào 2 ma trận A và B gồm m hàng và n cột, các phần tử nguyên, sau đó:

- Tính và in ra ma trận tổng  $C = A+B$
- In ra ma trận chuyển vị của ma trận C.

```
Nhap so hang m = 2
Nhap so cot n = 3
Nhap ma tran A:
A[0][0] = 1
A[0][1] = 1
A[0][2] = 1
A[1][0] = 1
A[1][1] = 1
A[1][2] = 1
Nhap ma tran B:
B[0][0] = 2
B[0][1] = 1
B[0][2] = 3
B[1][0] = 4
B[1][1] = 5
B[1][2] = 2
Ma tran C = A + B:
3      2      4
3      6      3
Ma tran chuyen vi cua C:
3      3      4
2      6      3
1      3      3
```

# Ma trận tổng, chuyển vị

```

1  #include<stdio.h>
2  const MAX = 100;
3  int main(){
4      int A[MAX][MAX], B[MAX][MAX], C[MAX][MAX], m, n, i, j;
5      printf("Nhap so hang m = "); scanf("%d",&m);
6      printf("Nhap so cot n = "); scanf("%d",&n);
7      puts("Nhap ma tran A:");
8      for(i = 0; i < m; i++)
9          for(j = 0; j < n; j++)
10             {
11                 printf("A[%d][%d] = ",i,j);
12                 scanf("%d", &A[i][j]);
13             }
14      puts("Nhap ma tran B:");
15      for(i = 0; i < m; i++)
16          for(j = 0; j < n; j++)
17             {
18                 printf("B[%d][%d] = ",i,j);
19                 scanf("%d", &B[i][j]);
20             }

```

```

21      puts("Ma tran C = A + B:");
22      for(i = 0; i < m; i++)
23          {
24              for(j = 0; j < n; j++)
25                  {
26                      C[i][j] = A[i][j] + B [i][j];
27                      printf("%d \t", C[i][j]);
28                  }
29              printf("\n");
30          }
31      puts("Ma tran chuyen vi cua C:");
32      for(i = 0; i < n; i++)
33          {
34              for(j = 0; j < m; j++)
35                  printf("%d \t",C[j][i]);
36              printf("\n");
37          }
38      return 0;
39  }

```

# Ma trận tích $C = AB$

Viết chương trình nhập ma trận A gồm m hàng và k cột, ma trận B gồm k hàng và n cột, các phần tử nguyên, sau đó:

- Tính và in ra ma trận tích  $C = A.B$
- In ra ma trận chuyển vị của ma trận C.

```
Nhap m = 2
Nhap k = 3
Nhap n = 2
Nhap ma tran A:
A[0][0] = 1
A[0][1] = 1
A[0][2] = 2
A[1][0] = 3
A[1][1] = 4
A[1][2] = 5
Nhap ma tran B:
B[0][0] = 3
B[0][1] = 2
B[1][0] = 1
B[1][1] = 4
B[2][0] = 5
B[2][1] = 6
Ma tran C = A*B:
  14  18
  38  52
Ma tran chuyen vi cua C:
  14  38
  18  52
```

# Ma trận tích $C = AB$

Nhân hai ma trận ma trận A gồm m hàng và k cột, ma trận B gồm k hàng và n cột

$$C_{m \times n} = A_{m \times k} * B_{k \times n}$$

$$C = (c_{ij})_{m \times n}, c_{ij} = \sum_{t=0}^{k-1} a_{it} * b_{tj}$$
$$i = 0, 1, \dots, m - 1,$$
$$j = 0, 1, \dots, n - 1$$

# Ma trận tích $C = AB$

```

1  #include<stdio.h>
2  const MAX = 100;
3  int main(){
4      int A[MAX][MAX], B[MAX][MAX], C[MAX][MAX], m, k, n, i, j, t;
5      printf("Nhap m = "); scanf("%d",&m);
6      printf("Nhap k = "); scanf("%d",&k);
7      printf("Nhap n = "); scanf("%d",&n);
8      puts("Nhap ma tran A:");
9      for(i = 0; i < m; i++)
10         for(j = 0; j < k; j++)
11         {
12             printf("A[%d][%d] = ",i,j);
13             scanf("%d", &A[i][j]);
14         }
15     puts("Nhap ma tran B:");
16     for(i = 0; i < k; i++)
17         for(j = 0; j < n; j++)
18         {
19             printf("B[%d][%d] = ",i,j);
20             scanf("%d", &B[i][j]);
21         }

```

```

22     puts("Ma tran C = A*B:");
23     for(i=0;i<m;i++)
24     {
25         for(j=0; j<n;j++)
26         {
27             C[i][j] = 0;
28             for(t = 0; t<k;t++)
29                 C[i][j] += A[i][t]*B[t][j];
30             printf("%4d",C[i][j]);
31         }
32         printf("\n");
33     }
34     puts("Ma tran chuyen vi cua C:");
35     for(i = 0; i < n; i++)
36     {
37         for(j = 0; j < m; j++)
38             printf("%4d",C[j][i]);
39         printf("\n");
40     }
41     return 0;
42 }

```