# KỸ THUẬT LẬP TRÌNH

*GV: TRẦN THANH TUẤN*

# MẢNG (tt)

## ĐỊNH NGHĨA

Mảng là **danh sách các phần tử** có **cùng kiểu dữ liệu.**

## HÌNH ẢNH MINH HỌA

Một dãy các ô nhớ liền kề nhau:

| gt1 | gt2 | gt3 | gt4 | gt5 | gt6 | gt7 | gt8 | ... | gtN |
|---|---|---|---|---|---|---|---|---|---|

## KHAI BÁO

```
<kiểu dữ liệu> <tên biến>[HẰNG SỐ]; // khai báo
```

**Lưu ý:** HẰNG SỐ (số nguyên): kích thước mảng → cho biết số lượng phần tử tối đa mảng có thể chứa được.

```
<kiểu dữ liệu> <tên biến>[] = {gt1, gt2, ..., gtn}; // khai báo + khởi tạo
```

**Ví dụ:**

int a[100]; *// khai báo mảng tên a, chứa tối đa 100 phần tử là các số nguyên.*

int b[] = {1, 6, 3, 9}; *// khai báo mảng tên b, chứa 4 số nguyên lần lượt là: 1, 6, 3, 9.*

| **Chỉ số** | 0 | 1 | 2 | 3 |
|---|---|---|---|---|
| **Giá trị** | 1 | 6 | 3 | 9 |

## TRUY XUẤT CÁC PHẦN TỬ TRONG MẢNG

**Cú pháp:** <tên biến>[<chỉ số>]

Trong đó: *0 ≤ <chỉ số> ≤ (Số lượng phần tử trong mảng – 1)*

*Với mảng có tên **arr** hiện chứa **10** phần tử*

| ***Chỉ số*** | *0* | *1* | *2* | *3* | *4* | *5* | *6* | *7* | *8* | *9* |
|---|---|---|---|---|---|---|---|---|---|---|
| ***Giá trị*** | *34* | *2* | *12* | *43* | *19* | *76* | *89* | *56* | *28* | *30* |
| ***Truy xuất*** | *arr[0]* | *arr[1]* | *arr[2]* | *arr[3]* | *arr[4]* | *arr[5]* | *arr[6]* | *arr[7]* | *arr[8]* | *arr[9]* |

**Ví dụ:**

```
int a[100]; // khai báo mảng tên a, chứa tối đa 100 phần tử là các số nguyên.
a[0] = 1; // gán giá trị 1 cho phần tử đầu tiên của mảng a
a[2] = 6; // gán giá trị 6 cho phần tử thứ 3 của mảng a
cin >> a[5]; // nhập giá trị từ bàn phím cho phần tử thứ 6 của mảng a
cout << a[2]; // in giá trị của phần tử thứ 3 của mảng a ra màn hình
int tong = a[0] + a[5]; // tính tổng phần tử đầu tiên và phần tử thứ 6, sau đó gán giá trị tính được vào biến tong
```

## THAO TÁC DUYỆT MẢNG

**Sử dụng vòng lặp** (while, do...while, for)

Giả sử: **n** = ***<số lượng phần tử hiện có trong mảng>***;

```
for(int i = 0; i < n; i++)
{
      Truy xuất phần tử <tên biến mảng>[i]
}
```

**Ví dụ:**

***Nhập n phần tử vào mảng số nguyên a***

```
for (int i = 0; i < n; i++)
{
    cout << "Nhap a[" << i << "] = ";
    cin >> a[i];
}
```

***Xuất mảng số nguyên a (hiện có n phần tử)***

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

## HÀM VỚI THAM SỐ TRUYỀN VÀO LÀ MẢNG

Tham số mảng ➔ truyền tham chiếu *(giá trị bị thay đổi sau khi kết thúc hàm)*

**Ví dụ:**

***Định nghĩa hàm nhập n phần tử vào mảng số nguyên a***

```
void nhap_mang(int a[], int n)
{
    for (int i = 0; i < n; i++)
    {
        cout << "Nhap a[" << i << "] = ";
        cin >> a[i];
    }
}
```

***Định nghĩa hàm xuất mảng số nguyên a (hiện có n phần tử)***

```
void xuat_mang(int a[], int n)
{
    for (int i = 0; i < n; i++)
    {
        cout << "a[" << i << "] = " << a[i] << endl;
    }
}
```

***Thực hiện gọi hàm:***

```
void main()
{
    // Khai báo mảng
    int b[100];
    int so_pt;

    do
    {
        cout << "Nhap so luong phan tu: ";
        cin >> so_pt;
    } while (so_pt < 0 || so_pt > 100);

    // Nhập mảng
    nhap_mang(b, so_pt);

    // Xuất mảng
    cout << "Mang vua nhap: \n";
    xuat_mang(b, so_pt);

    system("pause");
}
```

# BÀI TẬP

Viết các hàm sau:

1. Nhập mảng, xuất mảng số nguyên a có n phần tử.

   *void nhap_mang(int[], int);*

   *void xuat_mang(int[], int);*

2. Tính tổng các phần tử của mảng

   *int tong_mang(int[], int);*

3. Tính tổng các phần tử có giá trị chẵn trong mảng

   *int tinh_tong_pt_chan(int[], int);*

4. Tính tổng các phần tử tại vị trí (chỉ số) lẻ trong mảng

   *int tinh_tong_pt_vt_le(int[], int);*

5. Tính tổng các số dương trong mảng

   *int tinh_tong_pt_duong(int[], int);*

6. Tính tổng các số âm trong mảng

   *int tinh_tong_pt_am(int[], int);*

7. Tính tổng các phần tử là số nguyên tố

   *int tinh_tong_pt_nguyen_to(int[], int);*

8. Tính tổng các phần tử là số hoàn thiện

   *int tinh_tong_pt_hoan_thien(int[], int);*

9. Tính trung bình cộng các giá trị của mảng

*float tinh_trung_binh_cong_mang(int[], int);*

10. Cho biết giá trị X có trong mảng hay không? *(nếu có trả về vị trí đầu tiên tìm thấy, ngược lại trả về -1)*

    *int tim_x(int[], int, int);*

11. Tìm giá trị lớn nhất trong mảng

    *int tim_gt_lon_nhat(int[], int);*

12. Đếm số lần xuất hiện của phần tử X trong mảng

    *int dem_x(int[], int, int);*

13. Cho biết vị trí cuối cùng của giá trị X trong mảng. Nếu X không có trong mảng thì trả về -1.

    *int tim_vt_x_cuoi_cung(int[], int, int);*

14. Tìm giá trị nhỏ nhất trong mảng.

    *int tim_gt_nho_nhat(int[], int);*

15. Tìm giá trị dương nhỏ nhất trong mảng. Nếu không có trả về -1.

    *int tim_gt_duong_nho_nhat(int[], int);*

16. Tìm giá trị âm lớn nhất trong mảng. Nếu không có trả về 0

    *int tim_gt_am_lon_nhat(int[], int);*

17. Sắp xếp mảng theo thứ tự các giá trị tăng dần

    *void sap_xep_tang_dan(int[], int);*

18. Sắp xếp mảng theo thứ tự các giá trị giảm dần

    *void sap_xep_giam_dan(int[], int);*

19. Sắp xếp các phần tử có giá trị chẵn tăng dần

*void sap_xep_chan_tang_dan(int[], int);*

20. Sắp xếp các phần tử có giá trị lẻ giảm dần

    *void sap_xep_le_giam_dan(int[], int);*

21. Thêm phần tử X vào cuối mảng

    *void them_cuoi(int[], int&, int);*

22. Thêm phần tử X vào đầu mảng

    *void them_dau(int[], int&, int);*

23. Thêm phần tử X vào vị trí K (với 0 <= K < N)

    *void them_vt_k(int[], int&, int, int);*

    ***Lưu ý:*** *Nếu số lượng phần tử hiện có trong mảng bằng kích thước mảng thì không thể thêm giá trị vào mảng.*

24. Xoá phần tử ở cuối mảng

    *void xoa_cuoi(int[], int&);*

25. Xóa phần tử ở đầu mảng

    *void xoa_dau(int[], int&);*

26. Xóa phần tử ở vị trí K (với 0 <= K < N)

    *void xoa_vt_k(int[], int&, int);*