

CONFIDENTIAL

# C Programming Introduction

## week 10: Arrays

For HEDSPI Project

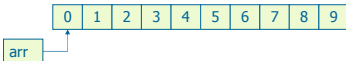
## Mảng

- 1 vùng bộ nhớ quản lý các biến cùng kiểu.
- Mảng có thể khai báo với kiểu bất kỳ
  - Vd: `int Arr[10]` là mảng 10 số nguyên.
- VD:
  - Liệt kê điểm sinh viên
  - Liệt kê các số do người dùng nhập

## Cách lưu mảng trong bộ nhớ

- Chuỗi các biến với kiểu nhất định
- Mỗi biến mảng lưu địa chỉ bắt đầu của mảng trong bộ nhớ
- VD:

```
int arr[10];
```



- Phần tử thứ n được gọi đến qua `arr[n-1]`  
(phần tử đầu là 0)

## Khởi tạo

- Giống như các biến khác, có thể khởi tạo các giá trị biến mảng khi khai báo
- Số các giá trị khởi tạo:
  - Không thể > số phần tử của mảng
  - Có thể < số phần tử của mảng
  - Các phần tử không được khởi tạo sẽ có giá trị 0

## Khởi tạo

- Kích thước mảng có thể suy ra từ số các phần tử khởi tạo nếu giá trị trong [] bị bỏ trống
  - 2 khai báo sau tương đương:
    - `int array1 [8] = {2, 4, 6, 8, 10, 12, 14, 16};`
    - `int array2 [] = {2, 4, 6, 8, 10, 12, 14, 16};`

## VD: Khởi tạo với câu lệnh for

- Tạo mảng gồm 20 số chẵn từ 2 đến 20. In các số có trong mảng.

```
#include <stdio.h>
#define arraySize 10

int main()
{
    int s[ arraySize ]; // mảng 10 phần tử
    int i;
    for ( i = 0; i < arraySize; i++ )
        s[ i ] = 2 + 2 * i;
    printf("Element \t Value\n");
    for ( i = 0; i < arraySize; i++ )
        printf("%d\t%d\n", i, s[i]);
    return 0;
}
```

## Vào/ra dữ liệu cho mảng

- Chạy 1 vòng lặp, tại mỗi lần lặp:
  - Dùng hàm nhập liệu như scanf
- Hoặc
  - Dùng hàm in dữ liệu như printf

month	rainfall (in mm)
1	40
2	45
3	95
4	130
5	220
6	210
7	185
8	135
9	80
10	40
11	45
12	30

*table of rainfall*

## Example

```
#include <stdio.h>
#define MONTHS 12

/* store and display rainfall in all months of the year */

int main()
{
    int rainfall[MONTHS];
    int i;

    for ( i=0; i < MONTHS; i++){
        printf("Enter the rainfall(mm):"); scanf("%d", &rainfall[i] );
    }
    /* Print from January to December */
    for ( i=0; i < MONTHS; i++ ) {
        printf( "%5d ", rainfall[i]);
    }
    printf("\n");
    return 0;
}
```

## Bài 10.1

Viết chương trình nhập 1 mảng lưu 100 số nguyên

- Tính tổng các số lẻ trong dãy
- Tìm giá trị nhỏ nhất

## Bài 10.2

- Cho 1 mảng với dữ liệu nhập từ bàn phím. Tính tổng các giá trị lớn nhất cục bộ của dãy (giá trị lớn nhất cục bộ là giá trị lớn nhất so với 2 giá trị bên cạnh nó)

## Sử dụng mảng như tham số hàm

- Các hàm có thể nhận mảng làm tham số
- Thông thường, kích thước mảng cũng cần là 1 tham số của hàm

## Sử dụng mảng như tham số hàm

- VD:  

```
int calc_sum(int arr[], int size);
```
- Trong hàm này, **arr** được truy cập như bình thường
- Các thay đổi trong hàm sẽ tác động lên mảng gốc

## Example

```
int calc_sum(const int arr[], int size)
{
    int i = 0;
    int sum = 0;

    for (i = 0; i < size; ++i)
        sum += arr[i];

    return sum;
}
```

## Bài 10.3

- Viết 1 hàm nhận 1 mảng chứa 2 số nguyên và trả về 1 nếu 2 số bằng nhau, 0 nếu ngược lại
- Viết chương trình nhận 2 mảng chứa các số nguyên và kiểm tra xem 2 mảng đó có bằng nhau hay không

## Bài 10.4

- Viết 2 hàm:
  - Hàm 1 sắp xếp các số nguyên trong mảng theo thứ tự giảm dần
  - Hàm 2 sắp xếp các số lẻ trong mảng theo thứ tự giảm dần
- Viết chương trình yêu cầu người dùng nhập 10 số nguyên và hiển thị kết quả của 2 cách sắp xếp trên

## Bài 10.5

- Cho mảng các số nguyên:
  - a) Đếm số các số 0 trong mảng
  - b) Tìm độ dài của dãy con = 0 lớn nhất trong mảng
  - c) Tìm số lần xuất hiện của các số trong mảng

## Mảng nhiều chiều

- Mảng của mảng:  

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

1	2	3
4	5	6
- Là mảng chứa 2 mảng nguyên, mỗi mảng có 3 phần tử
- Cách truy cập: phần tử thứ j của mảng thứ i  

```
A[i][j]
```

## Example: Matrix addition

```
#include <stdio.h>
#define SIZE 3

int main()
{
    int A[SIZE] = {1,2,3}, B[SIZE] = {4,5,6}, C[SIZE] = {7,8,9};
    int B[SIZE] = {1,1,1}, C[SIZE] = {2,2,2}, D[SIZE] = {3,3,3};
    int i = 0, j = 0;

    for (i = 0; i < SIZE; ++i)
        for (j = 0; j < SIZE; ++j)
            C[i][j] = A[i][j] + B[i][j];

    return 0;
}
```



### Bài 10.6

- Viết chương trình định nghĩa 3 ma trận A, B, C kích thước 3x3 với các phần tử kiểu int. Khởi tạo 2 ma trận A, B.
- Tính ma trận  $C = A * B$
- In cả 3 ma trận ra màn hình



### Bài 10.7

- Nhập mảng n phần tử, với n nhập từ bàn phím. Kiểm tra xem mảng có đối xứng hay không



### Bài 10.8

- Viết hàm đảo ngược các phần tử của mảng.
- Trong chương trình lớn, yêu cầu người dùng nhập 1 mảng thực. Sau đó dùng hàm trên để đảo ngược mảng mà không dùng mảng phụ