

CONFIDENTIAL

C Programming Introduction

week 12: Arrays and Pointers

For HEDSPI Project

Con trỏ và mảng

- Nhắc lại: 1 mảng `s` lưu địa chỉ phần tử đầu tiên của nó `s[0]`
- `s` thực chất là con trỏ trỏ đến `s[0]`

```
int s[10];  
int *iptr;  
iptr=s; /* Từ bây giờ iptr tương đương với s */
```
- Cả `iptr` và `s` giờ đều trỏ đến `s[0]`

Mảng con trỏ tương đương

- Mảng thực chất là 1 dạng của con trỏ
- Khi định nghĩa mảng, 1 vùng nhớ cố định (kích thước mảng) được giữ cho nó
 - Biến mảng trỏ đến địa chỉ đầu của vùng nhớ khai báo
- Khi khai báo 1 con trỏ, nó chưa được khởi tạo (giống 1 biến thường)

Các phép toán đối với con trỏ

- Có thể tăng, giảm biến trỏ
- Nếu `p` là con trỏ trỏ đến 1 kiểu cụ thể, `p+1` sẽ trỏ đến **địa chỉ của biến tiếp theo cùng kiểu**
- `p++`, `p+i`, và `p += i` đều có nghĩa

Các phép toán đối với con trỏ

- Nếu `p` và `q` trỏ đến các phần tử trong 1 mảng, `q-p` là số các phần tử giữa `p` và `q`
- Các phép toán số học với con trỏ khác các phép toán số học thông thường

Pointer arithmetic - example

```
int main(void)
{
    int a[3] = {17, 289, 4913}, *p, *q;

    p = a; /* p trỏ đến phần tử đầu tiên của a, nghĩa là &a[0] */
    q = p+2; /* q trỏ đến a[2]. Tương đương với q = &a[2] */

    printf("a = %p\n", a);
    printf("p = %p, q = %p\n", p, q);
    printf("p trỏ đến %d và q trỏ đến %d\n", *p, *q);
    printf("Khoảng cách con trỏ giữa p và q là %d\n", q-p);
    printf("Khoảng cách nguyên giữa p và q là %d\n",
           (int)q-(int)p);
    return 0;
}
```

a is 0012FECC
p is 0012FECC, q is 0012FED4
p points to 17 and q points to 4913
The pointer distance between p and q is 2
The integer distance between p and q is 8

Truyền mảng cho hàm

- Có thể truyền mảng cho hàm dùng con trỏ
- Thực chất là truyền địa chỉ phần tử đầu tiên của mảng cho hàm (cụ thể là con trỏ trỏ đến mảng).
- Hàm tính tổng các phần tử của mảng.

```
#include <stdio.h>
int addNumbers(int *fiveNumber){
    int i, sum=0;
    for(i=0; i<5; i++, fiveNumbers++){
        sum+= *fiveNumbers
    }
    return sum;
}
```

Bài 12.1

- Viết hàm countEven(int*, int) với đầu vào là 1 mảng nguyên và kích thước của nó, trả về số phần tử của mảng.

Bài 12.2

- Viết hàm trả về con trỏ trỏ đến giá trị lớn nhất của mảng các số double. Nếu mảng rỗng, trả về NULL.
double* maximum(double* a, int size);

Bài 12.3

- Viết hàm getSale dùng con trỏ để truyền vào địa chỉ của mảng. Hàm hỏi người dùng nhập vào các doanh số bán hàng và lưu chúng trong mảng.
- Viết hàm totalSale trả về tổng các giá trị trong mảng.
- Dùng 2 hàm này trong chương trình để nhập các doanh số bán hàng từ các đơn vị khác nhau và hiển thị tổng của từng đơn vị. Dùng con trỏ thay vì mảng khi truyền tham số.

Bài 12.4

- Viết chương trình liệt kê tất cả các mảng con trong 1 mảng. Ví dụ, mảng 1 3 4 2 có các mảng con sau:

```
1
1 3
1 3 4
1 3 4 2
3
3 4
3 4 2
4
4 2
2
```

Bài 12.5

- Viết chương trình đảo ngược 1 mảng theo 2 cách: sử dụng con trỏ và sử dụng chỉ số mảng