


Trường ĐH CNTP TP.HCM Khoa: CNTT Bộ môn: CNPM Môn: TH NMLT	BÀI 5. MẢNG 1 CHIỀU	
---	----------------------------	---

A. MỤC TIÊU:

- Khai báo được cấu trúc dữ liệu mảng 1 chiều.
- Phân tích được tình huống sử dụng mảng 1 chiều để lưu trữ dữ liệu.
- Cài đặt được các hàm nhập, xuất, xử lý dữ liệu trong các thao tác tính tổng, đếm giá trị phần tử.
- Áp dụng được mảng 1 chiều để xây dựng cách giải bài toán đơn giản trên tập hợp.

B. DỤNG CỤ - THIẾT BỊ THÍ NGHIỆM CHO MỘT SV:

STT	Chủng loại – Quy cách vật tư	Số lượng	Đơn vị	Ghi chú
1	– Máy tính (có thể là máy bàn hoặc laptop).	1	cái	

C. VẬT LIỆU

- Visual C++.Net.
- Phần mềm hỗ trợ dạy học: NetOp School hoặc NetSupport School.

D. NỘI DUNG THỰC HÀNH

I. Tóm tắt lý thuyết

1. Định nghĩa và khai báo mảng 1 chiều:

- Khái niệm:

Mảng một chiều là một nhóm các phần tử có cùng kiểu dữ liệu và cùng một tên, các phần tử trong mảng được lưu liên tiếp với nhau trong bộ nhớ máy tính.

- Khai báo:

<Kiểu dữ liệu> <Tên mảng>[<Số phần tử tối đa của mảng>];

Lưu ý: Chỉ số mảng luôn luôn bắt đầu từ 0.

Ví dụ:

int a[100]; //Mảng có tên a chứa tối đa 100 số nguyên

#define MAX_B 30

float b[MAX_B]; // Mảng có tên b chứa tối đa 30 số thực

const int MAX_C = 50;

char c[MAX_C]; Mảng có tên c chứa tối đa 50 kí tự

Ví dụ: Cho mảng 1 chiều tên a chứa 10 số nguyên như sau:

Chỉ số mảng	0	1	2	3	4	5	6	7	8	9
Mảng a	7	4	10	15	90	30	3	21	56	80

Phản tử a[3]

Trong đó:

- i là chỉ số của mảng ($i \in [0,9]$).
- a[i] là giá trị của phần tử thứ i

2. Tham số mảng của một hàm

```
<Kiểu dữ liệu> <Tên hàm> (<kiểu dữ liệu mảng> <Tên mảng>[], int <ten so phan tu>)\n{\n    //Khởi tạo các biến\n    //Duyệt mảng để tính toán\n    //Trả về giá trị\n}
```

3. Những thao tác trên mảng:

Xử lý mảng là việc xử lý nhóm các phần tử trong mảng lần lượt từng phần tử một.

- Duyệt theo chiều xuôi
 - Duyệt mảng không có điều kiện:

```
for (int i = 0; i < n; i++)\n{\n    //xử lý a[i];\n}
```

Ví dụ:

```
for(int i = 0; i < 5; i++)\n    printf("%5d", a[i]);
```

- Duyệt mảng có điều kiện

```
for (int i = 0; i < n; i++)\n{\n    if (<Điều kiện>)\n        //xử lý a[i];\n}
```

Ví dụ:

```
int s=0;\nfor(int i = 0; i < 5; i++)\n    if(a[i]%2 == 0)\n        s = s + a[i];\nprintf("\\n Tong cac so chan trong mang la: %d", s);
```

– Duyệt theo chiều ngược

- Duyệt mảng không có điều kiện:

```
for(int i=n-1;i>=0;i--)
{
    //xử lý a[i];
}
```

- Duyệt mảng có điều kiện

```
for(int i=n-1;i>=0;i--)
{
    if(<Điều kiện>)
        //xử lý a[i];
}
```

II. Bài tập mẫu hướng dẫn thực hành.

1. Viết chương trình nhập các giá trị cho mảng 1 chiều chứa các số nguyên, sau đó xuất mảng vừa nhập (Lưu ý: Viết chương trình dạng hàm).

Yêu cầu:

- Viết 2 hàm Nhập và Xuất mảng 1 chiều chứa các số nguyên.
- Gọi thực hiện 2 hàm này trong hàm main.

Hướng dẫn:

- Khai báo thêm các thư viện cơ bản cho chương trình.

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

- Khai báo hằng số cho chương trình.

```
#define MAX_SIZE 100
```

- Viết các định nghĩa hàm nhập mảng, xuất mảng của chương trình như sau:

```
voidNhapMang1C_SoNguyen(int a[], int &n)
{
    do
    {
        printf("Cho biet so phan tu cua mang: ");
        scanf("%d", &n);
    }while(n <= 0);
    for(int i=0; i<n; i++)
    {
        printf("Gia tri phan tu a[%d]=", i);
        scanf("%d", &a[i]);
    }
}
```

```
void XuatMang1C_SoNguyen(int a[], int n)
{
    for(int i=0; i<n; i++)
        printf("%4d", a[i]);
}
```

- Viết hàm main để thực thi chương trình.

```
void main()
{
    int A[MAX_SIZE];
    int N;
    NhapMang1C_SoNguyen(A, N);
    printf("\nNoi dung cua mang la: ");
    XuatMang1C_SoNguyen(A, N);
    getch();
}
```

2. Bổ sung vào bài 1 các thao tác sau (Lưu ý: Viết chương trình dạng hàm):

- Xuất các giá trị chẵn của mảng ra màn hình.
 - Tính tổng các giá trị mảng.
 - Tính tổng chẵn trong mảng
- Hướng dẫn:
- Bổ sung các định nghĩa hàm cần thiết cho chương trình như sau:

```
void Xuat_SoChan(int a[], int n)
{
    // SV tự làm
}
int TongMang (int a[], int n)
{
    // SV tự làm }
int TongChan (int a[], int n)
{
    // SV tự làm
}
```

- Bổ sung vào hàm main để thực thi chương trình.

```
void main()
{
    int a[MAX_SIZE], n;
    //*****
```

```

TaoMang1C_SoNguyen(a, n);
printf("\nNoi dung cua mang vua tao la:");
XuatMang1C_SoNguyen(a, n);
//*****
Xuat_SoChan(a, n);
int t = TongMang(a, n);
printf("\Tong cua mang la:%d", t);
int tc = TongChan(a, n);
printf("\Tong cua cac gia tri chan trong mang la:%d", tc);
}

```

III. Bài tập ở lớp

Cho một mảng 1 chiều chứa các số nguyên. Hãy viết một chương trình cho phép người dùng chọn một trong các yêu cầu sau:

1. Tạo một mảng 1 chiều chứa n số nguyên

Gợi ý: làm 2 cách

Cách 1: nhập giá trị phần tử mảng.

Cách 2: dùng lệnh **rand()** để sinh số nguyên ngẫu nhiên (**rand()** thuộc thư viện **stdlib.h**).

2. Xuất một mảng 1 chiều chứa n số nguyên.
3. Xuất ra các phần tử ở vị trí chẵn trong mảng.
4. Đếm phần tử dương trong mảng.
5. Tổng các phần tử âm trong mảng.
6. Tìm vị trí của giá trị x trong mảng, x nhập từ phím.
7. Tìm phần tử lớn nhất/nhỏ nhất trong mảng.
8. Kiểm tra mảng số nguyên có tồn tại giá trị x hay không?
9. Thêm vào mảng một giá trị x tại vị trí k (*nhập từ bàn phím*).
10. Tính tổng các phần tử chẵn/lẻ.
11. Kiểm tra mảng có toàn số chẵn không?
12. Tạo mảng a chứa giá trị chẵn/lẻ ngẫu nhiên
13. Kiểm tra a có chứa các giá trị chẵn lẻ xen kẽ không?
14. Xuất các phần tử tròn chục.
15. Thêm phần tử x sau phần tử đầu tiên.
16. Thêm phần tử tại vị trí k.
17. Thêm phần tử x sau phần tử max.
18. Xóa phần tử có giá trị x (xuất hiện đầu tiên).
19. Xóa vào mảng một giá trị x tại vị trí k (*nhập từ bàn phím*).
20. Xóa tất cả các phần tử có giá trị x.

IV. Bài tập về nhà:

Cho một mảng 1 chiều chứa các số thực. Hãy viết một chương trình cho phép người dùng chọn một trong các yêu cầu sau:

1. Tạo một mảng 1 chiều chứa n số thực (nhập từ bàn phím và sinh tự động ngẫu nhiên).

2. Xuất một mảng 1 chiều chứa n số thực.
3. Tính giá trị trung bình các phần tử trong mảng.
4. Tính tổng các giá trị dương trong mảng.
5. Đảo ngược các giá trị mảng.
6. Kiểm tra mảng có tính đối xứng hay không?
7. Viết hàm kiểm tra mảng có phải là mảng tăng không?
8. Viết hàm tìm giá trị phần tử xuất hiện nhiều nhất trong mảng, nếu có nhiều phần tử có số lần xuất hiện bằng nhau thì xuất phần tử đầu tiên.
9. Đếm số nguyên tố trong mảng.
10. Tìm số chẵn lớn nhất trong mảng.
11. Tìm số nguyên tố nhỏ nhất trong mảng.
12. Đếm số nguyên tố trong mảng.
13. Tìm “vị trí của giá trị chẵn đầu tiên” trong mảng 1 chiều các số nguyên. Nếu mảng không có giá trị chẵn nào thì trả về -1.
14. Tìm “số chẵn cuối cùng” của mảng. Nếu mảng không có giá trị chẵn thì trả về -1.
15. Đảo ngược các giá trị mảng.
16. Xuất các phần tử lớn hơn các phần tử xung quanh nó.
17. Tìm số lớn thứ 2 trong mảng.
18. Tìm số chẵn lớn nhất.
19. Đếm xem mảng có bao nhiêu số lớn nhất.
20. Xóa phần tử max/min

-Hết-