



BÀI THỰC HÀNH SỐ 9

1. MỤC TIÊU

- Hiểu và cài đặt được các hàm nhập, xuất, liệt kê trên cấu trúc mảng 1 chiều trong chương trình.

2. LÝ THUYẾT CẦN GHI NHỚ

2.1. Khai báo mảng một chiều

2.1.1. Khai báo tường minh (số phần tử xác định)

- Cú pháp **<kiểu cơ sở> <tên mảng> [<số phần tử>]**
- Ví dụ
 - `int songuyen [10] ;`
 - `float float sothuc [15] ;`
 - `char daykytu [30] ;`
 - `const int MAX=500;`
`int diem[MAX];`
 - Hay `const int MAX=8;`
`char A[MAX];`

2.1.2. Khai báo không tường minh (số phần tử không xác định)

- Cú pháp **<kiểu cơ sở> <tên mảng> []**
Kiểu khai báo không cho biết rõ số phần tử của mảng thường được áp dụng trong các trường hợp: vừa khai báo vừa gán giá trị, hoặc khai báo mảng là tham số hình thức của hàm.
- Trường hợp 1. Vừa khai báo vừa gán giá trị. Theo mặc định C sẽ hiểu số phần tử của mảng là số giá trị mà chúng ta gán cho mảng trong cặp dấu {}.
 - Cú pháp:
<Kiểu> <Tên mảng> [] = {Các giá trị cách nhau bởi dấu phẩy}
 - Ví dụ `float x[] = {12.1 , 7.23 , 5.0 , 27.6 , 87.9 , 9.31};`
khi đó : `x[0]=12.1 , x[1]=7.23 , x[2]=5.0 , x[3]=27.6 , x[4]=87.9 ,`
`x[5]=9.31`

Có thể sử dụng hàm **sizeof()** để biết số phần tử của mảng như sau:

Số phần tử = sizeof(tên mảng) / sizeof(kiểu)

- Trường hợp 2. Khai báo mảng là tham số hình thức của hàm,
 - Trong trường hợp này, không cần chỉ định số phần tử của mảng là bao nhiêu.
 - Ví dụ `void nhapmang (int a[], int n)`



2. 2. Truy cập vào các phần tử của mảng

- Mỗi phần tử của mảng được truy xuất thông qua **Tên biến mảng** theo sau là **chỉ số** nằm trong cặp dấu ngoặc vuông `[]`.
- Ví dụ

```
float t=10.0;
int i=1;
//gán giá trị trực tiếp cho biến mảng
a [0] = 4.2;
//gán giá trị cho biến mảng thông qua 1 biến khác
a [2]=t;
/*gán giá trị cho biến mảng thông qua các biến khác trong cùng mảng*/
a[i] = (a[0] + a[2])/2;
a[2*i+1] = a[2*i] + a[2*i+2];
/*gán giá trị cho biến mảng từ giá trị nhập từ bàn phím */
printf("\nGia tri : %f", a[1]);
scanf("%f ", &a[4]);
//gán giá trị của biến mảng cho biến khác
t= a [4];
```

3. BÀI TẬP THỰC HÀNH CƠ BẢN

Bài 1. Viết chương trình dưới dạng các hàm chức năng để thực hiện các yêu cầu sau:

- Viết hàm nhập số nguyên n ($0 < n \leq 100$). Nếu nhập sai, yêu cầu nhập lại cho đến khi nhập đúng. Hàm trả về giá trị của n vừa nhập.
- Viết hàm nhập vào mảng một chiều các số nguyên A gồm n phần tử.
- Viết hàm xuất mảng số nguyên A gồm n phần tử vừa nhập ở trên.
- Viết hàm main() kết hợp 3 hàm trên. Khi chương trình chạy ổn định mới viết tiếp hàm khác.
- Viết hàm in ra các số lẻ có trong mảng tính tổng các phần tử có trong mảng A. Thực hiện gọi hàm tính tổng trong hàm main().
- Viết hàm tính tổng các phần tử có trong mảng A. Hàm trả về giá trị tổng vừa tính được. Thực hiện gọi hàm tính tổng trong hàm main() để in ra kết quả.
- Viết hàm nhận tham số là số nguyên k tính tổng các phần tử có trong mảng A. Hàm trả về giá trị tổng vừa tính được. Thực hiện gọi hàm tính tổng trong hàm main() để in ra kết quả.

Hướng dẫn

- (i). Viết hàm nhập số phần tử của mảng ($0 < n \leq 100$).

```
void NhapN(int &n)
{
    do
    {
        printf("Nhap so phan tu cua mang(tu 1 - 100): ");
        scanf("%d", &n);
        if ((n <= 0) || (n > 100))
```



```
        printf("Chi nhan cac so tu 1 den 100.");
    } while ((n <= 0) || (n>100));
}
```

(ii). Viết hàm nhập vào mảng một chiều các số nguyên gồm n phần tử.

```
void NhapMang(int a[], int n)
{
    for (int i = 0; i<n; i++)
    {
        printf("Nhap a[%d]: ", i);
        scanf("%d", &a[i]);
    }
}
```

(iii). Viết hàm xuất mảng số nguyên n phần tử vừa nhập ở trên

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

(iv). Viết hàm main() kết 3 hàm trên. Khi chương trình đã chạy ổn định rồi mới viết tiếp hàm khác

```
#include <stdio.h>
#include <conio.h>
void NhapN(int &n);
void NhapMang(int a[], int n);
void XuatMang(int a[], int n);
void main()
{
    int a[100], n;
    NhapN(n);
    NhapMang(a, n);
    printf("Mang vua nhap: ");
    XuatMang(a, n);
    getch();
}
```

4. BÀI TẬP NÂNG CAO

4.1. Nhập giá trị các phần tử của mảng từ bàn phím

Bài 1. Tạo mảng A gồm n phần tử ($n > 0$), với yêu cầu giá trị của các phần tử của mảng thỏa điều kiện đều là số dương:

☞ Mở rộng:

- Các trường hợp số âm, số chẵn, số vừa âm vừa lẻ, ...
- Nằm trong khoảng từ 50 đến 100.
- Nhỏ hơn 10 hoặc lớn hơn hay bằng 50.



Bài 2. Tạo mảng A gồm n phần tử ($n > 0$), với yêu cầu những vị trí (hay chỉ số mảng) là số lẻ chỉ nhận giá trị nhập vào là số lẻ, và những vị trí (hay chỉ số mảng) là số chẵn chỉ nhận giá trị nhập vào là số chẵn.

➤ Mở rộng cho trường hợp ngược lại: vị trí lẻ chỉ nhận số chẵn; vị trí chẵn chỉ nhận số lẻ.

Bài 3. Tạo mảng A gồm n phần tử ($n > 0$), với yêu cầu giá trị của phần tử nhập sau trong mảng phải lớn hơn hoặc bằng phần tử liền trước (sau khi nhập hoàn tất, ta thu được mảng được sắp xếp tăng dần).

VD1: 2 4 7 7 8 11 \Rightarrow Hợp lệ

VD2: 2 4 3 \Rightarrow không hợp lệ \Rightarrow yêu cầu nhập lại

Bài 4. Tạo mảng A gồm n phần tử ($n > 0$), với yêu cầu chỉ cho người dùng nhập giá trị của các phần tử của mảng là số nguyên tố

➤ Mở rộng cho các trường hợp số hoàn thiện, số chính phương, ...

4. 2. Tạo mảng với giá trị của các phần tử được phát sinh ngẫu nhiên

Bài 5. Lần lượt viết các hàm phát sinh ngẫu nhiên mảng 1 chiều các số nguyên dương gồm n phần tử trong từng trường hợp sau (mỗi hàm xử lý cho 1 trường hợp):

- Với $5 < n < 200$, và giá trị của các phần tử trong khoảng từ 0 đến 99.
- Với $5 < n < 50$, và cho giá trị của các phần tử trong khoảng từ -100 đến 100.
- Với $5 < n < 50$, và các số nguyên sao cho giá trị phát sinh ngẫu nhiên phải toàn là số lẻ.

➤ Mở rộng cho các trường hợp số chẵn, số vừa âm vừa lẻ, số nguyên tố, số hoàn thiện, ...

Bài 6. Viết hàm phát sinh ngẫu nhiên mảng 1 chiều các số nguyên gồm n phần tử, sao cho số phát sinh sau phải lớn hơn hay bằng số phát sinh liền trước đó (các phần tử của mảng được sắp xếp tăng dần).

➤ Mở rộng cho trường hợp số giảm dần.

Bài 7. Viết hàm phát sinh ngẫu nhiên mảng 1 chiều các số nguyên gồm n phần tử, với yêu cầu những vị trí (hay chỉ số mảng) là số lẻ chỉ nhận giá trị nhập vào là số lẻ, và những vị trí (hay chỉ số mảng) là số chẵn chỉ nhận giá trị nhập vào là số chẵn.

➤ Mở rộng cho trường hợp ngược lại: vị trí lẻ chỉ nhận số chẵn; vị trí chẵn chỉ nhận số lẻ.

4. 3. Liệt kê theo giá trị phần tử

Bài 8. Liệt kê các phần tử trong mảng có giá trị lẻ.

➤ Mở rộng: giá trị chẵn, giá trị âm, giá trị dương, ...

Bài 9. Liệt kê các phần tử trong mảng có giá trị là số chẵn và nhỏ hơn 20.

➤ Mở rộng: giá trị chẵn và là số âm, giá trị lẻ và là số dương, ...

Bài 10. Viết hàm (ngoài các tham số cần thiết) nhận tham số là số x, và in ra tất cả các số có giá trị lớn hơn x.



➤ Mở rộng: khác X, nhỏ hơn hay bằng X, bội số của X, ước số của X, ...

Bài 11. Liệt kê các giá trị là số nguyên tố có trong mảng.

➤ Mở rộng cho các trường hợp số đối xứng, số may mắn, số chính phương, số hoàn thiện, ...

Bài 12. Liệt kê các số trong mảng có giá trị thuộc $[x,y]$ cho trước (x và y là tham số của hàm)

➤ Mở rộng: liệt kê các số chẵn trong mảng nguyên thuộc $[x,y]$.

4. 4. Liệt kê theo chỉ số (vị trí) của phần tử

Bài 13. Liệt kê các phần tử nằm tại vị trí lẻ.

➤ Mở rộng cho trường hợp: vị trí chẵn, vị trí là số hoàn thiện, ...

Bài 14. Liệt kê các phần tử trong mảng có vị trí là số chẵn và giá trị của phần tử nhỏ hơn 20.

➤ Mở rộng cho trường hợp: vị trí từ p đến q (thay vì từ 0 đến n-1 như thường dùng), vị trí từ 0 đến p hoặc vị trí từ q đến n-1, ...

Bài 15. Liệt kê các phần tử trong mảng có vị trí là số nguyên tố.

➤ Mở rộng cho các trường hợp số đối xứng, số may mắn, số chính phương, số hoàn thiện, ...

Bài 16. Viết hàm (ngoài các tham số cần thiết) nhận tham số là số x, và in ra tất cả các số có vị trí lớn hơn x.

➤ Mở rộng: khác X, nhỏ hơn hay bằng X, bội số của X, ước số của X, ...).

4. 5. Liệt kê dựa trên kết hợp giữa vị trí và giá trị

Bài 17. In các phần tử có giá trị là số nguyên tố nằm tại những vị trí chẵn trong mảng.

Bài 18. Liệt kê tất cả các cặp giá trị (a,b) trong mảng thỏa điều kiện $a \geq b$ và vị trí (chỉ số) chứa số a < vị trí (chỉ số) chứa số b.

Bài 19. Viết hàm nhận tham số vào 1 số K (với $-999 \leq K \leq 999$). In ra cách đọc chữ số tương ứng (sử dụng mảng chứa các chuỗi “không”, “một”, “hai”, ...).

Ví dụ nhập -456 in ra: Am bon nam sau.

➤ Mở rộng: cho trường hợp giá trị của $K \leq \pm 2.000.000.000$.

Bài 20. Viết chương trình nhập vào năm. In ra tên của năm âm lịch tương ứng.

Ví dụ: nhập năm 2018 in ra Mậu Tuất.

Biết rằng:

CAN	Giáp	At	Bính	Đinh	Mậu	Kỷ	Canh	Tân	Nhâm	Quý		
CHI	Tý	Sửu	Dần	Mão	Thìn	Tỵ	Ngọ	Mùi	Thân	Dậu	Tuất	Hợi

4. 6. Các trường hợp liệt kê khác

Bài 21. Mỗi yêu cầu sau đây được viết thành một hàm riêng biệt:

a)- In mảng vừa tạo thành 2 dòng: ♡ Dòng 1: các số lẻ.



✧ Dòng 2: các số chẵn.

b)- In mảng vừa tạo thành 2 dòng: ✧ Dòng 1: các số âm

✧ Dòng 2: các số dương.

c)- Vẽ biểu đồ ngang theo giá trị của các số có trong array.

Ví dụ: array có 3 số 2, 7, 4 in ra:

```

      x x
      x x x x x x x
      x x x x
  
```

d)- Vẽ biểu đồ đứng theo giá trị của các số có trong array.

Ví dụ: array có 3 số 2, 0, 4, 1 sẽ in ra:

```

                                     x
                                     x
      x                             x
      x                             x  x
  
```