

Bài thực hành số 8

Mảng trong lập trình

Ví dụ minh họa:

Viết chương trình cho phép người dùng nhập vào số phần tử n ($n > 0$). Sau đó người dùng nhập lần lượt các số nguyên, chương trình sẽ sắp xếp và in ra các số đã nhập theo thứ tự tăng dần.

```
Nhap vao so phan tu N < N > 0 >
N = -3
Nhap vao so phan tu N < N > 0 >
N = 5
So thu 1: 4
So thu 2: 7
So thu 3: 2
So thu 4: 9
So thu 5: 5

Day da sap thu tu: 2 4 5 7 9
```

Hình 1 – Thực thi chương trình

Gợi ý giải:

Bước 1: Viết mã lệnh cho phép người dùng nhập vào số phần tử n .

Bước 2: Kiểm tra giá trị n . Nếu n không phù hợp, hiển thị cho người dùng nhập lại.

Bước 3: Dùng vòng lặp hiển thị yêu cầu người dùng nhập số và lưu vào mảng.

Bước 4: Viết hàm sắp xếp các phần tử trong mảng theo thứ tự tăng dần.

Bước 5: Gọi hàm sắp xếp và in các phần tử trong mảng.

Đầu tiên, ta sẽ viết câu lệnh cho phép người dùng nhập vào số phần tử n .

```
int n;
printf(" Nhap vao so phan tu N ( N > 0 )\n N = ");
scanf("%d", &n);
```

Hình 2 – Cho phép người dùng nhập vào số phần tử n

Ta thêm vòng lặp để người dùng nhập lại nếu n bé hơn hoặc bằng 0. Chương trình chỉ thực thi tiếp khi $n > 0$.

```
int n;
do {
    printf(" Nhap vao so phan tu N ( N > 0 )\n N = ");
    scanf("%d", &n);
} while(n <= 0);
```

Hình 3 – Thêm vòng lặp

Các số do người dùng nhập vào sẽ được lưu vào mảng. Do đó ta tiến hành khai báo mảng. Ở mỗi lần người dùng nhập số, lưu số đó vào mảng.

```
const int MAXSIZE = 256;
int arr[MAXSIZE];
int i;
for(i=0; i<n; i++) {
    printf("So thu %d: ", i+1);
    scanf("%d", &(arr[i]));
}
```

Hình 4 – Lưu số do người dùng nhập vào mảng

Hiện thực hàm sắp xếp thứ tự các phần tử trong mảng (Bubble sort).

```
void sapthutu(int arr[], int soPhanTu) {
    int i, j, temp;
    for(i=0; i<soPhanTu-1; i++) {
        for(j=i+1; j<soPhanTu; j++) {
            if(arr[i] > arr[j]) {
                // hoan doi gia tri 2 phan tu thu i va j
                temp = arr[i];
                arr[i] = arr[j];
                arr[j] = temp;
            }
        }
    }
}
```

Hình 5 – Hàm sắp thứ tự

Ta gọi hàm sắp thứ tự truyền vào mảng và số phần tử có trong mảng. Sau khi hàm thực thi, các phần tử trong mảng được sắp thứ tự. Do đó ta in mảng đã sắp xếp ra màn hình.

```
sapthutu(arr, n);

printf("\n Day da sap thu tu: ");
for(i=0; i<n; i++) {
    printf("%d ", arr[i]);
}
```

Hình 6 – Gọi hàm, in kết quả ra màn hình

Chương trình hoàn chỉnh:

```

1  #include <stdio.h>
2  void sapthutu(int arr[], int soPhanTu) {
3      int i, j, temp;
4      for(i=0; i<soPhanTu-1; i++) {
5          for(j=i+1; j<soPhanTu; j++) {
6              if(arr[i] > arr[j]) {
7                  // hoan doi gia tri 2 phan tu thu i va j
8                  temp = arr[i];
9                  arr[i] = arr[j];
10                 arr[j] = temp;
11             }
12         }
13     }
14 }
15 void main()
16 {
17     const int MAXSIZE = 256;
18     int n, i;
19     int arr[MAXSIZE];
20     do {
21         printf("Nhap vao so phan tu N ( N > 0 )\nN = ");
22         scanf("%d", &n);
23     } while(n<=0);
24
25     for(i=0; i<n; i++) {
26         printf("So thu %d: ", i+1);
27         scanf("%d", &(arr[i]));
28     }
29     sapthutu(arr, n);
30     printf("\nDay da sap thu tu: ");
31     for(i=0; i<n; i++) {
32         printf("%d ", arr[i]);
33     }
34     _getch();
35 }

```

Hình 7 – Chương trình hoàn chỉnh

Câu 1: Viết chương trình cho phép người dùng nhập vào số phần tử n ($n > 0$). Sau đó người dùng nhập lần lượt các số thực, chương trình sẽ sắp xếp và in ra các số đã nhập theo thứ tự giảm dần.

Câu 2: Viết chương trình cho phép người dùng nhập lần lượt các số vào mảng và 1 số nguyên dương M . Chương trình in ra M số nhỏ nhất (lớn nhất) trong mảng theo thứ tự tăng dần (giảm dần).

Câu 3: Viết chương trình cho phép người dùng nhập số nguyên dương N . Chương trình tạo ra 1 mảng N phần tử tương ứng với N số Fibonacci đầu tiên và in ra mảng này trên màn hình.

Câu 4: Viết chương trình cho phép người dùng nhập vào mảng số nguyên, sau đó xóa các phần tử trùng nhau và in ra mảng mới.

Câu 5: Biết: nếu số nguyên a chia hết cho số nguyên d thì số d được gọi là ước của số a . Số nguyên dương d lớn nhất là ước của hai hay nhiều số nguyên được gọi là ước chung lớn nhất (ƯCLN). Cho mảng nguyên dương a có n phần tử:

a) Viết hàm tìm ƯCLN của n phần tử

b) Viết hàm tìm và in ra màn hình các ƯCLN của k ($0 < k < n$) số nguyên dương thuộc mảng a