BUỔI 11: MẢNG MỘT CHIỀU – FULL HOUSE

I. Định nghĩa:

Mảng là một tập hợp tuần tự các phần tử có **cùng kiểu dữ liệu** và các phần tử được lưu trữ trong một dãy các **ô nhớ liên tục trên bộ nhớ**. Các phần tử của mảng được truy cập bằng cách **sử dụng "chỉ số".** Mảng có kích thước N sẽ có chỉ số từ **0 tới N – 1**.

Ví dụ: mảng có 9 phần tử:

arr	-2153	738	0	963	7283	6	-8	12	3847
Index	0	1	2	3	4	5	6	7	8

- Nếu muốn tìm tới phần tử 6 bạn phải gọi phần tử thứ 5 của mảng hoặc nếu bạn muốn tìm đến phần tử -2153 bạn phải gọi phần tử thứ 0.

II. Khai báo mảng một chiều.

type name[*size*];

size: kích thước của mảng, việc này xác định số lượng phần tử có thể được lưu trữ trong mảng.

type: kiểu dữ liệu của mảng, việc này chỉ định kiểu dữ liệu của các phần tử trong mảng; là số nguyên, số thực, ký tự hay là kiểu dữ liệu nào đó.

Name: tên mảng do người lập trình đặt.

III. Khởi tạo mảng.

type arr[size] = {elements};

Ví dụ 1: int a[5] = {88, 5, -8, 12, 4};

Ví dụ 2:

```
int a[8];
a[0] = 12;
a[55] = -24;
```

IV. Các thao tác với mảng một chiều

```c
#include <stdio.h>

int main(){
    int a[100] = {3, 6, -7, 9, 32, 24};

    for(int i=0;i<=10;++i){
        printf("%d ",a[i]);
    }
}
```

1. Nhập, xuất mảng 1 chiều

a. Viết trong hàm main

```c
int main(){
    int a[100],n;
    scanf("%d",&n);
    for(int i=0;i<n;++i){
        scanf("%d",&a[i]);
    }
    for(int i=0;i<n;++i){
        printf("%d ",a[i]);
    }
}
```

b. Viết hàm rời

```
26 void nhap(int a[],int n){
27     for(int i=0;i<n;++i){
28         scanf("%d",&a[i]);
29     }
30 }
31
32 void xuat(int a[], int n){
33     for(int i=0;i<n;++i){
34         printf("%d ",a[i]);
35     }
36 }
37
38 int main(){
39     int a[100],n;
40     scanf("%d",&n);
41
42     nhap(a,n);
43     xuat(a,n);
44 }
```

V. Các bài tập làm quen với mảng

1. Tính tổng các phần tử trong mảng

```
56  void nhap(int a[],int n){
57      for(int i=0;i<n;++i){
58          scanf("%d",&a[i]);
59      }
60  }
61
62  int tong(int a[],int n){
63      int s=0;
64      for(int i=0;i<n;++i){
65          s += a[i];
66      }
67      return s;
68  }
69
70  int main(){
71      int a[100],n;
72      scanf("%d",&n);
73
74      nhap(a,n);
75      printf("%d",tong(a,n));
76  }
```

2. In ra các phần tử là số nguyên tố trong mảng

```
89  void nhap(int a[],int n){
90      for(int i=0;i<n;++i){
91          scanf("%d",&a[i]);
92      }
93  }
94
95  int snt(int n){
96      for(int i=2;i*i<=n;++i){
97          if(!(n%i)) return 0;
98      }
99      return n>1;
100 }
101
102 void in(int a[],int n){
103     for(int i=0;i<n;++i){
104         if(snt(a[i])) printf("%d ",a[i]);
105     }
106 }
107 int main(){
108     int a[100],n;
109     scanf("%d",&n);
110
111     nhap(a,n);
112     in(a,n);
113 }
```

3. In phần ra các phần tử không vượt quá x (x nhập từ bàn phím)

```
125 void nhap(int a[],int n){
126     for(int i=0;i<n;++i){
127         scanf("%d",&a[i]);
128     }
129 }
130
131 void in(int a[],int n,int x){
132     for(int i=0;i<n;++i){
133         if(a[i]<=x) printf("%d ",a[i]);
134     }
135 }
136 int main(){
137     int a[100],n,x;
138     scanf("%d%d",&n,&x);
139
140     nhap(a,n);
141     in(a,n,x);
142 }
```

4. Tìm phần tử lớn nhất trong mảng

```
147 void nhap(int a[],int n){
148     for(int i=0;i<n;++i){
149         scanf("%d",&a[i]);
150     }
151 }
152
153 int max(int a[],int n){
154     int max = a[0];
155     for(int i=1;i<n;++i){
156         if(a[i]>max) max = a[i];
157     }
158     return max;
159 }
160 int main(){
161     int a[100],n;
162     scanf("%d",&n);
163
164     nhap(a,n);
165     printf("%d",max(a,n));
166 }
```

5. **Đếm số lần xuất hiện của x trong mảng**

```
180 int dem(int a[],int n,int x){
181     int count = 0;
182     for(int i=0;i<n;++i){
183         if(a[i]==x) ++count;
184     }
185     return count;
186 }
187
188 int main(){
189     int a[100],n,x;
190     scanf("%d%d",&n,&x);
191
192     nhap(a,n);
193     printf("%d",dem(a,n,x));
194 }
```