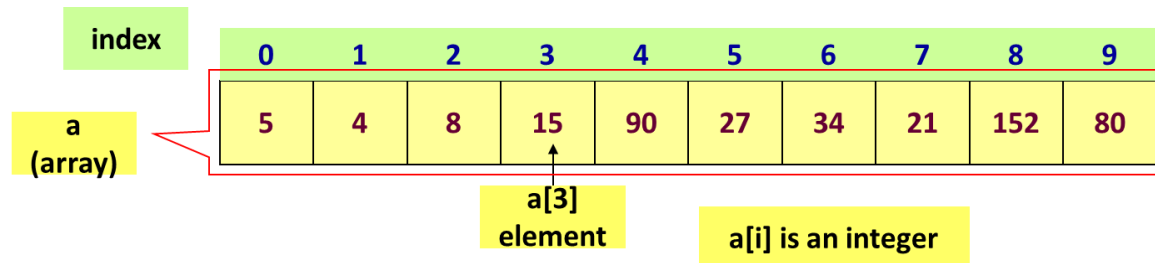


Lab 6: MẢNG MỘT CHIỀU (One-dimensional Arrays)

1-D array: tập các phần tử cùng kiểu dữ liệu và được lưu trữ liên tiếp trong bộ nhớ.



I. Khai báo

```
DataType ArrayName[NumberOfElements];
```

Examples:

```
int a1[5];
```

```
char s[12];
```

```
double a3[100];
```

II. Duyệt mảng

A way to visit each element of an array

Suppose that the 1-D array, named *a*, containing *n* elements.

Forward traversal:

```
int i;
```

```
for (i=0; i<n; i++)
```

```
{ [if (condition)] Access a[i];
}
```

Backward traversal:

```
int i;
```

```
for (i=n-1; i >=0; i--)
```

```
{ [if (condition)] Access a[i];
}
```

III. Mảng một chiều là tham số của hàm

```
void input (int *a, int n)
```

```
void input (int a[], int *pn)
```

```
int sum (int a[], int n)
```

Ví dụ: Viết chương trình:

- Nhập các giá trị cho một mảng nguyên có thể chứa tối đa 100 phần tử
- In ra giá trị lớn nhất
- In ra các giá trị của mảng
- In ra các giá trị chẵn.

```

1 /* Array Parameters Demo.*/
2 #include <stdio.h>
3 #define MAXN 100
4 /* Prototypes */
5 void input(int*a, int n);
6 int max(int a[], int n);
7 void print (int* a, int n);
8 void printEven (int* a, int n);
9 int main()
10 {   int a[MAXN]; /* static array of 100 integers */
11     int n; /* real used number of elements */
12     int maxVal;
13     do
14     {   printf("How many elements which will be used 1..%d:", MAXN);
15         scanf("%d", &n);
16     }
17     while (n<1 || n>MAXN);
18     printf("Enter %d values of the arrays:\n", n);
19     input(a,n);
20     maxVal = max (a,n);
21     printf("Max value:%d\n", maxVal);
22     printf("\nInputted array:");
23     print(a,n);
24     printf("\nEven values in array:");
25     printEven(a,n);
26     while (getchar()!='\n');getchar();
27     return 0;
28 }

```

```

29 void input(int*a, int n)
30 { /* Use forward traversal, accept each value */
31     int i;
32     for (i=0; i<n; i++) scanf("%d", &a[i]);
33 }
34 int max(int a[], int n)
35 { int result = a[0];
36     /* Use forward traversal, compare each value with result */
37     int i;
38     for (i=1; i<n; i++)
39         if (result<a[i]) result=a[i];
40     return result;
41 }
42 void print (int* a, int n)
43 { /* Use forward traversal, print out each value */
44     int i;
45     for (i=0; i<n; i++) printf("%d ", a[i]);
46 }
47 void printEven (int* a, int n)
48 { /* Use forward traversal, print out each value */
49     int i;
50     for (i=0; i<n; i++)
51         if (a[i]%2==0) printf("%d ", a[i]);
52 }

```

IV. Tìm kiếm trên mảng một chiều

```

int firstLinearSearch ( int x, int a[], int n)
{ int i;
  for ( i=0; i<n; i++)
    if ( x == a[i] ) return i;
  return -1;
}

```

```

int lastLinearSearch ( double x, double *a, int n)
{ int i;
  for ( i=n-1; i>=0; i--)
    if ( x == a[i] ) return i;
  return -1;
}

```

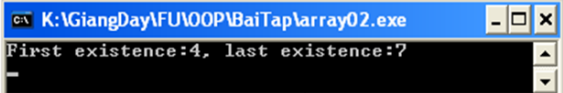
Ví dụ:

```

/* Linear search Demo. */
#include <stdio.h>
int firstLinearSearch ( int x, int a[], int n)
{
    /* Your code */
}
int lastLinearSearch ( int x, int a[], int n)
{
    /* Your code */
}

16 int main()
17 {   int a[] = { 3,34,5,1,2,8,9,2,9 }, x=2;
18     int pos1= firstLinearSearch(x,a,9);
19     if (pos1>=0)
20     {   int pos2= lastLinearSearch(x,a,9);
21         printf("First existence:%d, last existence:%d\n", pos1, pos2);
22     }
23     else printf("%d does not exist!\n", x);
24     getchar();
25     return 0;
26 }

```



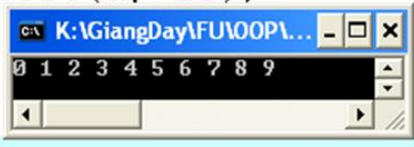
V. Sắp xếp mảng một chiều

```

2 #include <stdio.h>
3 void ascSelectionSort( int* a, int n)
4 {   int minIndex; /* index of min. value in a group */
5     int i,j ; /* vars for looping */
6     /* Group begins at position i to n-1 */
7     for (i=0; i< n-1; i++)
8     {   minIndex = i; /* init minimum position */
9         /* update minIndex of the group at i, i+1,..., n-1 */
10        for (j=i+1; j<n; j++) if (a[minIndex]> a[j]) minIndex= j;
11        /* Move minimum value to the begin of the group */
12        if (minIndex > i)
13        {   int t = a[minIndex];
14            a[minIndex] = a[i];
15            a[i] = t;
16        }
17    }
18 }

```

```
20 void print (int*a, int n)
21 {
22     /* Your code */
23 }
24 int main()
25 {   int a[] = { 1,3,5,7,9,2,4,6,8, 0 };
26     ascSelectionSort(a, 10);
27     print(a,10);
28     getchar();
29     return 0;
30 }
```



VI. Bài tập

Bài 1

Viết chương trình sử dụng mảng 1-D để quản lý một dãy số nguyên (tối đa 100 phần tử), thực hiện các công việc sau:

- Nhập n giá trị.
- In n giá trị của dãy số.
- In các giá trị trong khoảng ($\text{minVal} \leq \text{value} \leq \text{maxVal}$)
- Tìm giá trị nhỏ nhất của dãy.
- Tìm giá trị X trong dãy.
- In dãy theo thứ tự tăng dần.

Bài 2

Viết chương trình sử dụng mảng 1-D để quản lý một dãy số nguyên (tối đa 100 phần tử), dùng MENU đơn giản thực hiện các công việc sau:

- 1- Thêm một giá trị vào cuối mảng
- 2- Tìm kiếm giá trị
- 3- Xóa giá trị
- 5- In mảng
- 6- In mảng theo thứ tự giảm dần
- Others- Thoát