"Heaven's Light is Our Guide"

Department of Computer Science & Engineering

RAJSHAHI UNIVERSITY OF ENGINEERING & TECHNOLOGY

Programming in C

Lab Manual

Lab 4

Array

INDEX

Lab Objectives:

- To know about array that provides convenient structure for representing data
- To study about the array definition and types
- To apply array in C program

Background:

An array (or one dimensional array) is a fixed-size collection of consecutive memory locations. Each memory location in an array is accessed by a relative address called an index.

Basically there are two ways to define one dimensional array:

➢ **datatype Arrayname[Size];**

Example:

❖ int x[100];

❖ char text[80];

➢ **datatype Arrayname[] = { value1, value2,….., valueN };**

Example:

❖ int x[]={1,2,3,4,5};

Values assigned to the individual array elements:

x[0] = 1; x[1] = 2; x[2] = 3; x[3] = 4; x[4] = 5;

❖ char dept[ ] = "C.S.E.";

Values assigned to the individual array elements:

dept [0]='C'; dept [1]='.'; dept [2]='S'; dept [3]='.'; dept [4]='E'; dept[5]='.'; dept[6]='\0';

Two dimensional Arrays are declared as:

➢ **datatype array_name [row_size][column_size];**

Example of initialization:

❖ int table[2][3]= {0,0,0,1,2,3}; OR int table[2][3]= {{0,0,0}, {1,2,3}};

Values assigned to the individual array element:

table[0][0] = 0; table[0][1] = 0; table[0][2] = 0; table[1][0] = 1;

table[1][1] = 2; table[1][2] = 3;

Some Examples:

1. Write a program that will put N numbers in an array and display each element of that array in a new line.

Source code:

```
#include<stdio.h>
int main()
{
        int count, n, number[100];
        printf("How many numbers? : ");
        scanf("%d",&n);
        printf("\n Enter %d numbers:\n\n",n);
        for(count = 0; count < n; count++)
        {
                scanf("%d",&number[count]);
        }

        // Display the element of array number[] :
        printf("\n\n Display: \n\n");
        for(count = 0; count < n; count++)
        {
                printf("The value assigned in number[%d] = %d \n",count, number[count]);
        }
                return 0;
}
```

2. Write a program that will put N numbers in an array and display the sum.

Source code:

```
#include<stdio.h>
int main()
{
        int count, n, number[100], sum=0;
        printf("How many numbers? :  ");
        scanf("%d",&n);
        for(count = 1; count <= n; count++)
        {
                printf("Enter number[%d]:  ",count);
                scanf("%d",&number[count]);
                sum = sum +  number[count];
        }
        printf("\n Sum = %d\n",sum);
        return 0;
}
```

3. Write a program to find the smallest element of an array.

Source code:

```
#include<stdio.h>
int main()
{
        int count, n, number[100], min;
        printf("How many numbers? :  ");
        scanf("%d",&n);
        // Take Input:
        for(count = 0; count < n; count++)
        {
                printf("Enter number[%d]:  ",count);
                scanf("%d",&number[count]);
        }

        // Find the Minimum:
        min = number[0];     // Initially the 1st element of the array is assigned as minimum
        for(count = 1; count < n; count++)
        {
                if(number[count] < min)
                        min = number[count];
        }

        printf("\n The Smallest Element = %d\n",min);
        return 0;
}
```

4. Write a program that will put the even elements and the odd elements of an array into two separate arrays.

Source code:

```
#include<stdio.h>
int main()
{
        int count, n, number[100], even[100], odd[100], countEVEN = 0, countODD = 0;
        /* even[] : Array to store even numbers,
           odd[] : Array to store odd numbers,
           countEVEN = Variable used to count even numbers,
           countODD = Variable used to count odd numbers  */
        printf("Enter number of elements: ");
        scanf("%d",&n);

        for(count = 1; count <= n; count++)
        {
                printf("Enter number[%d]:  ",count);
                scanf("%d",&number[count]);          // Inputs are taken
        if(number[count]%2 == 0){     // If the number is even, then perform the following tasks
                countEVEN++;        // Increment of index of array even to insert a new element
                even[countEVEN] = number[count];
            }
            else{                   // If the number is odd, then perform the following tasks
                countODD++;      // Increment of index of array odd to insert a new element
                odd[countODD] = number[count];
            }
        }

        // Display the Even Numbers:
        printf("\n\n Even Numbers: ");
        for(count = 1; count <= countEVEN; count++)
        {
            printf("\t %d, ",even[count]);
        }
        // Display the Odd Numbers:
        printf("\n\n Odd Numbers: ");
        for(count = 1; count <= countODD; count++)
        {
            printf("\t %d, ",odd[count]);
        }

        return 0;
}
```

5. Write a program that will read and display a matrix.

Source code:

```
#include<stdio.h>
int main(){
        int i, j, row, column, A[20][20];
        printf("How many rows in matrix A: ");
        scanf("%d", &row);
        printf("How many columns in matrix A: ");
        scanf("%d", &column);
        printf("\n Enter the elements of matrix A: \n");
        for(i=1; i<=row; i++){
           for(j=1; j<=column; j++){
                printf("A[%d][%d] = ",i,j);
                      scanf("%d", &A[i][j]);
            }
        }
        printf("\n Matrix A: \n");
        for(i=1; i<=row; i++){
           for(j=1; j<=column; j++){
                      printf("\t %d ",A[i][j]);
                  }
                  printf("\n");
        }
        return 0;
}
```

Exercise:

1. **Write a program that will store N numbers in an array and display the elements of that array in reverse order of index.**

Sample Input:

```
Enter the number of elements: 5
Enter 5 elements:
10
9
8
7
5
```

Sample Output:

```
The elements in reverse order:
Element[5] = 5 ,
Element[4] = 7 ,
Element[3] = 8 ,
Element[2] = 9 ,
Element[1] = 10 ,
```

2. **Write a program that will store the class test marks of N students in an array and find the average class test mark. [**Consider that class test marks can only be integer and display the average class test mark in floating point format**]**

3. **Write a program to find the largest element of an array.**

4. **Write a program that will count the number of odd elements stored in an array.**

5. **Write a program that will search a particular element from an array.**

Sample Input:

```
Enter the number of elements: 5
Enter 5 elements:
10
9
8
7
5
Element going to be searched = 7
```

Sample Output:

```
### 7 is found at position: 4
```

Sample Input:

```
Enter the number of elements: 5
Enter 5 elements:
19
39
18
27
15
Element going to be searched = 5
```

Sample Output:

```
### 5 is not found
```