

LAB 3

Experiment No. 1

TITLE:

To declare and initialize and access one and two dimensional array.

OBJECTIVE:

- To learn how to declare, initialize and access 1 and 2 D array.
- To learn how to use array in C program.

THEORY:

The group of similar type of data in a single variable is known as array. To create array type variable, we have to define data type of array (like int) and its size i.e. how many elements it can store in a single variable. There are two types of array in C programming.

1. One Dimensional Array
2. Two Dimensional Array

- **One Dimensional Array :** A one dimensional array in C contains a series of elements of the same datatype. The elements are zero-indexed, meaning the first element is in position or index 0, and the last element is in position or index (array size) - 1.

syntax

```
data_name array_name[size_of_array];
```

- **Two Dimensional Array :** A two dimensional array in C is a type of array in which the data are organized in rows and column. It is also known as matrix.

syntax

```
data_name array_name[size_of_row][size_of_column];
```

PROGRAM:

1. One Dimensional Array

```
#include <stdio.h>
void main() {
    //declaration and initialization of array
    int a[] = {1,2,3,4,5,6,7,8,9,10};
    int i;

    // Accessing array elements....
    for(i=0;i<10;i++){
        printf("%d ",a[i]);
    }
}
```

```
}
```

Output:

```
1 2 3 4 5 6 7 8 9 10
```

2. One Dimensional Array

```
#include <stdio.h>
```

```
void main() {  
    //declaration and initialization of array  
    int a[][3] = {{1,2,3},{4,5,6},{7,8,9}};  
    int i,j;  
  
    //displaying matrix elements....  
    for(i=0;i<3;i++){  
        for(j=0;j<3;j++){  
            printf(" %d",a[i][j]);  
        }  
        printf("\n");  
    }  
}
```

Output:

```
1 2 3  
4 5 6  
7 8 9
```

RESULTS AND DISCUSSION:

We have declared, initialized and access the one and two dimensional array in the above program. The result of array has been displayed using for loop.

CONCLUSION:

Through this lab, we have successfully demonstrated the implementation of declare, initialize and access one and two dimensional array in C programming. After end of the experiment, we have well knowledge and hands-on experience about how the array works in C programming.

Experiment No. 2

TITLE:

To perform various operations in one dimensional array.

OBJECTIVE:

- To learn how to perform operations in 1 dimensional array.
- To implement the logic using 1 D array.

THEORY:

A one dimensional array in C contains a series of elements of the same datatype. The elements are zero-indexed, meaning the first element is in position or index 0, and the last element is in position or index (array size) - 1.

syntax

```
data_name array_name[size_of_array];
```

PROGRAM:

1. Finding maximum and minimum number between n number of inputs.

```
#include<stdio.h>
```

```
void main(){
    int a[100]; //Array deceleration.....
    int n,i;
    int min, max;

    printf("Enter the size of array: ");
    scanf("%d",&n);

    // initialization array elements.....
    printf("\nEnter %d numbers: \n",n);
    for(i=0;i<n;i++)
        scanf("%d",&a[i]);

    min = a[0];
    max = a[0];

    // Maximum and Minimum value finding operation...
    for(i=0;i<n;i++){
        if(a[i] < min)
            min = a[i];
```

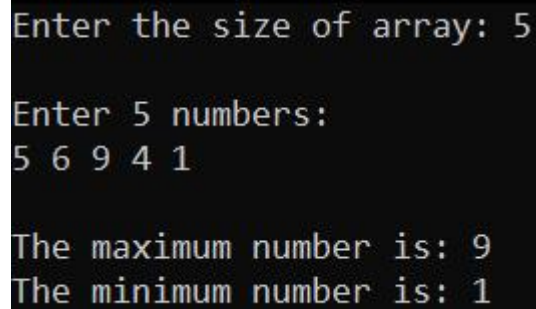
```

        if(a[i] > max)
            max = a[i];
    }

    printf("\nThe maximum number is: %d",max);
    printf("\nThe minimum number is: %d",min);
}

```

Output:



```

Enter the size of array: 5

Enter 5 numbers:
5 6 9 4 1

The maximum number is: 9
The minimum number is: 1

```

2. Finding ascending order of n number of inputs.

```

#include<stdio.h>
void main(){
    int a[100],n; //Array deceleration.....
    int i , j , temp;

    printf("Enter the size of array: ");
    scanf("%d",&n);

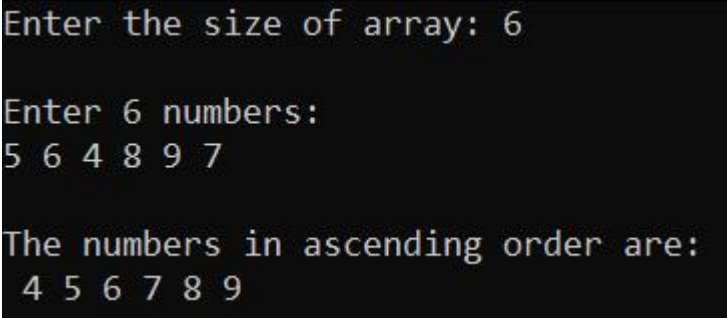
    // initialization of array elements.....
    printf("\nEnter %d numbers: \n",n);
    for(i=0;i<n;i++)
        scanf("%d",&a[i]);

    // Ascending ordering of inputs.....
    for(i=0;i<n-1;i++){
        for(j=i+1;j<n;j++){
            if(a[i] > a[j]){
                temp = a[i];
                a[i] = a[j];
                a[j] = temp;
            }
        }
    }
}

```

```
printf("\nThe numbers in ascending order are:\n");  
for(i=0;i<n;i++)  
    printf(" %d",a[i]);  
  
}
```

Output:

A screenshot of a terminal window with a black background and white text. It shows the execution of a C program. The first prompt is "Enter the size of array: 6". The second prompt is "Enter 6 numbers:", followed by the input "5 6 4 8 9 7". The final output is "The numbers in ascending order are:" followed by "4 5 6 7 8 9".

```
Enter the size of array: 6  
  
Enter 6 numbers:  
5 6 4 8 9 7  
  
The numbers in ascending order are:  
4 5 6 7 8 9
```

RESULTS AND DISCUSSION:

Some operations like finding maximum and minimum, ascending arrangement of n no. of inputs are implemented in the above experiment. User give the size of array and enter the elements of array and the operations will performed. The array elements are displayed with the help of for loop which access the array from 0 position to its last position.

CONCLUSION:

The experiment successfully demonstrated the how to perform operations in one dimensional array in C programming language. After successfully completing and perform the operation in array, we get welled knowledge about how to perform operations in one dimensional array.

Experiment No. 3

TITLE:

To perform various operations in two dimensional array.

OBJECTIVE:

- To learn how to perform operations in 2 dimensional array.
- To implement the logic using 2 D array.

THEORY:

A two dimensional array in C is a type of array in which the data are organized in rows and column. It is also known as matrix.

syntax

```
data_name array_name[size_of_row][size_of_column];
```

PROGRAM:

1. Perform matrix multiplication.

```
#include<stdio.h>
```

```
void main(){
    int a[10][10],b[10][10],c[10][10];
    int i,j,k;
    int ar,ac, br,bc;

    printf("Enter the size of 1st matrix ([A]r*c): ");
    scanf("%d%d",&ar,&ac);

    printf("Enter the size of 2st matrix ([B]r*c): ");
    scanf("%d%d",&br,&bc);

    if(ac == br){
        printf("\nEnter 1st matrix elements ([A]%dx%d):\n",ar,ac);
        for(i=0;i<ar;i++){
            for(j=0;j<ac;j++){
                scanf("%d",&a[i][j]);
            }
        }

        printf("\nEnter 2nd matrix elements ([B]%dx%d):\n",br,bc);
        for(i=0;i<br;i++){
            for(j=0;j<bc;j++){
                scanf("%d",&b[i][j]);
            }
        }
    }
}
```

```

// multiplying 2 matrixes...
for(i=0;i<ar;i++){
    for(j=0;j<bc;j++){
        c[i][j] = 0;
        for(k=0;k<br;k++){
            c[i][j] += a[i][k] * b[k][j];
        }
    }
}

//displaying result.....
printf("\nThe result is A(%dx%d) X B(%dx%d):\n\n",ar,ac,br,bc);
for(i=0;i<ar;i++){
    for(j=0;j<bc;j++){
        printf("\t%d",c[i][j]);
        printf("\n");
    }
}
else{
    printf("\nRow of 1st matrix is must be equal to column of 2nd
matrix.");
}
}

```

Output:

```

Enter the size of 1st matrix ([A]r*c): 3 2
Enter the size of 2st matrix ([B]r*c): 2 3

Enter 1st matrix elements ([A]3x2):
6 5
4 2
9 8

Enter 2nd matrix elements ([B]2x3):
4 6 2
6 3 1

The result is A(3x2) X B(2x3):

      54      51      17
      28      30      10
      84      78      26

```

2. Perform matrix transpose.

```
#include<stdio.h>
void main(){
    int a[10][10],tp[10][10];
    int i , j , row , col ;

    printf("Enter the size of 1st matrix ([A]r*c): ");
    scanf("%d%d",&row,&col);

    printf("\nEnter matrix elements ([A]%dx%d):\n",row,col);
    for(i=0;i<row;i++){
        for(j=0;j<col;j++){
            scanf("%d",&a[i][j]);
        }

        // transposing matrix...
        for(i=0;i<row;i++){
            for(j=0;j<col;j++){
                tp[j][i] = a[i][j];
            }
        }

        //displaying result.....
        printf("\nThe transpose of A(%dx%d):\n\n",row,col);
        for(i=0;i<col;i++){
            for(j=0;j<row;j++){
                printf("\t%d",tp[i][j]);
            }
            printf("\n");
        }
    }
}
```

Output:

```
Enter the size of 1st matrix ([A]r*c): 3 2

Enter matrix elements ([A]3x2):
2 5
3 1
5 6

The transpose of A(3x2):

    2    3    5
    5    1    6
```


RESULTS AND DISCUSSION:

Some operations like matrix multiplication, transpose of matrix; are implemented in the above experiment. User give the size of array and enter the elements of array and the operations will performed.

CONCLUSION:

The experiment successfully demonstrated the how to perform operations in two dimensional array in C programming language. After successfully completing and perform the operation in two array, we get welled knowledge about how to perform operations in one dimensional array.