Programming Fundamentals CT-175

Arrays

Objectives

The objective of this lab is to familiarize students with Array data structure. By the end of this lab students will be able to write programs by using single and multidimensional arrays.

Tools Required

DevC++ IDE

Course Coordinator –
Course Instructor –
Lab Instructor –
Department of Computer Science and Information Technology
NED University of Engineering and Technology

Array

An array is a collection of data items of the same type. Programming problems can be solved efficiently by grouping the data items together in main memory than allocating an individual memory cell for each variable. For example, a program that processes exam scores for a class, would be easier to write if all the scores were stored in one area of memory and were able to be accessed as a group. C allows a programmer to group such related data items together into a single composite data structure called array.

One Dimensional (1D) Arrays

In one-dimensional array, the components are arranged in the form of a list. The syntax for decalring and initialization 1D arrays in C is as follows.

Data_type array_name [size]; /* uninitialized */
Data_type array_name [size] = { initialization list }; /* initialized */
Data_type array_name [ ] = { initialization list }; /* initialized */

- ✓ The general uninitialized array declaration allocates storage space for array array_name consisting of “size” number of items (memory cells).
- ✓ Each memory cell can store one data item whose data type is specified by data_type (i.e., double, int , or char).
- ✓ The individual array elements are referenced by the subscripted variables array_name [0] , array_name [1] , . . , array_name [size −1] .
- ✓ A constant expression of type int is used to specify an array’s size . In the initialized array declaration shown, the size shown in brackets is optional since the array’s size can also be indicated by the length of the initialization list .
- ✓ Element 0 of the array being initialized is set to the first entry in the initialization list , element 1 to the second, and so forth.

All arrays consist of contiguous memory locations. The lowest address corresponds to the first element and the highest address to the last element. For example,
double x[5] = { 5.0, 2.0, 3.0, 1.0, -4.5}; in the memory is as follows,

x[0]	x[1]	x[2]	x[3]	x[4]
5.0	2.0	3.0	1.0	-4.5

```
/* 1D Array Example */
#include<stdio.h>
int main( ){
        int avg, sum = 0;
        int i;
        int marks[5]; // array declaration
        for(i = 0; i<=4; i++){
                printf("Enter student marks: ");
                scanf("%d", &marks[i]); /* stores the data in array*/
        }
        for(i = 0; i<=4; i++)
                sum = sum + marks[i]; /* reading data from array */
        avg = sum/5;
        printf("Average marks are:%d\n", avg);
        printf("total marks are :%d\n", sum);
        return 0;
}
```

Two Dimensional (2D) Arrays

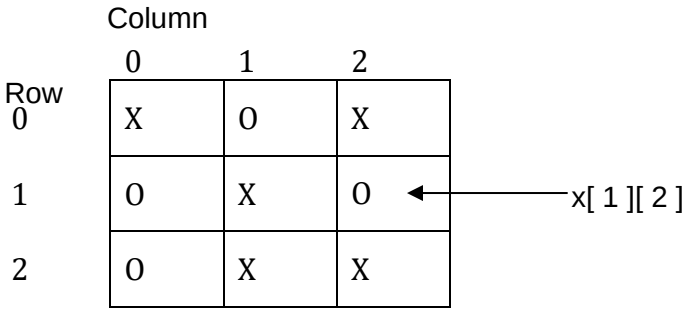
A two dimensional array is a collection of a fixed number of components arranged in rows and columns (that is, in two dimensions), wherein all components are of the same type. Two-dimensional arrays are used to represent tables of data, matrices, and other two-dimensional objects. The syntax of for declaring two dimensional arrays is as follows.

Data_type array_name [size1] [size2]; /* uninitialized */

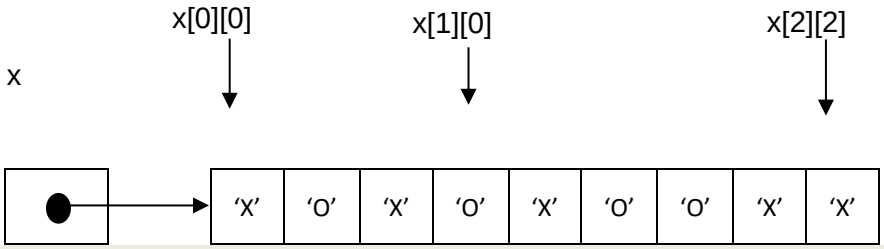
- ✓ Allocates storage for a two-dimensional array (array_name) with size1 number of rows and size2 number of columns.
- ✓ This array has size1*size2 number of elements, each of which must be referenced by specifying a row subscript (0 , 1 ,… size1-1) and a column subscript (0 , 1 ,…size2-1).
- ✓ Each array element contains an element of type data_type.

2D arrays are very useful for storing a table of data (not the only way). Any kind of matrix processing as a 2D array really is a matrix. The two array is represented in the memory as:

char x[3][3] = {{'X', '0', 'X'}, {'0', 'X', '0'}, {'0', 'X', 'X'}};



Because memory is addressed linearly, a better representation is like:




```
//Example Program to display the transpose of given 2x2 matrix(2D array)
#include <stdio.h>
int main( ){
  int matrix[2][2], transpose[2][2], row, col;
  printf("\nEnter elements of matrix:\n");
   for(row=0; row<2; row++)              // Storing elements of the matrix
      for(col=0; col<2; col++){
         printf("Enter element a[%d][%d]: ",row,col);
         scanf("%d", &matrix[row][col]);
      }
   printf("\nEntered Matrix: \n"); // Displaying the matrix[][]
   for(row=0; row<2; row++)
      for(col=0; col<2; col++){
         printf("%d  ", matrix[row][col]);
         if (col == 1)
            printf("\n\n");
       }
    for(row=0; row<2; row++) // Finding the transpose of matrix
      for(col=0; col<2; col++){
         transpose[col][row] = matrix[row][col];
      }
```

```
  printf("\nTranspose of Matrix:\n"); // Displaying the transpose of matrix
  for(row=0; row<2; row++)
    for(col=0; col<2; col++){
      printf("%d  ",transpose[row][col]);
      if(col==1)
        printf("\n\n");
    }
  return 0;
}
```

String as Char Array

String is a sequence of characters that are treated as a single data item and terminated by a null character '\0'. Remember that the C language does not support strings as a data type. A string is actually a one-dimensional array of characters in C language. These are often used to create meaningful and readable programs. Strings can be declared or initialized in C by using char array as follows.

```
// valid
char name[13] = "StudyTonight";
char name[10] = {'c','o','d','e','\0'};

// Illegal
char ch[3] = "hello";
char str[4];
str = "hello";
```

You can use string data type for input and output in C as follows.

```
char str[20];
printf("Enter a string");
scanf("%[^\n]", &str);
printf("%s", str);
```

Gets( ) and Puts( ) string Functions

A string can be read using the %s placeholder in the scanf function. However, it has a limitation that the strings entered cannot contain spaces and tabs. To overcome the problem with scanf, C provides the **gets** function. It allows us to read a line of characters (including spaces and tabs) until the newline character is entered, i. e., the Enter key is pressed. A call to this function takes the following form:

gets(sentence);

where, sentence is an array of char, i.e., a character string. The function reads characters entered from the keyboard until newline is entered and stores them in the argument string sentence, the newline character is read and converted to a null character (\0) before it is stored in s. C also provides another function named puts to print a string on the display. A typical call to this function takes the following form:

puts(sentence);

where, sentence is an array of characters, i.e., a character string. This string is printed on the display followed by a newline character.

Arrays of Strings

One string is an array of characters, an array of strings is a two-dimensional array of characters in which each row is one string. An array of strings can be initialized at declaration in the following manner:

char day[7][10] = {"Sunday" , "Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday"}

Exercise

1. Write a program that reads the numbers from user and store these numbers into an array of same size. Find and display the sum of all positive numbers and calculate the average.

2. Write a C program to read elements in a matrix and check whether matrix is Sparse matrix or not. **Logic:** To check whether a matrix is sparse matrix we only need to check the total number of elements that are equal to zero. The matrix is sparse matrix if T ≥ ((m * n) / 2) where T defines total number of zero elements where m and n are rows and columns respectively.

3. Write down a program which asks user to input his first name and last name in a separate array. After taking the input your program should be able to concatenate first name and last name and return it as full name. Count down number of characters in the full name as well. For example: First name: Muhammad, Second name: Ahmed, Full name: Muhammad Ahmed

4. You taking a square matrix as input from keyboard and then you transpose the same matrix after meeting the requirements you are also interested to find out whether original Matrix A and transpose of Matrix A are equal are not. If the answer is yes, then you print the matrix along with message “matrix is symmetric” otherwise you print the “matrix is asymmetric”.

5. Write a program which takes a matrix of any size as user input and returns the maximum element of matrix as output. Your code should also show the entered matrix on the screen.