

**CL11**  
**8**  
**Programmin**  
**g**  
**Fundamental**  
**s**

**Lab 06**  
**Arrays & Strings**

**NATIONAL UNIVERSITY OF COMPUTER AND EMERGING SCIENCES**

# LAB 06

## Learning Objectives

This lab will cover the following topics:

- Arrays
- 1D Array
- 2D Arrays

### **Arrays:**

An array is a compound data type or a named collection of homogeneous items in which individual items are accessed by their place within the collection. The place within the collection is called an index. All the item in the array must be of the same type, but you can create arrays of integers, arrays of characters and the like.

### **Array Declaration:**

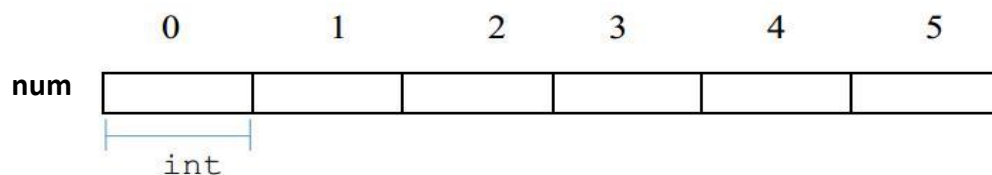
An array declaration specifies the name of the array, the type of elements stored in the array, and the number of array elements (the size of the array).

### **type name [elements];**

For example, six values of type int can be declared as an array without having to declare 6 different variables (each with its own identifier). Instead, using an array, the six int values are stored in contiguous memory locations, and six can be accessed using the same identifier, with the proper index.

Int num[6];

The declaration creates an array of integers named num and reserves enough memory to store six integers.



Where each blank panel represents an element of the array. In C, the first element in an array is always numbered with a zero (not a one), no matter its length. You refer to an individual array element by placing its number (called its subscript or array index) within brackets immediately after the array name. in the num array you denote the first element num[0], the second element num [1] , and so on.

The screenshot shows a C program in a text editor on the left and its execution output in a terminal window on the right. The program, named main.c, includes <stdio.h> and defines a main function. It declares an array num of size 6 and two integer variables j and k. A loop for(j=0; j<6; j++) is used to read six integers from the user using scanf("%d", &num[j]). After the loop, an if(j==5) condition triggers a printf statement that prints "The numbers are : \n\n" followed by a nested loop for(k=0; k<6; k++) that prints each element of the array num[k] separated by a space. The program returns 0. The terminal window shows the user input "1 2 3 4 5 6", the output "The numbers are :", and the array elements "1 2 3 4 5 6". It also shows the process exit message: "Process exited after 6.079 seconds with return value 0" and "Press any key to continue . . .".

```
main.c
1  #include <stdio.h>
2
3  int main() {
4      int num[6];
5      int j,k;
6      printf("Enter the six num");
7      for(j=0; j<6; j++)
8      {
9          scanf("%d", &num[j]);
10         if(j==5)
11             printf("\n The numbers are : \n\n");
12         for(k=0; k<6; k++)
13         {
14             if(j==5)
15                 printf("%d ", num[k]);
16         }
17     }
18
19     return 0;
20 }
```

```
G:\Engr.Kariz\pf fall 2020\Project2.exe
Enter the six num1 2 3 4 5 6

The numbers are :

1 2 3 4 5 6
-----
Process exited after 6.079 seconds with return value 0
Press any key to continue . . .
```

// a program that reads six integers into array and displays them

In the program ,numbers are read by means of a loop in which the loop counter ,j , ranges from 0 to 5 .for each value of j, the statement

**scanf("%d",&num[j]);**

read in numbers and stores it in array element num[j].

## Array Initialization

An array initialization specifies the name of the array, the type of elements stored in the array, and the number of array elements (the size of the array) and the elements also. `int num[6]={ 2, 4, 6, 8, 10};`

## Characters Array or Strings

A string constant is a one-dimensional array of characters terminated by a null ('`\0`'). The terminator defines the end of string.

For example,

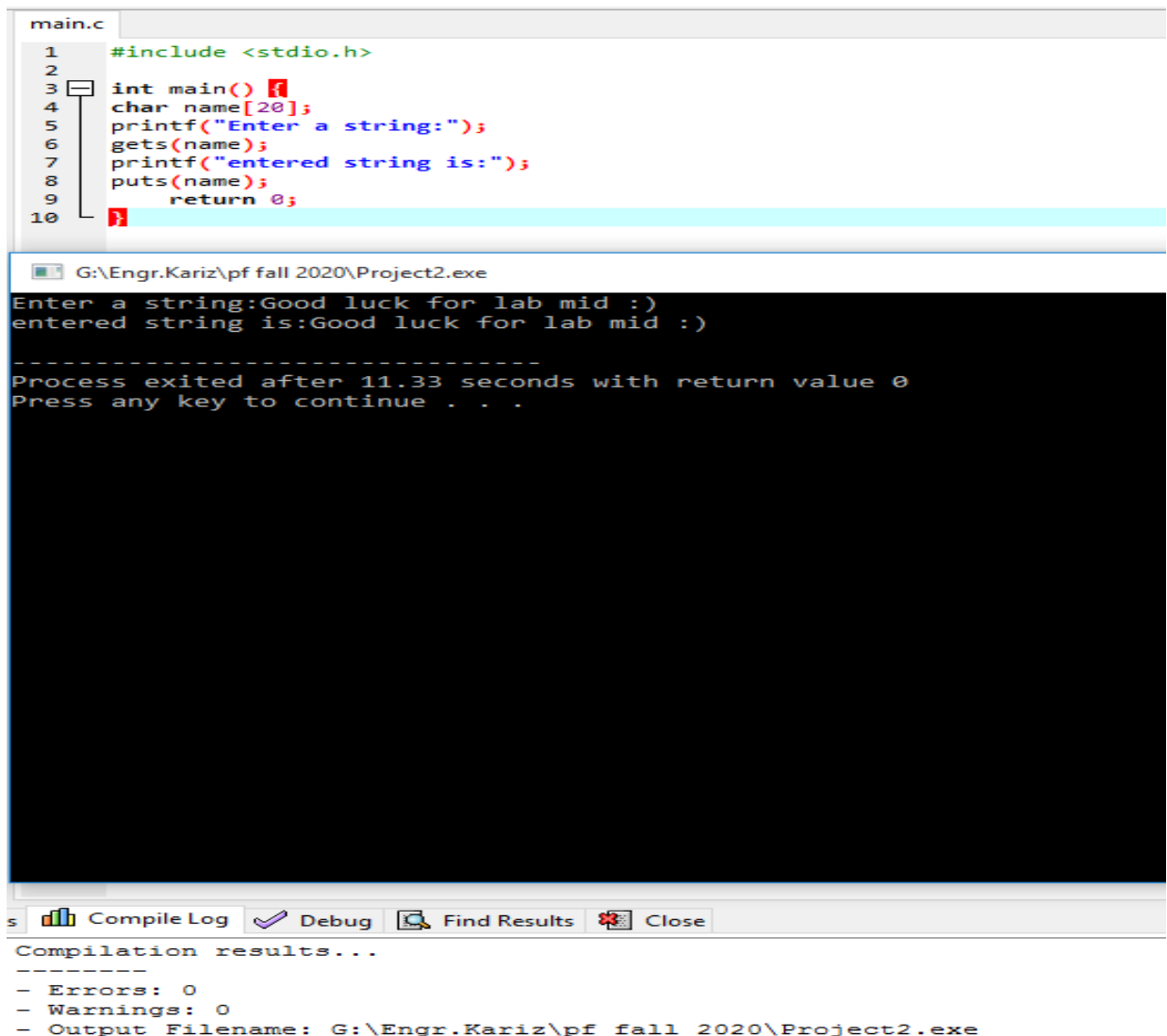
```
char name[5] = { 'F', 'A', 'S', 'T', '\0' };
```

Input function scanf() can be used with %s format specifier to read a string input from the terminal. But there is one problem with scanf() function, it terminates its input on the first white space it encounters. Therefore if you try to read an input string "Hello World" using scanf() function, it will only read Hello and terminate after encountering white spaces.

### The String I/O Function gets() & puts()

scanf() and printf() is not versatile for string I/O we can use gets() and puts() function from stdio library.

For Example:



The image shows a screenshot of a C program being executed. The top window displays the source code for 'main.c', which includes the stdio.h header and uses printf, gets, and puts functions. The bottom window shows the program's output, where the user has entered 'Good luck for lab mid :)' and the program has printed the same string back. Below the output, a status bar indicates the process exited successfully after 11.33 seconds. At the very bottom, a 'Compile Log' window shows that there were no errors or warnings during compilation, and the output file is 'Project2.exe'.

```
main.c
1  #include <stdio.h>
2
3  int main() {
4      char name[20];
5      printf("Enter a string:");
6      gets(name);
7      printf("entered string is:");
8      puts(name);
9      return 0;
10 }
```

```
G:\Engr.Kariz\pf fall 2020\Project2.exe
Enter a string:Good luck for lab mid :)
entered string is:Good luck for lab mid :)

-----
Process exited after 11.33 seconds with return value 0
Press any key to continue . . .
```

Compile Log   Debug   Find Results   Close

```
Compilation results...
-----
- Errors: 0
- Warnings: 0
- Output Filename: G:\Engr.Kariz\pf fall 2020\Project2.exe
```

**C supports a wide range of functions that manipulate null-terminated strings –**

<b>Sr.No.</b>	<b>Function &amp; Purpose</b>
1	<b>strcpy(s1, s2);</b> Copies string s2 into string s1.
2	<b>strcat(s1, s2);</b> Concatenates string s2 onto the end of string s1.
3	<b>strlen(s1);</b> Returns the length of string s1.
4	<b>strcmp(s1, s2);</b> Returns 0 if s1 and s2 are the same; less than 0 if s1<s2; greater than 0 if s1>s2.

Example:

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

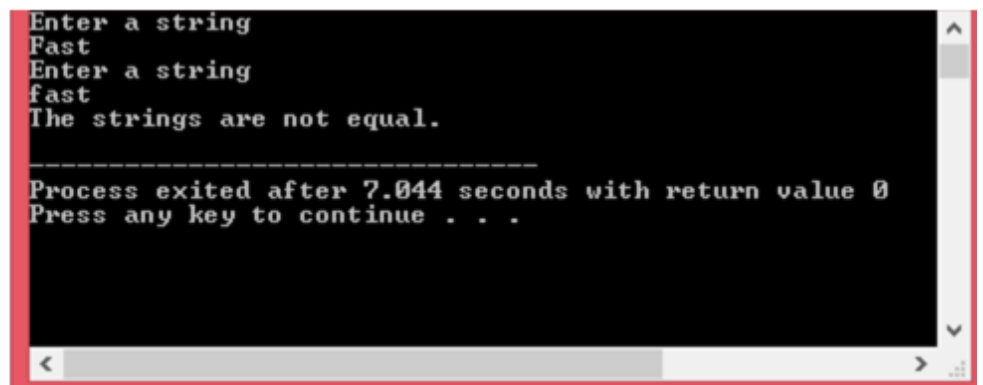
int main()
{
    char str1[100], str2[100];

    printf("Enter a string\n");
    gets(str1);

    printf("Enter a string\n");
    gets(str2);

    if (strcmp(str1, str2) == 0)
        printf("The strings are equal.\n");
    else
        printf("The strings are not equal.\n");

    return 0;
}
```



```
Enter a string
Fast
Enter a string
fast
The strings are not equal.
-----
Process exited after 7.044 seconds with return value 0
Press any key to continue . . .
```

## Two Dimensional Arrays:

In many programming applications you naturally organize data into rows and columns. In C you can use a two-dimensional array to store data in this form. The two dimensional arrays can be describe as “arrays of arrays”. All of them of a same uniform data type.

		0	1	2	3	4
matrix	0					
	1					
	2					

Matrix represents a two-dimensional array of 3 per 5 elements of type int. The C syntax for this is:

```
int matrix [3][5];
```

and, for example, the way to reference the second element vertically and fourth horizontally in an expression would be:

```
matrix[1][3]
```

		0	1	2	3	4
matrix	0					
	1					
	2					

matrix[1][3]

```
//assign row*column integers in the array matrix
```

```
#define ROW 5
```

```
#define COLUMN 3
```

```

int matrix [ROW][COLUMN];
int n,m;

void main ()
{
    for (n=0; n<ROW; n++)
        for (m=0; m<COLUMN; m++)
        {
            matrix[n][m]=(n+1)*(m+1);
        }
}

```

		0	1	2	3	4
matrix	0	1	2	3	4	5
	1	6	7	8	9	10
	2	11	12	13	14	15

Or you can initialize the array as:

```

int matrix [ROW][COLUMN]={1,2,3,4,5
                           6, 7, 8,9,10,
                           11, 12, 13, 14, 15};

```



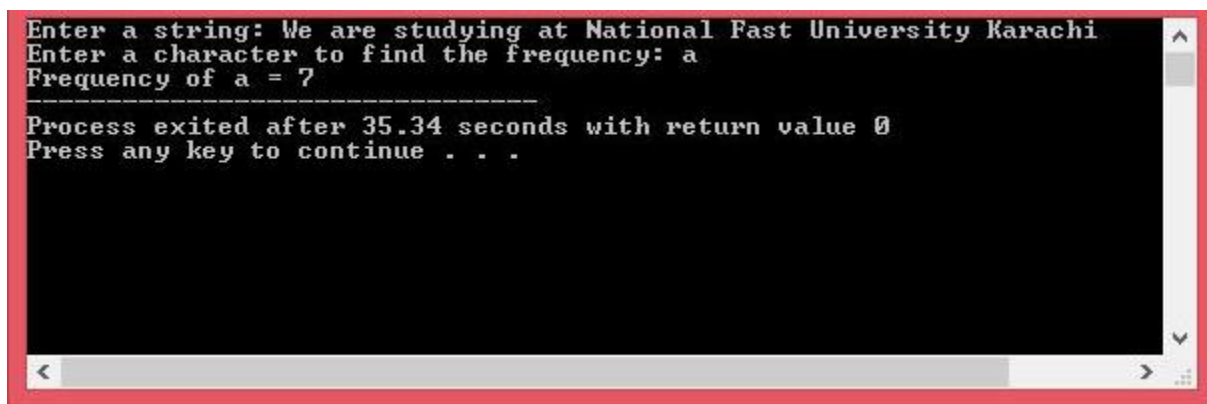
## Lab Activity

### Task# 01

Searching has different techniques when it comes to programming. Finding any certain value from a given range can be done by the help of different algorithms. Generate a random list consisting of 10 numbers and search for a value taken from user. Stop only when user finds specified value. Also mention at which position it was found.

### Task# 02

Write a program in c to show following output

A screenshot of a terminal window with a black background and white text. The text shows the execution of a C program. It starts with the prompt 'Enter a string:' followed by the input 'We are studying at National Past University Karachi'. Then it prompts 'Enter a character to find the frequency:' with the input 'a'. The output shows 'Frequency of a = 7' followed by a horizontal line. Below the line, it says 'Process exited after 35.34 seconds with return value 0' and 'Press any key to continue . . .'. The terminal window has a red border and a scrollbar on the right side.

```
Enter a string: We are studying at National Past University Karachi
Enter a character to find the frequency: a
Frequency of a = 7
-----
Process exited after 35.34 seconds with return value 0
Press any key to continue . . .
```

### Task# 03

Waheed Ali, a student of 2<sup>nd</sup> year studying in university is suffering from fever since last week. Doctor wants to diagnose his temperature values for whole week. Write a program in C that inputs the temperature values of entire week and displays the day on which he had maximum and minimum temperature.

### Task# 04

Write a program in C that inputs the elements of an integer array containing 8 numbers from the user and displays the number of positive, negative and zero values and also find the sum of inputted positive numbers present in an array.

## Task# 05

Yasir Waheed is working in an organization and he gets the different salary in every month. Following table shows his salary each month.

Month	Per Month Salary (in thousands)
January	50K
February	70K
March	55K
April	66K
May	67K
June	72K

He has hired you so that you can make a C program for him in which he can find

- Sum of total salary of all the months.
- Sort the salary in ascending order w.r.t. amount.
- Average of the salary of six months