

LABORATORY SESSION #9

(Program Structure, Arrays)

1. Write a program which asks the user to enter  $N$  integers using the keyboard and store them in a one-dimensional array of type integers. Also, print the square of each array element using a `for` loop. You can take the value of  $N$  from the user ensuring that  $0 < N \leq 100$ .
2. Write a program to reverse array elements (by swapping first element to last, second to second last and soon), without using an additional array.
3. Without running the following program, predict its output. Now run the code and match your output. If it's not matching, track down each of the variables after each statement.

```
#include <stdio.h>
int x = -1, y = -2, z = -3;

void f1 (int x, int z)
{
    y = ++x + --z + (y+=2);
    printf("%d %d %d\n", x, y, z);
}

int main()
{
    int z = 3;
    x += y += ++z;

    printf("%d %d %d\n", x,
            y, z);
    f1(x, z);
    {    /* new block begins */
        float y = 4.0;
        int x = 0;
        x += z = 5 * y;
        printf("%d %.2f %d\n", x,
                y, z);
    }    /* end of block */
    printf("%d %d %d\n", x, y,
            z);
}
```

Notice that a block is a piece of code enclosed within `{` and `}` and introduces its own separate scope. Every function is also a block, and hence has its own scope – called the *function scope*.

4. A string in C is nothing but a one-dimensional character array whose last element is a `'\0'` or null character.
  - a. Write a program which takes a string of characters of size  $\leq 20$  as input from the user and stores it into a 1-dimensional character array. To read the input string entered by the user, you have to use `getchar()` function repeatedly till a new line character (`'\n'`) is found and when a `'\n'` is entered by the user, add a null character as the last element of the array. Also prints the string stored in the array using `for` loop.
  - b. Extend you program to print the number of occurrences of a particular character in the string that you stored in form of the character array. You may require an additional array of size 26 to solve this problem.

#### SAMPLE OUTPUT

Enter the string: BITSPILANI

Number of Occurrences: A:1,B:1,I:3, L:1,N:1,P:1,S:1,T:1

- c. Write a program which takes two strings `str1` and `str2` of length `str1 <=40` and `str2 = <10` as input from the user, store them into two different 1-dimensional character arrays and finds whether `str2` is a substring of `str1`.

#### SAMPLE OUTPUT

Enter the string 1: BITSPILANI

Enter the string 2: PILANI

YES, PILANI is present as the substring in BITSPILANI

5. For the piece of code shown below that finds out the amount of memory allocated (in bytes) for two arrays, and written as part of `main()`, can you predict what gets printed?

```
int arr[] = {10,20,45,67,68};  
printf("Sizeof(arr) = %lu\n", sizeof(arr));
```

Now when the array is passed to a function in which the size is calculated (as shown below), what do you think should get printed?

```
void foo(int arr[], char str[]){  
    printf("Sizeof(arr) = %lu\n", sizeof(arr));  
}
```

- (a) Write a program with these snippets. Call the function appropriately from `main()`, and check whether the outputs you get match what you had predicted.
- (b) What can you say about how the name of the array, `arr`, is interpreted by the compiler according to the context of its use?

#### **Additional Practice Problem**

Your task is to design an evaluation system for online quiz based on multiple choice questions. The program should ask the user to enter the number of questions in the quiz and the correct answer pattern of the questions. 1 (one) mark is given to each correct answer and 0 (zero) for the wrong one. There is no negative marking and 0 marks should be given for un-attempted questions. Each student's answer pattern would be given as input at a time, and evaluated against the "keys" pattern. The output should print total score obtained by the student. After printing the result of the student, the program should ask the user whether he/she wants to continue evaluating another student or stop the program. Make use of arrays to store the keys and the answer pattern of the student entered by the user.

#### Sample Output:

Enter the number of questions (Q): 25

Input key to the quiz:

Keys = abcdabcdabcdabcdabcdabcd

Input the name of the student: Vivek

Input response of the Vivek:

cbcdcdcbabcdabcdcdcdcdcdcd

Marks obtained by Vivek is = 14 out of 25.

Do you want to evaluate another student (Y/N): Y

Input the name of the student: Aakash

Input response of the Aakash:

cbcd dcba abcd abcd d d d d d d d d a

Marks obtained by Akash is = 15 out of 25

Do you want to evaluate another student (Y/N): N

Good Bye!