

BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE, PILANI (RAJ.)
Second Semester 2017-18 CS F111 Computer Programming
LABORATORY SESSION #9
(Pointers; Multi-dimensional Arrays)

1. For the piece of code shown below that finds out the amount of memory allocated (in bytes) for two arrays, can you guess what gets printed?

```
int arr[] = {10,20,45,67,68};  
int mat[][2] = {{10,20}, {30,40}, {50,60}};  
printf("Sizeof(arr) = %lu; sizeof (mat) = %lu\n", sizeof(arr),  
                                             sizeof(mat));
```

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

```
void foo(int arr[], int mat[][2]){  
    printf("Sizeof(arr) = %lu; sizeof (mat) = %lu\n", sizeof(arr),  
                                                    sizeof(mat));  
}
```

A copy of the program is stored at /home/share/9.1.c. Copy it to your working directory, compile and check the outputs. Were they what you expected? Record them in your lab notebook.

2. Copy the program at /home/share/9.2.c into your working directory.

```
1 #include <stdio.h>  
2 int main()  
3 {  
4     int i = 7;  
5     int arr[5] = {i, 2*i, 3*i}, *pa, *pb;  
6     pa = &arr[0];  
7     pb = arr+4;  
8     for (i = -4; i <= 0; ++i)  
9         printf("%d, ",pb[i]); /* negative array index! */  
10    putchar('\n');  
11  
12    printf("pb - pa = %ld\n", pb-pa); /* pointer subtraction */  
13    return 0;  
14 }
```

Several interesting concepts are illustrated in this program. Understand these:

- a. What aspects of array initialization are illustrated (lines 5, 9)? Write in your lab notebook.
 - b. What aspects of pointer arithmetic are illustrated (lines 9, 12)? Write down.
 - c. Try subtracting two pointers of different data types (e.g., `char *` and `int *`). Try adding two pointers of the same type; multiplying; adding a floating point number to a pointer. Write down which of these the compiler permits.
- 3.** ABC bank is trying to establish ATM branches in different regions of a city. There are **m** branches and each branch does **n** transactions per day. The transaction could be a positive value indicating deposit, or negative indicating withdrawal, or zero indicating it was neither a deposit nor a withdrawal. Write a complete C program that accomplishes these tasks:
- a. Reads values for **m** and **n**, and then reads and stores the transaction values in an **m x n** matrix. Data for the program is stored at `/home/share/9.3.txt`. Take a look at the file – the first row has values to be stored in **m** and **n**, and the rest of the matrix follows. (Tip: Use input redirection to read the data from file.)
 - b. Prints the branch number (same as row number) of the branch that had the maximum amount of deposits. Use the following function to achieve this task:

```
int computeMaxDeposit(int row[], int size);
```
 - c. Prints the branch number of the branch that had the maximum transaction value (Hint: Use the `abs()` function from the `<stdlib.h>` library, or implement the equivalent functionality yourself).

```
int computeMaxTransac(int row[], int size);
```

You can view sample output of the program by executing `/home/share/9.3` to see how your program is expected to work.