BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE, PILANI (RAJ.)

Second Semester 2019-20 CS F111 Computer Programming

LABORATORY SESSION 10

(Pointers, Pointers to arrays, Multidimensional arrays and Strings)

1. Write a function circleData() which takes as one of its arguments the radius of a circle, and then computes and makes available in the calling function the area and perimeter of the circle.

2. Write a program to accept an English sentence and an English word from the user, and then to check whether the word exists in the sentence or not. If it occurs, the program should print out how many times the word occurs in the sentence and then remove all the occurrences of the word from the sentence. For taking the input, you have to use input redirection, that is, you have to create a file named input.txt and then take the input from input.txt instead of the keyboard. A sample input.txt could be as shown below

```
 hi hello how are you hello hi
hi
```

input.txt

```
No. of occurrences of hi  is 2
output string is: hello how are you hello
```

3. Write a program to find the transpose of a 2D matrix using two functions *function*$_1$ and *function*$_2$. While *function*$_1$ should accept the original matrix in form of a 2D array as its argument, *function*$_2$ should accept a pointer to the original matrix as the argument.

4. Implement a function convertLower() to convert a given string into one that contains entirely lower case letters. The function performs the conversion on the string received as its argument. First implement your function using ASCII codes only. Next, use the library function tolower() to do the job. Remember to include the header <ctype.h>. You can learn more about this function by reading the manual pages.

[Note: There are a suite of useful library functions like tolower() and toupper() available in the ctype library. Get to know them by typing man isalpha at the shell prompt. You should be familiar with the first 13 functions – starting from isalnum() and ending with isblank(), whose descriptions are also provided in the manual page, so you can use them in your programs when needed.]

5. For the piece of codes shown below find out the output. First note down the answer you expect on your notebook then cross check the answer by executing the code.

a

```
#include<stdio.h>
  int main()
 {
      printf("%d", sizeof(void *));
      return 0;
 }
```

b.

```
#include<stdio.h>
void function(char**);
int main()
{
char *arr[] = { "ant", "bat", "cat", "dog", "egg", "fly" };
function(arr);
return 0;
}
```

```
void function(char **ptr)
{
char *ptr1;
ptr1 = (ptr += sizeof(int))[-2];
printf("%s\n", ptr1);
}
```

LABORATORY SESSION 10 (Practice Problems)

(Pointers, Pointers to arrays, Multidimensional arrays and Strings)

1. ABC bank is trying to establish ATM branches in different regions of a city. There are **m** branches and each branch does a maximum of **n** transactions per day. The transaction could be a positive value indicating deposit, or negative indicating withdrawal, or zero indicating no transaction. Write a complete C program to do these tasks:

> a. Read and store the transaction details of one given day for all the branches in a 2D array. First, the program should read values for **m** and **n**, and then read and store the transaction values in an **m** x **n** array.
>
> b. Find and print out the maximum value of amount withdrawn on that particular day (across all branches). Write a function (whose prototype is given below) that accomplishes this: (COLS and ROWS are preprocessor declared constants).
> int findMaxWithdrawal(int data[][COLS], int m, int n);
>
> **c.** Print the branch number (same as row number) of the branch that recorded the maximum total amount in deposits from the data. Write and use the following function to achieve this task:
> int computeMaxDeposit(int row[], int n, int *max);
> While this function returns back the branch number to the calling function, it also stores the maximum value of the deposits using a pointer)
> [Hint: This function should be repeatedly called to find out the maximum total deposits of a given row (which it takes as the first argument.)]
>
> **d.** Print the branch number of the branch that had recorded maximum transaction value.
> int computeMaxTransac(int row[], int size);
> [Hint: Use the abs() function whose prototype is stored in the <stdlib.h> library, or implement the equivalent functionality yourself.]

2. Write a C program that receives two strings S1 and S2 as input. Then it compares S1 and S2 and then copy the two strings in a new string S3 in alphabetical order. If both the strings are equal then either S1 or S2 is copied into S3.

Few sample I/P and O/P instance is given below:

S1 = Programming S2 = Computer
S3 = Computer Programming

S1 = Computer S2 = Conputer
S3 = Computer

S1 = Computer S2 = Program
S3 = Computer Program

Note: To achieve the task, you are not allowed to any string processing functions.