

SINDH MADRESSATUL ISLAM UNIVERISTY, KARACHI

DEPARTMENT OF SOFTWARE ENGINEERING

FALL 2022

CSC103 - PROGRAMMING FUNDAMENTALS

ZUBAIR-UDDIN SHAIKH

SECTION SE1A/SE1B/SE1C/CS1D^e

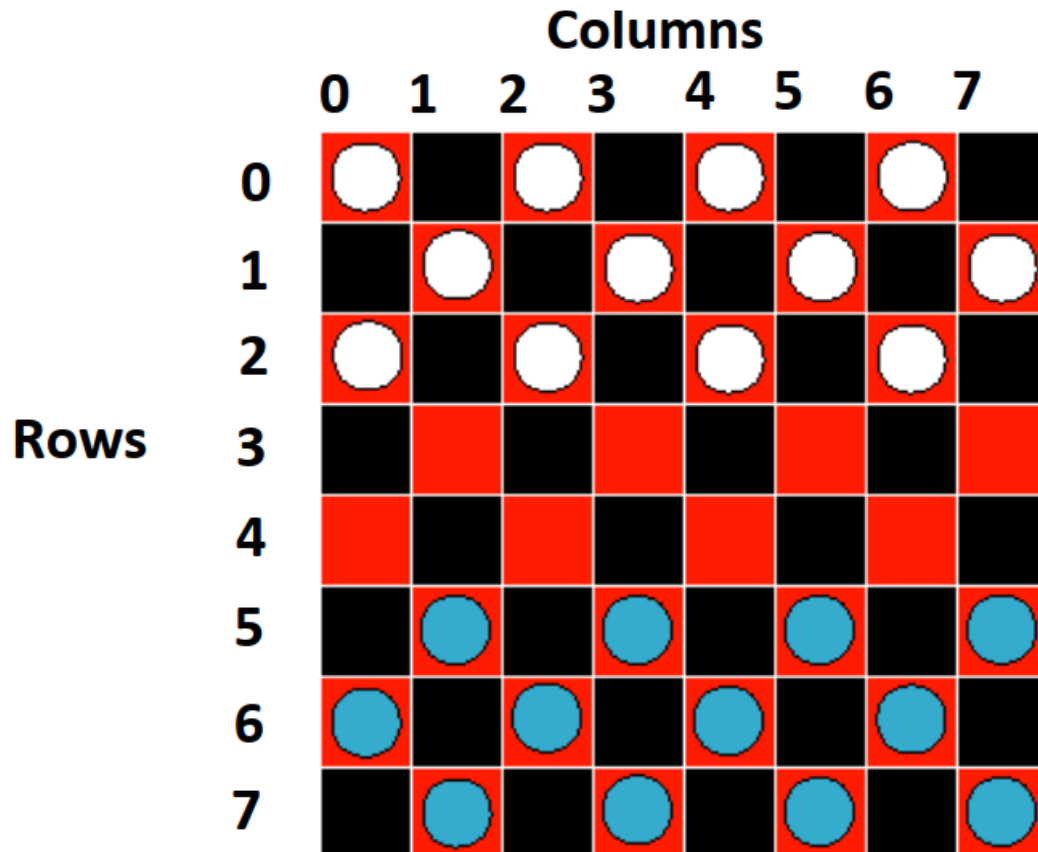
LAB MANUAL 10

2D ARRAYS AND POINTERS

2D ARRAYS AND POINTERS

1. 2-Dimensional Arrays

- Suppose we are writing a program to play the board game, checkers



- We want to store information about what pieces are in which locations.
- The most natural way to store it would be to index locations by the row and column. This is easily done with a 2-dimensional array.

2. Declaring and Accessing 2-D Array:

- `int board[8][8];` //2-D integer array having 8 rows and 8 columns
- `[0 – blank],[1 - white],[2-blue]`
- `board[0][0] = 1;` // a white piece in row 0, column 0
- `board[0][1] = 0;` // row 0, column 1 is empty
- `board[6][0] = 2;` // blue piece in row 6, column 0

Example 01:

```
#include <stdio.h>
int main()
{
    int abc[5][4] = {
        {0, 1, 2, 3},
        {4, 5, 6, 7},
        {8, 9, 10, 11},
        {12, 13, 14, 15},
        {16, 17, 18, 19}
    };
    for (int i=0; i<=3; i++)
    {
        printf("%d ", abc[i]);
    }
    return 0;
}
```

3. Nested Loops and 2-D Array

- Use a nested loop to load the entire array full of blanks:

```
for (i=0; i < 8; i++)
{
    for (j=0; j < 8; j++)
        board[i][j] = 0;
}
```

Example 02: Using nested loops for 2-D Array I/O

```
#include<stdio.h>
int main()
{
    int board[3][3], i,j;
    for(i = 0 ; i < 3 ; i++)
    {
        for (j=0; j < 3; j++)
        {
            printf("Enter value for row %d column %d ", i, j);
            scanf("%d", &board[i][j]);
        }
    }
    for(i = 0 ; i < 3; i++)
    {
        for (j=0; j < 3; j++)
        {
            printf ("%d ", board[i][j]);
        }
        printf("\n");
    }
}
```

Example 03: C code for Determinant of 2X2 matrix

```
#include<stdio.h>
int main()
{
    int a[2][2],i,j;
    long determinant;

    printf("Enter the 4 elements of matrix: ");
    for(i=0;i<2;i++)
        for(j=0;j<2;j++)
            scanf("%d",&a[i][j]);

    printf("\nThe matrix is\n");
    for(i=0;i<2;i++)
    {
        printf("\n");
        for(j=0;j<2;j++)
            printf("%d\t",a[i][j]);
    }

    determinant = a[0][0]*a[1][1] - a[1][0]*a[0][1];

    printf("\nDeterminant of 2X2 matrix: %ld",determinant);
    return 0;
}
```

4. Pointers

- Pointers are used in C programs to access the memory and manipulate the address.
- **Reference Operator(&):** If var is a variable then, &var is the address of that variable in memory.

Example 04:

```
#include <stdio.h>
int main()
{
    int var=5;
    printf("Value: %d\n",var);
    printf("Address: %d",&var); //Notice, the ampersand(&) before var.
    return 0;
}
```

5. Declaring Pointers

- Dereference operator(*) are used for defining pointer variable
- Below statement defines, p as a pointer variable of type int.
- **Syntax:** Type_Name *Variable_Name1, *Variable_Name2,...;
- **Example:** int *p;

Example 05:

```
/* Source code to demonstrate, handling of pointers in C program */
#include <stdio.h>
int main()
{
    int* pc;
    int c;
    c=22;
    printf("Address of c:%d\n",&c);
    printf("Value of c:%d\n\n",c);

    pc=&c;
    printf("Address of pointer pc:%d\n",pc);
    printf("Content of pointer pc:%d\n\n",*pc);

    c=11;
    printf("Address of pointer pc:%d\n",pc);
    printf("Content of pointer pc:%d\n\n",*pc);
    *pc=2;
    printf("Address of c:%d\n",&c);
    printf("Value of c:%d\n\n",c);

    return 0;
}
```

Lab Task 10:

1. Write a C program to check whether a given 2D array is column-magic or not? (every column has the same column sum).
2. Write a program that takes input in two dimensional arrays as matrix and perform its addition and multiplication
 - a. Note: Check the compatibility of both matrix for addition and multiplication
 - b. Your output should be in given format

Sample input:

MatrixA[2][2]={0,1,2,3}

MatrixB[2][2]={0,1,2,3}

Sample Output

Additions:

0 2

4 6

Submission Instructions:

Due Date: Jan 08, 2023

1. For C files, name your C files as questionNumber_yourRollNum_yourSection_LTNumber.c (e.g. Q1_BSE-22F-123_SE1A_LT1.c).
2. Place all files in a folder and name the folder as yourRollNum_yourSection_LTNumber (e.g. BSE-22F-123_SE1A_LT1).
3. Compress the folder by using either Winrar or 7Zip with the same name.
4. Go to tiny.cc/pffall2022smiu and in the “Coordination Document Folder” open the “PF-Activity Submission Form”.
5. Fill out all the details with your correct password and submit the form by the due date.