

Lab Program 12

For this and all future labs, I strongly encourage you to begin your programs with a comment header similar to the following:

```
/*  
Programmer: First & Last Name  
Class: CSCE 1030 Lab  
Date: Today's Date  
Assignment: Lab 12  
cspXX.csci.unt.edu  
*/
```

You are welcome to include more information if you like. This week's assignment we will continue our look at user defined functions, dig deeper into arrays, and introduce pointers. Please avoid “**magic numbers**” when possible and have clean and clear indentions. **I would highly suggest you take advantage of the started code that I have provided at the end of assignment.** Name your file accordingly: **LastnameProg12.c.**

Functions:

1. **GenerateRandomNumbers:** this returns a pseudo-random number that will be used to fill the array. It has been provided.
2. **FillArray:** this function fills a two-dimensional array with valid values. It has also been provided.
3. **PrintArray:** this function will print a two-dimensional integer array to the screen using standard array indexes.
4. **PrintArrayByPointer:** this function will print a two-dimensional integer array to the screen using pointers to index the array.
5. **FindMaxValue:** this function will find the largest value in a two-dimensional integer array. Once the largest value has been discovered, it will dereference the integer pointer that was passed as a parameter, and assign the largest value to it. It will also find the address of the largest value in the array and print it to the screen.
6. **Main:** The main function is mostly provided, please provided the missing code.

The program should:

1. Print the two-dimensional array using indexes.
2. Print the two-dimensional array using pointers.
3. Print the address of the largest value in the array as well as the value.
4. Print the address of variable maxVal as well as its value.
5. Print the address that PtrInt1 is pointing to as well as its dereferenced value. (PtrInt1 should be pointing to the address of variable maxVal)
6. Print the address of arrayInt[0][0] as well as the value it contains.
7. Print the address that Pt2Int2 is pointing to as well as its dereferenced value. (PtrInt2 should be pointing to address of arrayInt[0][0])

Example:

XXXXXXXXXXXXXXXXXXXX

Print Array By Indexes:

30	32	42	69	20
53	58	71	53	59
88	8	96	24	21
17	67	48	70	1
86	45	85	62	2

Print Array By Pointers:

30	32	42	69	20
53	58	71	53	59
88	8	96	24	21
17	67	48	70	1
86	45	85	62	2

Address of highest value in array: 0x7fff1f7aa550 dereferenced: 96

maxValue variable address: 0x7fff1f7aa59c and value 96

PtrInt1 address: 0x7fff1f7aa59c dereferenced: 96

arrayInt[0][0] address: 0x7fff1f7aa520 value at [0][0]: 30

PtrInt2 address: 0x7fff1f7aa520 dereferenced: 30

XXXXXXXXXXXXXXXXXXXX

Sample Code

```
/* Programmer Name:
   Class: CSCE 1030 Lab
   Date: Today's Date
   Assignment: Lab 12
   cspXX.csci.unt.edu
*/

#include<stdio.h>
#include<stdlib.h>
#include<time.h>
#include<unistd.h>

#define ROWS 5
#define COLS 5
#define SEED 100
#define CROSSBAR printf("XXXXXXXXXXXXXXXXXXXX\n");

int GenerateRandomNumber()
{
    int n;
    time_t seconds;
    seconds = time(NULL);
    sleep(1);
    srand((unsigned int) seconds);
    // printf("Seconds: %ld\n", seconds);
    n=random() % SEED; /* n is psydo-random number in range of 0 - SEED */
    return(n);
}

void FillArray(int multidimensional_Array[][COLS])
{
    int i, j;

    for(i = 0; i < ROWS; i++){
        for(j = 0; j < COLS; j++){
            multidimensional_Array[i][j] = GenerateRandomNumber();
        }
    }

    return;
}

void PrintArray(int multidimensional_Array[][COLS])
{
    int i, j;

    printf("\nPrint Array By Indexes: \n");

    return;
}
```

```

void PrintArrayByPointer(int *ptrToArray)
{
    int i, j;

    printf("\nPrint Array By Pointers: \n");

    return;
}

void FindMaxValue(int multidimensional_Array[][COLS], int *max)
{
    int i, j;
    int *PtrToMax = NULL;

    //This is print statement is not correct, fix it.
    printf("Address of highest value in array: %p dereferenced: %d\n\n",
PtrToMax, PtrToMax);

    return;
}

int main(void) {

    int arrayInt[ROWS][COLS];
    int maxValue;
    int *PtrInt1 = NULL; //Set to correct value
    int *PtrInt2 = NULL; //Set to correct value

    FillArray(); //Missing parameter value
    PrintArray(); //Missing parameter value
    PrintArrayByPointer(); //Missing parameter value

    FindMaxValue(); //Missing parameter values

    //Remember that %p is looking for an address

    //Missing parameter values
    printf("maxValue variable address: %p and value      %d\n", , );

    //Missing parameter values
    printf("PtrInt1 address:                %p dereferenced: %d\n\n", , );

    //Missing parameter values
    printf("arrayInt[0][0] address: %p value at [0][0]:%d\n", , );

    //Missing parameter values
    printf("PtrInt2 address:                %p dereferenced:      %d\n", , );

    return 0;
}

```