Lab 8: 2-Dimensional Arrays

2-dimensional arrays are very common in computer programming. They are often used to store tables of data and (as in this lab) to represent data that will be displayed on the screen. Two (or more) dimensional arrays are declared in a manner similar to 1-dimensional arrays: you must give the type of the data, the name of the array and the size of each dimension of the array. For example, a 10 x 20 array of doubles would be defined as:

```c
double sampleArray[10][20];
```

The size of the array must be declared using a size that the compiler can determine, either a value, a constant variable, or a ‘defined’ value. Thus, either of the following would be correct:

```c
const int maxx = 10;
const int maxy = 20;

double sampleArray[maxx][maxy];
```

or

```c
#define maxx 10
#define maxy 20

double sampleArray[maxx][maxy];
```

It’s better programming technique to use a constant variable or a defined value rather than a value (like 10 or 20). With a constant variable or defined value if you want to change the size of the array in the program there is only one place that the program needs to be changed. It also means that you don’t have to remember arbitrary numbers (the array sizes) instead you can remember more meaningful variable names.

To access or to set the value of an element in an N dimensional array you have to give all N index values using the standard square bracket notation:

```c
x = sampleArray[3][4];
```

or

```c
sampleArray[3][4] = 3.1415;
```

As with any array the index values can be integer values or integer variables or expressions that generate integer values. And, as always with arrays, it is the programmer’s responsibility to make sure that the array bounds are not exceeded. E.g., don’t try to access element `sampleArray[12][25]` in an array that was defined to be 10 x 20.
Exercise 8.1

Below is a small program that roughly represents a robot in a room. The room is defined in a 2 dimensional array of characters. Objects in the room are stored as characters in the array. For example, the location of the robot is stored as an 'R' in the array. The walls of the room are stored in the array as a - for the top and bottom walls and as | for the side walls.

The function named setup() puts all of the initial characters into the 2 dimensional array.

For this lab you need to use the curses library (see curses handout) to write a function that prints the room. The function should take the room array as an argument (look at the syntax of the setup function to help with passing the array). The function should loop through the entire array, moving to the correct location of the screen and printing the character stored in the array. E.g., the function should 'move' to location (r,c) and print the [r][c]'th element of the array for each r and c within the dimensions of the room.

```cpp
#include <curses.h> // include the curses library

using namespace std;

const int MAXX = 40; // maximum size of the 'room'
const int MAXY = 20;

void setup( char[][MAXY], int x, int y ); // room setup

int main()
{
    WINDOW *wnd;
    char room[MAXX][MAXY]; // stores the room
    int x = MAXX/2, y = MAXY/2; // robot's initial location

    wnd = initscr(); // 'initializes' the window
    clear(); // clears the window
    refresh(); // reprints the window

    setup(room,x,y); // setup the room with the robot

    print(room); // write this function!!

    endwin(); // frees the screen for normal use
}
```
/* Sets up the room, adding walls, and the robot */
void setup( char r[][MAXY], int x, int y )
{
    for(int i = 0; i < MAXX; i++)
    {
        for(int j = 0; j < MAXY; j++)
        {
            r[i][j] = ' '; // empty the room
        }
    }
    for(int i = 0; i < MAXY; i++)
    {
        r[0][i] = '|'; // left wall
        r[MAXX-1][i] = '|'; // right wall
    }
    for(int i = 0; i < MAXX; i++)
    {
        r[i][0] = '-'; // top wall
        r[i][MAXY-1] = '-'; // bottom wall
    }
    r[x][y] = 'R'; // place the robot
}

• Turn in your program and sample output showing the room drawn on the screen.