Mir Hossain

CS302 HW1

Problems 1,2,4

This file is going to have problems exercise 1,2 and 4. Exercise 3 is going to be on a separate file for the actual program itself. Exercise 1 and 4 were done on a text editor and a compiler so that's why my explanations are in comment format.

Ex1:

```
/*
Mir Hossain
First problem from the first homework assignment
CS302
*/

#include <iostream>
#include <stdlib.h>
using namespace std;

void twist(int a, int& b){

  int c;

  c = a+3; //c = 3 + 3 -> c now holds the value of 6
  a = a*3+c; //a = 3*3 + 6 -> a now holds the value of 15
  b = c+a; //b = 15 + 6 -> b now holds the value of 15
}

int main(){

  int r = 2;
  int s = 3; //s corresponds with a
  int t = 3; //t corresponds with b
  twist(t, s);

  cout << r << " " << s << " " << t << endl;
```

```
  return 0;
}
/*
```

The output of this program is 2 21 3

The reason why the value of S changed is because b is passed by as reference so the function twist only affects the value of s. If a was passed by as reference then the output would 2 21 15. The function would only change the values of any variables if they were passed as reference.

```
*/
```

Exercise 2: On next page, I did it on a separate piece of paper.

Mr Hossain
CS - 302
HW 1

Exercise 2:

A.) The number of rows and columns must be specified in the function prototype because the numbers of rows and cols are compile time constants. The compiler needs to know how much memory the variable needs to occupy for the program to compile & function.

B.)
```
#include <iostream>

int row = 3;
int col = 5;

int **array = new int*[row]
    for(int i = 0; i < row; ++i)
    array[i] = new int[col];
```

Visual Drawing →

C.)

arr[0] →	arr[0][0]	arr[0][1]	arr[0][2]	arr[0][3]	arr[0][4]
arr[1] →	arr[1][0]	arr[1][1]	arr[1][2]	arr[1][3]	arr[1][4]
arr[2] →	arr[2][0]	arr[2][1]	arr[2][2]	arr[2][3]	arr[2][4]

D.) Since the above is an array of pointers, then the program can just find a specific location just by calling the location ex: find spot [2][1] -> std::cout<<arr[2] or use traversal methods to find location. To find address, you would just add & to the cout statement

EX4:

```
#include <iostream>
using namespace std; /*Error #1: cout is used multiple times so including namespace
std was needed, or I could have just used std::cout
*/

int main()
{
  int* ptr;
  int* temp;
  int x;

  ptr = new int;
  temp = new int; //dynamically allocated temp(for any possible memory issues)
  *ptr = 4;
  temp = ptr; //Got rid of * on both ptr and temp since it would cause type comparison issues
  cout << *ptr << *temp; //had to reference variables because they were pointers
  //output: 44

  x = 9;
  *temp = x; //Did not change anything
  cout << *ptr << *temp;
  //New output: 4499

  ptr = new int;
  *ptr = 5; //Had to add * to ptr since its original type was a pointer
  cout << *ptr << *temp;
```

```
  //Output is: 449959
  return 0;
}
```

For this program, there were issues with dereference and memory allocation throughout the program. I commented out what I fixed throughout the program and noted where I added the *. I included reasons for editing the program. At the beginning of the program I allocated temp so there would be no memory leaks in the program.