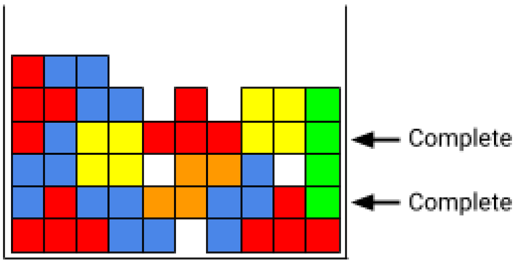
Computer Programming 1

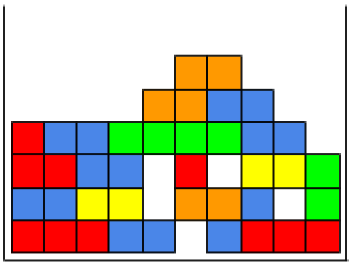
Midterm Exam - October 30th 2018

- You have **2 hours** to complete the assignment.
- **If the code does not compile, the exercise won't be accepted for submission.**
- Code is expected to be readable, clean, and optimal.
- A skeleton of the exercise is provided. **Use it! Don't create a new solution.**
- Inside the code, **replace "TYPE YOUR NAME HERE" with your complete name.**
- When you finish, **ZIP the whole folder** with a filename called **"lastname_name.zip"** and upload it to the **"Midterm Exam"** folder.

1. **(3.5 points)** You are programming the game Tetris. In the file *exercise1.cpp*, implement the body of the function *countCompleteLines* that, given a two-dimensional array representing the board, returns the number of horizontal lines (rows in the array) that contain no empty cells. For example:



In this case, the value returned by the function *countCompleteLines* should be 2.



In this case, the value returned should be 0, as there are no complete lines.

NOTE: If an element in the array equals 1, there is a piece in that cell. Otherwise, if it equals 0, the cell is empty.

2. **(3.5 points)** In the file *exercise2.cpp*, implement the function named *isPalindrome*. The function receives a single parameter: an array of characters. It must return an integer value: 1 if the word is a palindrome, or 0 otherwise.

NOTE: a word is a palindrome if it is read the same from left to right and from right to left.

Examples:

input array: **“reconocer”**	returned value: **1**
input array: **“palabra”**	returned value: **0**
input array: **“abba”**	returned value: **1**

TIP: First compute the length of the string to know which is the index of the last letter (beforehand you don’t know its length, but you know that **the last character found in the string will be '\0'**).

Example:

0	1	2	3	4
‘a’	‘b’	‘b’	‘a’	‘\0’

In this example, **the length of the string “abba” is 4**. Therefore, the last letter **‘a’** is at index **3**. The special character **'\0'** is at index **4**.

3. **(3 points)** Open the file *exercise3.cpp* and implement the function *sumDivisiblesBy5InRange* that, given two integers *begin* and *end*, computes and returns the addition of all numbers in the range [begin, end] (so both numbers included) which are divisible by 5.

   Examples:

   A call like ***sumDivisiblesBy5InRange(1, 2)*** should return **0** because there are no numbers divisibles by 5 between 1 and 2.

   A call like ***sumDivisiblesBy5InRange(3, 23)*** should compute and return the sum of **5 + 10 + 15 + 20**.

   A call like ***sumDivisiblesBy5InRange(-5, 15)*** should compute and return the sum of **-5 + 0 + 5 + 10 + 15**.