



The University of the West Indies, St. Augustine
COMP 1602 Computer Programming II
Summer, 2020/2021

Tutorial 2

1. Write a function, `compare`, that accepts two 1D arrays `list1` and `list2`, along with a size `n`. The function should return the following values:
 - **-1**: if all the elements of `list1` are smaller than the corresponding elements of `list2`.
 - **0**: if all the elements of `list1` are the same as the corresponding elements of `list2`.
 - **1**: if all the elements of `list1` are greater than the corresponding elements of `list2`.
 - **100**: otherwise
2. Write a function, `isValidLocation`, which when given a particular location (row, col), determines if the location is valid for the given 2D board. A location is valid if it is within the range of the rows and columns of the initialized 2D board. The function accepts the row-column combination to check for.
3. In searching for the neighbours (adjacent elements) of an element in a 2D array, it is important to make sure that the neighbours are within the boundary of the 2D board. Write a function `printValidNeighbours` which when given a integer 2D board, and a row and a column, determines the adjacent elements to the given cell and prints the locations of the valid ones.
4. A 2D board is crooked if the number of cells above the diagonal contain more ones than the cells below the diagonal. The number of rows in the 2D array must be even and equal to the number of columns. Write code to implement the crooked function. The prototype for the crooked function is given below:

```
bool crooked(int cells[][MAX_COLS], int size)
```

5. You are given the following struct declaration for BoardElement

```
struct BoardElement{  
    char data;  
    bool isBlack;  
};
```

- (a) Write code to declare a 2D array of BoardElements, `board`, with size 6×6.
- (b) Write code to initialize all elements in `board` to a 'c', with each element being black.
- (c) Write code to print all of the elements in the board.
- (d) Write code to search row 0 of the board for a given character `key`.