This assignment can be completed in two weeks. It will be of great benefit for you to complete your Lab 6 early and get started on the final project.

For this assignment, we will use the topics we have learned throughout the semester to program a simple “video game.” We will call the game “Line Explorer.” Your character travels left and right along a line of letters, choosing at each move whether to pick up the letter at her current location. Once you have picked up the letter, the location no longer holds a letter.

Your goal is to pick up 3 letters in a row that are consecutive in the alphabet. For example: if you pick up C, then E, then D, you win.

The game begins with:
- Your character at the center of an 11-location line
- Every location of the line assigned a letter, from A to G
  - Each letter should be randomly chosen from A to G, but you will get partial credit if you skip the randomization step (e.g., just say `line[3] = 'A'`)

In each turn, you are asked:
- What direction do you wish to move (left or right)? (You then move one step in the specified direction)
- Do you wish to pick up the letter at your location (yes or no)?

In each turn, the game displays:
- The list of the last three letters you have picked up
  - All previously picked-up letters are ignored
  - The display order of the listed letters does not matter to me!
- Your new location
- If you have won (three consecutive letters) or lost (out of turns)

The program must:
- maintain the letters remaining in the line
- maintain the list of the last three letters picked up by the player
- maintain the location of the player after each turn
- count the number of turns
- repeatedly check if the last three picked-up letters are consecutive in the alphabet
In terms of concepts from class, your program must

- Use at least one new function
- Use a loop
- Use at least one array

**Functions:**
I advise (but do not require) you write:

- a function to display the current line
- a function to test if you currently have collected three of the same letter

**Arrays:**
I advise (but do not require) you define:

- An array listing the last three letters that have been collected
- An array specifying the letters at each location on the line

**Submitting your file:**
Submit the final C++ code as lineTraveler.cpp using submit1600 (and verify proper submission using verify1600).

Please see the grading guidelines starting on the next page as you allot your time for different components of this assignment.

Below is an example execution of the game:

>`./lineExplorer`
Welcome to Line Explorer:

EGBDBCDBAFC
  X

Turn 1:
Your current letters: []

What direction do you wish to move ([L]eft or [R]ight)? R
EGBDBCDBAFC
  X
Do you wish to pick up the letter at your location ([Y]es or [N]o)? Y

Turn 2:
Your current letters: [D]
What direction do you wish to move ([L]eft or [R]ight)? R
EGBDBC-BAFC
  X
Do you wish to pick up the letter at your location ([Y]es or [N]o)? Y

Turn 3:
Your current letters: [BD]
What direction do you wish to move ([L]eft or [R]ight)? R
EGBDBC--AFC
  X
Do you wish to pick up the letter at your location ([Y]es or [N]o)? Y

Turn 4:
Your current letters: [ABD]
What direction do you wish to move ([L]eft or [R]ight)? R
EGBDBC---FC
  X
Do you wish to pick up the letter at your location ([Y]es or [N]o)? N

Turn 5:
Your current letters: [ABD]
What direction do you wish to move ([L]eft or [R]ight)? R
EGBDBC---FC
  X
Do you wish to pick up the letter at your location ([Y]es or [N]o)? Y

You won! Letters: [CAB]

Grading guidelines:
For your planning purposes, I am providing here the number of points you will need to be placed in the “A range”, “B range”, “C range” or “D range” for the Final Project. I also provide a breakdown of the number of points I will award for each component of the Final Project Assignment.

“A range” – 84-100 points
“B range” – 66-84 points
“C range” – 50-66 points
“D range” – 34-50 points
Points awarded:
As usual, 30 points for style and 70 points for implementation.

**Style**
Comments – **10 points**
Variable names – **10 points**
Spacing – **10 points**

**Implementation**
Declare variable to hold the 11-location line of letters, and initialize each position with a random letter between A and G – **10 points**
Print letter line and player location – **10 points**
Print last three player-selected letters – **10 points**
Remove letter from line and place it on player’s letter list – **15 points**
Looping for six turns, unless user wins earlier – **5 points**
Updating player location – **5 points**
Test if three most recently selected letters are consecutive in the alphabet – **15 points**
  •  Note: You will get 5 points if you instead successfully test if the three most recently selected letters are the same letter.