

Assignment 5

1-D Arrays and C-Strings

Due: Sunday, March 5, 2017 11:59pm

(90 pts) **Problem Statement: Go Bowling**

You have just been tasked with writing the software for your local bowling alley. You must simulate bowling with random numbers and make sure your software can keep score for different players in a game.

If you are not familiar with bowling, then you can read these instructions on how to play the game and keep score: <http://bowling.about.com/od/rulesofthegame/a/bowlingscoring.htm>

Our game can have 1-5 players, and it allows users to play again with a different number of players at the end of a game. The game begins by asking for the number of players and prompting each player for their name (**storing it in a C-style string/character array, not a C++ string/object!**). Then, the players can begin to bowl.

There are 10 total frames in a game, and for each frame, a player is given up to 2 bowls (chances) to knock down 10 total pins, except in the last frame when a player might get a 3rd chance/bowl. With the first bowl of any frame, a player can knock down 0-10 pins, and with the second bowl, a player can knock down 0 to those left remaining from the first bowl. If the player knocks down all 10 pins in their first bowl of a frame, then that is called a **strike** (denoted with an X), and the player doesn't get a second bowl/chance to knock down any remaining pins. If a player knocks down 0 pins, this is called a **gutter** ball and denoted with a dash, -, and if a player knocks down all 10 pins in a frame using two bowls, then it is called a **spare** (denoted with a forward slash, /). For each player, you must keep an array of pins knocked down with each bowl for each frame, and display this information after each bowl.

The score for a frame is determined by the total number of pins knocked down, but if the player makes a strike or spare, then the scoring is based on subsequent bowls. For a strike, the player receives a 10 plus the number of pins knocked down with the next two bowls. This can give a player 10-30 points for a frame with a strike! For a spare, the player receives a 10 plus the number of pins knocked down with the next bowl. This can give a player 10-20 points for a frame with a spare! Since a player might strike or spare in the 10th frame, then there is a 3rd bowl provided for a strike thrown with the first bowl or a spare thrown with both bowls of the 10th frame. For each player, you must keep an array with a sum of current and prior frame scores that is displayed with each frame.

A player's total score is the sum of all 10 frame scores. You must keep an array of total scores for all players that is updated and displayed after each bowl or frame. The player who has the most points at the end of 10 frames is the winner!

A few game specifics:

- The game scoresheet consists of player names with frame and total information. **The frame information consists of the number of pins knocked down with each bowl in each frame and the sum of current and prior frame scores for each frame.**
- After each bowl, you must print the scoresheet for the users displaying information about the number of pins knocked down and possibly updated total score.
- After each frame that isn't a strike or spare, you must print the scoresheet for the users displaying information with total frame scores and a total score, if they can be calculated at that time.
- There are 10 total frame scores and possibly 21 bowls for each player.
- **Bowling:**
 - A player gets two bowls/chances to knock down 10 pins in each of the 10 frames.

- **Scoring:**

- Only use C-style strings, character arrays!
- Your program must handle all errors, such as not entering the correct input for number of players or option for bowling.
- Each function, including main, may not have more than 15 lines of code (this doesn't include curly braces, variable declarations, comments, and blank spaces!).
- You are not allowed to use global variables.

```
How many players (1-5)? 2
Enter Player 1's name: Jennifer
Enter Player 2's name: Austin
```

[illegible][illegible][illegible]

Austin		5																5

Player 2, press enter to bowl.
 You knocked down 4 pins.

Name		1		2		3		4		5		6		7		8		9		10		Total

Jennifer		-	7																			7
			7																			

Austin		5	4																			9
			9																			

(10 pts) Extra Credit Dynamic Arrays

You can support N players, not just 1-5, using dynamic arrays allocated on the heap!!! Variable length arrays are not dynamically allocated on the heap, and therefore, you will not be given credit for these, i.e. `int array[num_players]`; In addition, you must not have an memory leaks to get the full 10 points. Make sure you use `valgrind`! ☺

(-10 pts) Automatic Deduction: You are not allowed to use global variables in any assignment in CS 161. There isn't any practical purpose for them in this course. Keep this in mind as you design your program with functions. Also, you must use C-style strings, not C++ strings!!!

(10 pts) Program Style/Comments

In your implementation, make sure that you include a program header in your program, in addition to proper indentation/spacing and other comments! Below is an example header to include. Make sure you review the style guidelines for this class, and begin trying to follow them, i.e. don't align everything on the left or put everything on one line!

http://classes.engr.oregonstate.edu/eecs/winter2017/cs161-001/161_style_guideline.pdf

You are graded on having a **header, function headers with pre/post conditions, proper comments, and readable code with indentation** and vertical spacing that is CONSISTENT throughout your program. DO NOT align your entire program on the left side. This will cause you to automatically lose the full 10 points. In addition, do not forget your program header!!!

Electronically submit your C++ program (**.cpp** file, not your executable!!!) by the assignment due date, using **TEACH**.