Unix shell scripts

Create a directory \texttt{username.2} in your home where \texttt{username} is your user name on delmar. Keep all programs and datafiles for this assignment in this directory. After you are done with the assignment, execute the following command:

\begin{verbatim}
% cd
% ~bhatias/bin/handin cs2750 2
\end{verbatim}

Write the code in bash. Make sure that temporary files, if any, are removed in the event of an interrupt.

1. **Olympic Time Trials.** Write a script to determine if you will qualify for the Olympic team. Prompt the user to specify the name and assign it to a variable. Verify that the name is not blank.

   Input five trial times. For example, let the times for Carl be 12.2, 11.8, 12.5, 10.9, and 11.1 seconds. Compute the average of those five times.

   For every person who tries, compute the average of the trials. If the average is less than or equal to 11.5 seconds, print the name and a message to say “Welcome to the team.” If the time is more than 11.5 seconds, the message should read “Close, but you did not make the cut.”

   Test your script with the data for three runners as:

   \begin{verbatim}
   Carlos 9.6 10.6 11.2 10.3 11.5
   Liu 10.6 11.2 9.4 12.3 10.1
   Timothy 12.2 11.8 12.5 10.9 11.1
   \end{verbatim}

2. **A Dice Game.** You can roll a virtual dice by generating a random number between 1 and 6. The score by rolling the dice twice is the sum of the two random numbers, and will range between 2 and 12.

   In the initialization phase of the script, compute the number of times each score can occur. You can achieve that in a nested loop. This loop will give you, for example, a count of 1 for the score of 2 (1 for each roll), a count of 2 for the score of 3 (1+2 and 2+1), and so on.

   Initialize your total reward to zero and enter a loop to play the game. Each iteration of the loop goes as follows. Roll the dice twice. If the two rolls are equal (you rolled a double), you receive a total reward of zero and the game terminates. If your score is 4, you receive $-50$ points. If your score is 10, you receive $-100$ points. Otherwise, you receive $(10 - P(s)) \times 10$ points. $P(s)$ is the number of possible ways that you can get a score of $s$ that you computed during the initialization phase. For example, $P(2) = 1$, $P(3) = 2$, and $P(12) = 1$.

   Towards the end of each iteration in the loop, print the score, the reward for the current iteration, and the total reward so far. Ask the user if he/she wants to continue the game. If the user answers 'y', continue back in the loop; otherwise terminate the game.