

Assignment Three

This assignment can be completed individually or in groups of 2 or 3

Date handed out: Sunday, March 15, 2020
Due Date: Saturday, April 4 (11:59 pm) via Blackboard

The purpose of this assignment is to provide you the opportunity to gain “hands on” experience with assembly language programming. Your task is to develop a number guessing game using the MIPS Assembly coding language. Your program must adhere to the following:

1. Prompt the user to enter an integer number N that will serve maximum (e.g., the number that will be guessed will range from $[0 \dots N]$)
2. Prompt the user to enter a seed S for the randomizer and initialize the randomizer
3. Using the randomizer initialized above, generate a random number R whose range is between $[0 \dots N]$
4. Prompt the user to guess a number X
5. Determine whether X is equal to R , “hot” (within 10% of R), or “cold” (greater than 10% of R)
6. Inform the user on whether their = X was correct (e.g., $X = R$) hot, or cold
7. If correct end program, If “hot” or “cold”, go back to step 4 (prompt the user to enter a number)

Bonus: Instead of “hot” or “cold” use “Very Hot” (X is within 5% of R), “Hot” (X is within 10% of R), “Correct”, “Cold” (X is within 10% to 25% of R), “Very Cold” (X is greater than 25% of R)

Note: You can assume that N , as entered, will be a valid positive integer and that N , as entered, will be large enough to ensure you can actually check for “very hot”, “hot”, “cold”, “very cold”. In other words, you are **NOT** required to perform and error checking on the user input.

What to Submit: Submit your solution. Your solution should compile and execute on its own without any additional code added. Your TA may assemble and execute the resulting program to test for the correct implementation. Your submitted code should also include (in the form of comments) the name of each group member and their corresponding student ID. You may have to provide a live working demo of your solution to your TA. This will involve downloading your submitted files from Blackboard and assembling/running them in the presence of your TA. **Programs that cannot be assembled may result in an automatic grade of 0.**