Project Report

1. The main function of the program is to “guess” the number the user is thinking about. 32 numbers are displayed on cards based on their binary value. Each number is representable by 6 bits in binary. The method we used to display this number is based on binary representation. The user inputs 1 if the number they’re thinking about is displayed and 0 if the number is not displayed. The values are then stored in the stack and used to calculate the integer value of the number the user was thinking about.
2. We encountered challenges when trying to figure out how to store the binary values from the user input. To solve this challenge, we decided to use the stack and simulate push and pop properties of the stack in order to move the data. Another challenge we encountered was how to display the 6 cards. We used bit masking and the stack to overcome this challenge. Also, I learned how hard it is to actually implement music or colors into assembly code and am thankful that high-level languages are the future.
3. I learned how to implement bit masking correctly. I also learned how to develop a method to interpret integer values as their binary representation in MIPS. I also learned how to properly read and write arrays in MIPS and overall make assembly code easier to read as our output file shows.
4. The first card displays every 6-bit binary number possible between 1 and 63 while keeping the first binary position as 1. The second car displays 32 numbers as well, but the second binary position is kept constant. This method is repeated throughout the remaining cards to display the correct values. In order to display the 6 cards, we masked the binary value of an integer with respect to its binary position, stored it in a register, and executed a loop that used bge to check if the number being displayed was less than 64. Using or, we incremented the integer by 2 binary values while skipping the binary position of the constant bit. This allowed us to traverse through all the numbers consistently and accurately When the user sees his number displayed, the user enters 1. Every time the user enters 1 or 0, the integer is pushed to the stack. When the user has gone through all the 6 cards, the values are restored from the stack and multiplied by 2 to the power of the respective bit position. The values are then added together and displayed as the number the user was thinking about.
5. Briscoe Fletcher and Nick Gover contributed to the development of the ideas and methods, problem solving, and the overall process of creating and analyzing the code. Omar Luna contributed to the creation and refinement of the code, as well as writing most of project report. I contributed to the development of the code, the analysis of our work, the narration and recording of the video and the guide to using our program. Everyone in the group did an excellent job of contributing, communicating, and getting things done with great quality and in a timely manner. I’m glad they were my teammates and wouldn’t have wanted it any other way.