Name: Hari Kiran Jana
Section: 2340.501

Mind Reader Game : Project Report

A. Description of the program: To play the game, we start by opening the MindReader.asm file in MarsPlus. Once we see the source code, we can run the file to begin the Mind Reader game. At the start of the game, the player will see a prompt message asking them if they want to play the game, and they can select ‘y’ or ‘n’ (yes or no, case-sensitive). If the user enters ‘n’, the program ends. However if the user enters ‘y’, then they are asked to think of a number between 1 - 63. Next, the game will display a card with 32 numbers and ask the user to see if the number that they were thinking of is on the card. If it is, then they will input ‘Y’, or ‘N’ otherwise (case-sensitive). This process will repeat until the user’s number is found. Furthermore, error messages were implemented to account for invalid values that the user enters throughout the game. At the end of the game, it will print a number on the console, in which the user can verify if that is the number they were thinking of. Finally, the player will see a prompt message asking them if they want to play the game again, and they can select ‘y’ or ‘n’ (yes or no, case-sensitive). If the user enters ‘y’, then the game will restart. If the user enters ‘n’, the program ends.

B. Some challenges we faced would be how to implement some of the ideas we came up with in HLL and translating them to mips. This was hard as MIPS commands require a lot more specificity than for example, c++ or Java. Moreover, we also had some issues figuring out how to store the bits in an array and how to mask them to get the numbers we wanted. Another major setback was the printing aspect of the game on generating the card and also dealing with the aspect of duplicates in a random generator.

Name: Hari Kiran Jana
Section: 2340.501

C. In this project I learned how to use loops, arrays and subroutines a lot better. I felt like before I had a vague idea on how to implement them. However, with this project I have come to understand how they work internally and how we can manipulate them to create a certain aspect of the mind reader game. Moreover, I never knew many functions like the random number generator existed in Mips and this was a great way for me to build and interact with them. Furthermore, I also understood how masks work and how to activate certain bits using a bitwise or operation as we did in the game. Overall, I felt like this game helped solidify my understanding of many mips concepts and also how to implement them given a certain scenario like this one.

D. Algorithms and techniques used in the program:

- Arrays, and base addressing:
    - Used to turn on the bits that are needed to find the magic number that the user is thinking of.
    - We use arrays and base addressing along with masked bits to both print the cards and also find magic numbers.
    - Certain bases are turned on when the user selects ‘Y’ and certain bits are turned off when they select ‘N’.
- Subroutines:
    - Used throughout the entire program. Separates the task given to us into smaller functions that we can reuse. For example, the errorChecker

subroutine checks for input validation and cardDis prints the final prediction result.

- Methods were implemented to create a perfect system that does not rely on hard coding the entire project.

- Looping:
  - Used for input validation. If the user enters a invalid input, they will be put into a loop until a valid input is reached.
  - Used to make the user play the game as many times as they want until they quit. If the user says no to the prompt that asks if they want to play the game, the program will end. Otherwise, the game will start.
  - Used to print the numbers on the cards. If the number has a certain bit activated, the number is printed on the card. Otherwise, move on to the number number.
- Branching and Jumping:
  - We used branching and jump instructions to write data to the PC.
  - We used brancing for statements that require other input from other areas of the code.
  - Jumping was used to loop back to a certain label so that validation, and printing could go smoothly.
- Bitwise operations and masks:
  - We create a mask and utilize the exclusive or operation to activate a particular bit in order to print certain numbers from 1-63 on a card. Also used to create the number that the user was thinking of.

Name: Hari Kiran Jana
Section: 2340.501

E. Peer Evaluation

Marcos Murillo (5/5):

- Created the video aspect of the project for our group. The video gives the player the basic format of the game and helps the user understand the basic structure of the game,
- Was proactive in creating a GitHub, so that everyone could contribute to the project on their own time. This allowed us to add things before the next meeting.
- Helped implement the validation part of our project that allowed us to see if there was anything wrong with user input.

Santana Lopez (5/5):

- Motivated the team and scheduled group meetings so that we could get work done.
- Made the user manual that shows a step by step process to run the program.
- Helped the group understand how we could use arrays and bit manipulation to implement some aspects of the project and provided clear explanations on them.

Anthony Ngo(5/5):

- Helped us create the algorithm for the random number generator and implement it.
- Researched topics like bit manipulation and helped us implement it. Also helped write the shared aspects of the project report.

- Set a tone for the group and helped us grasp the main concept of the project and also always came up with a solution when we were stuck.