David Wu

CS2340.004

a) The main purpose of this project was to recreate a functional code of the Hooda Math Multiplication Game, which in a brief description is when a player and a computer(AI) have to use the product of two numbers to get four consecutive parts in a 6x6 grid of numbers, either diagonally, horizontally, or vertically to win. These numbers that are being multiplied range from one through nine and the values in the grid range from one through eight-one while skipping a couple of numbers to fit the thirty-six square grid. Other purposes to this project as mentioned in the assignment information was to reflect on the MIPS code, making sure you not only understand how to code but also to improve our overall coding skills in MIPS. The project also allowed for our improvement in collaborating with one another, since the project called for us to create separate object modules that we eventually had to combine and debug together, which was one of the struggles that we as a team had overcome.
b) What I learned in the project is how array works in MIPS language. And I understand more about how to compare numbers in MIPS language. Later I also learned how does generating numbers work in MIPS language. Another thing I learned in this project is how to collaborate with teammates and integrating our code together and trying to understand others work. Last thing I become better is debugging the code by doing it step by step since there is always a lot of bug happening when combining or expanding codes.
c) The list is an array, so it is printed using a function that traverses each index. And then at every six numbers a new line is printed. The rest is just ascii symbols like | and + which are printed in between lines and columns to make the grid look nice.

User move check: traverses the array after the product has been calculated to make sure it is still available and updates to 11 if it is. If not the user has to re enter num1 or num2 until a valid product has been made.

Computer move: Random number generator from 1-2, and then 1-9. The number is multiplied by num1 if the random number generated 2, and vice versa. Then the same logic as the user move check is applied here.

Horizontal check: Every row has only 3 possibilities of having a row. So the loop checks the first three elements of each line and the 3 after that to see if there are consecutive elements.
Vertical check: The logic for this was a bit hard for this so we hard coded each possible instance a 4 in a row vertically could happen.

Diagonal check: This was also hard coded, but there were two parts: one left diagonal and right diagonal. There are 18 possible rows that can be made on the grid diagonally, 9 right and 9 left

d) Nachiket- Assist me a lot whenever I am struggling with my part of code. He mainly works on the grid part. He also contributes to changing the numbers taken to 11 or 22. He also made the array of numbers. A great teammate.

Aarez- He coded the random number generator to help assist the project, we used his code on generating numbers for npc move and also user moves where the slot is not taken. He also worked on the manual guide for the project.

Jousha- He coded the connect four to win the game, like when the four 11 or 22 matches up the program end and he also coded to decide who the winner is.