[9 points] – General Requirements

- [3 points] – coding style - proper comments, indentation
- [3 points] – correctness – compiles properly and gives correct output
- [3 points] – submission – no missing files, zip, properly submitted

[6 points] - dicegame.h file

- [2 points] - header guard
- [2 points] - struct
- [2 points] - function prototypes, enum

[15 points] - dicegame.c file

- [4 points] - getRandomNumber() – calculates and returns a random number within a range.
- [5 points] - fillRounds() – fills the DiceRound array created dynamically, with random values, using the random number function.
- [2 points] - getRoundPoints() – return the actual points to be used for the round, based on the game rules.
- [2 points] - printRoundInfo() - prints out the round info values.
- [2 points] - printPlayerInfo() – prints out the player points at the end of that round.

[20 points] - main.c file

- [2 points] - initialize srand to start the random generator.
- [2 points] - define and initialize P-1 and P-2 scores as well as other required variables.
- [2 points] - get the number of rounds from user and dynamically allocate memory for the struct array for DiceRound.
- [2 points] - check if dynamic allocation was successful and end the program if it was not.
- [1 point] - call fillRounds() to fill the array and printPlayerInfo() to print the initial zero values for player scores.
- [8 points] - implement the main gameplay with function calls and proper point calculations.
- [1 point] - Determine and print out the final outcome when the game is over - which player won.
- [2 points] - free the dynamically allocated array.