**Programming 1 – Fall 2020**  Student Name: \_\_\_\_\_\_\_\_\_\_\_\_\_Liam Martell\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**MP2 Scoresheet – 1 October 2020 (must be filled)**

**Source Code Validation – to be turned in by END OF DAY on Thursday**

**Source Code (the .cpp file), the screenshots, and the Scoresheet must be submitted via Moodle.**

1. CODE CHECK - Adherence to specifications (you must have these functions, they must have these prototypes, you must use the functions as defined here and exactluy as defined here):

|  |  |  |
| --- | --- | --- |
| **Function**  **Prototype?** | **Function**  **Definition?** |  |
| **YES** | no | **YES** | no | int rollDie( void ); |
| **YES** | no | **YES** | no | bool isTurnScoreLost( int die1value, int die2value ); |
| **YES** | no | **YES** | no | bool isGameScoreLost( int die1value, int die2value ); |
| **YES** | no | **YES** | no | char getUserInput( void ); |

**Second Chance (to be initialed by instructor/Lab TA):**

rollDie correct: \_\_\_\_\_\_\_ isTurnScoreLost correct: \_\_\_\_\_\_\_

isGameScoreLost correct: \_\_\_\_\_\_\_ getUserInput correct: \_\_\_\_\_\_\_

Instructor/Assistant Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ score: \_\_\_\_\_ / 20 points

1. Demonstrate the following tests of your program to either your instructor or one of the Lab Assistants. Tests do not have to be performed in order.

**Test #1:**

**Program displays human roll values – must play until you lose the turn. First Attempt Correct?** Y | N score: \_\_\_\_\_ / 5 points

**Test #2:**

**Program displays computer roll values – must stop at 25+ points. First Attempt Correct?** Y | N score: \_\_\_\_\_ / 5 points

**Test #3:**

**Program displays human roll values – must play one roll then stop to score. First Attempt Correct?** Y | N score: \_\_\_\_\_ / 5 points

**Test #4:**

**Program displays computer roll values – must lose the turn. First Attempt Correct?** Y | N score: \_\_\_\_\_ / 5 points

**Test #5:**

**Program correctly calculates and updates both turn and game scores. First Attempt Correct?** Y | N score: \_\_\_\_\_ / 5 points

**Test #6:**

**Game ends when 100 or more points are scored. First Attempt Correct?** Y | N score: \_\_\_\_\_ / 5 points

***(SCREENSHOTS AT END OF THIS DOCUMENT)***

**Second Chance (to be initialed by instructor/Lab TA):**

Test #1 correct: \_\_\_\_\_\_ Test #2 correct: \_\_\_\_\_\_ Test #3 correct: \_\_\_\_\_\_ Test #4 correct: \_\_\_\_\_\_ Test #5 correct: \_\_\_\_\_\_ Test #6 correct: \_\_\_\_\_\_

**Comments: Testing/Validation Subtotal: \_\_\_\_\_\_\_\_\_ / 50**

**Readability Subtotal: \_\_\_\_\_\_\_\_\_ / 30**

**Documentation Subtotal: \_\_\_\_\_\_\_\_\_ / 20**

**TOTAL: \_\_\_\_\_\_\_\_\_ / 100**

**Programming 1 – Fall 2020**  Student Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**MP2 Scoresheet – 1 October 2020 (must be filled)**

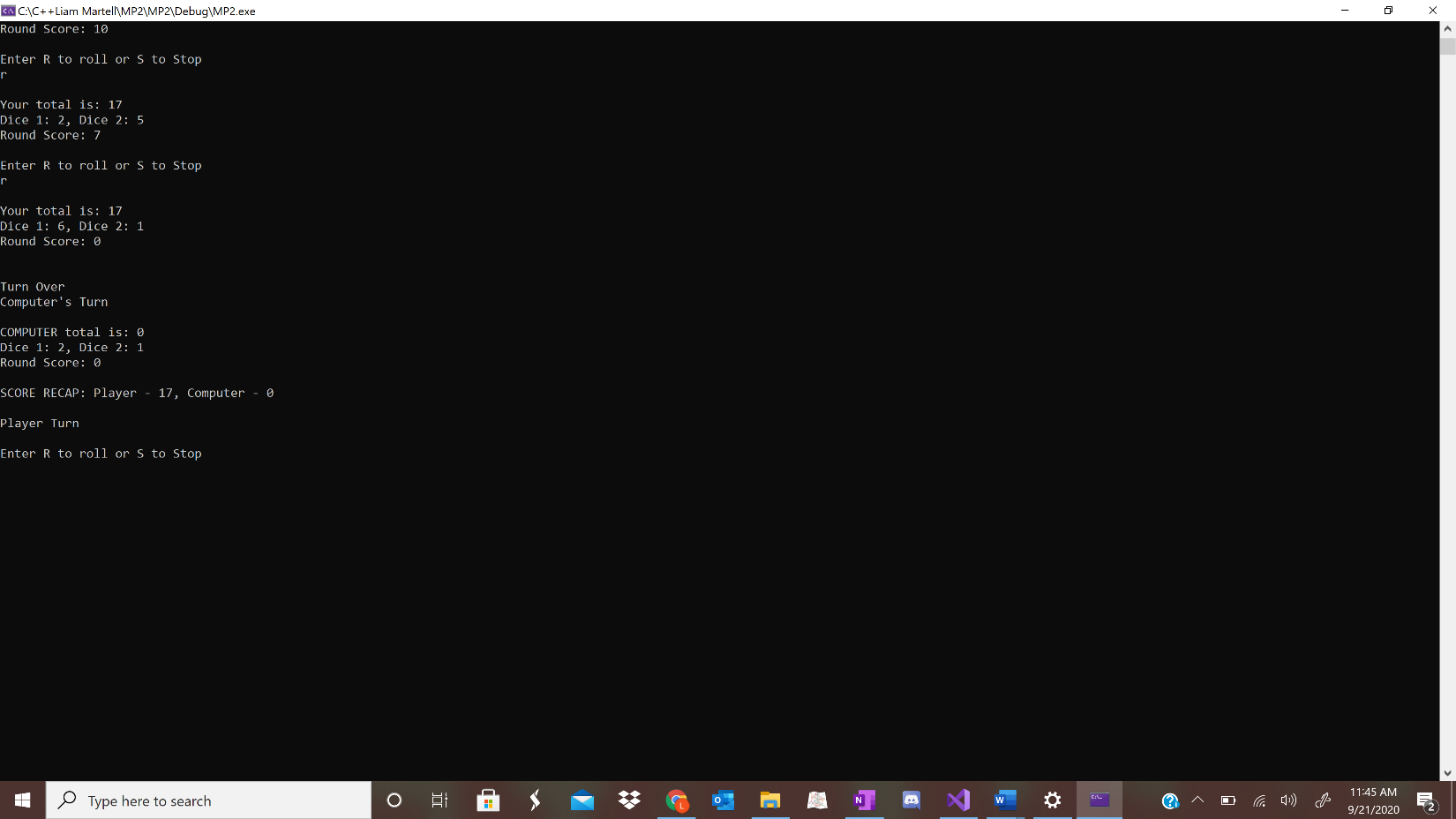
**Source Code Analysis Rubric (Note: This is slightly different from last time. New Section: Function Header)**

**Readability – 30 points total**

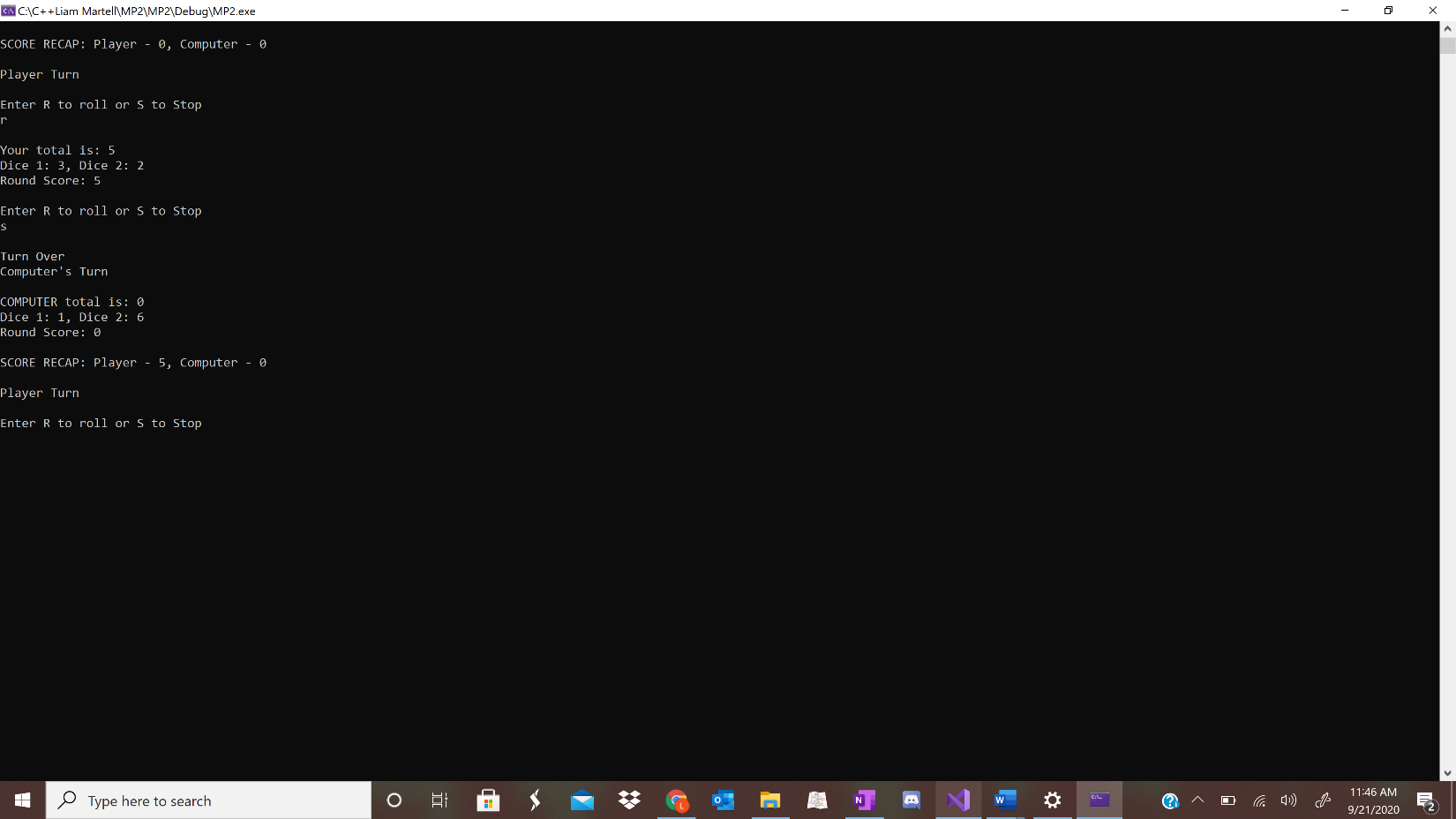
|  |  |  |  |
| --- | --- | --- | --- |
| **Criteria** | **Meets Expectations - 5** | **Needs Minor/Major Improvement – 4/3/2** | **Unacceptable/Missing - 0** |
| Organization  Score (x1): \_\_\_\_\_\_\_ | Code is broken down into clear, recognizable, well thought out sections of functional units; blank lines and comments used to establish visual structure. |  |  |
| Separation  Score (x1): \_\_\_\_\_\_\_ | Spaces used as appropriate to help differentiate distinct elements within each coding statement. |  |  |
| Alignment  Score (x1): \_\_\_\_\_\_\_ | Indentation emphasizes the body of an iterative or a conditional statement; braces and parentheses follow appropriate standards; start of comments are column-aligned as warranted. |  |  |
| Consistency  Score (x1): \_\_\_\_\_\_\_ | Similar coding constructs regularly use the same format regarding indentation and alignment; similar or related variable names follow an established pattern. |  |  |
| Nomenclature  Score (x2): \_\_\_\_\_\_\_ | All variables, save for common exceptions, have meaningful and informative names without being verbose; use of upper and lower case clearly differentiates variables, constants, and classes. |  |  |

**Documentation – 20 points total**

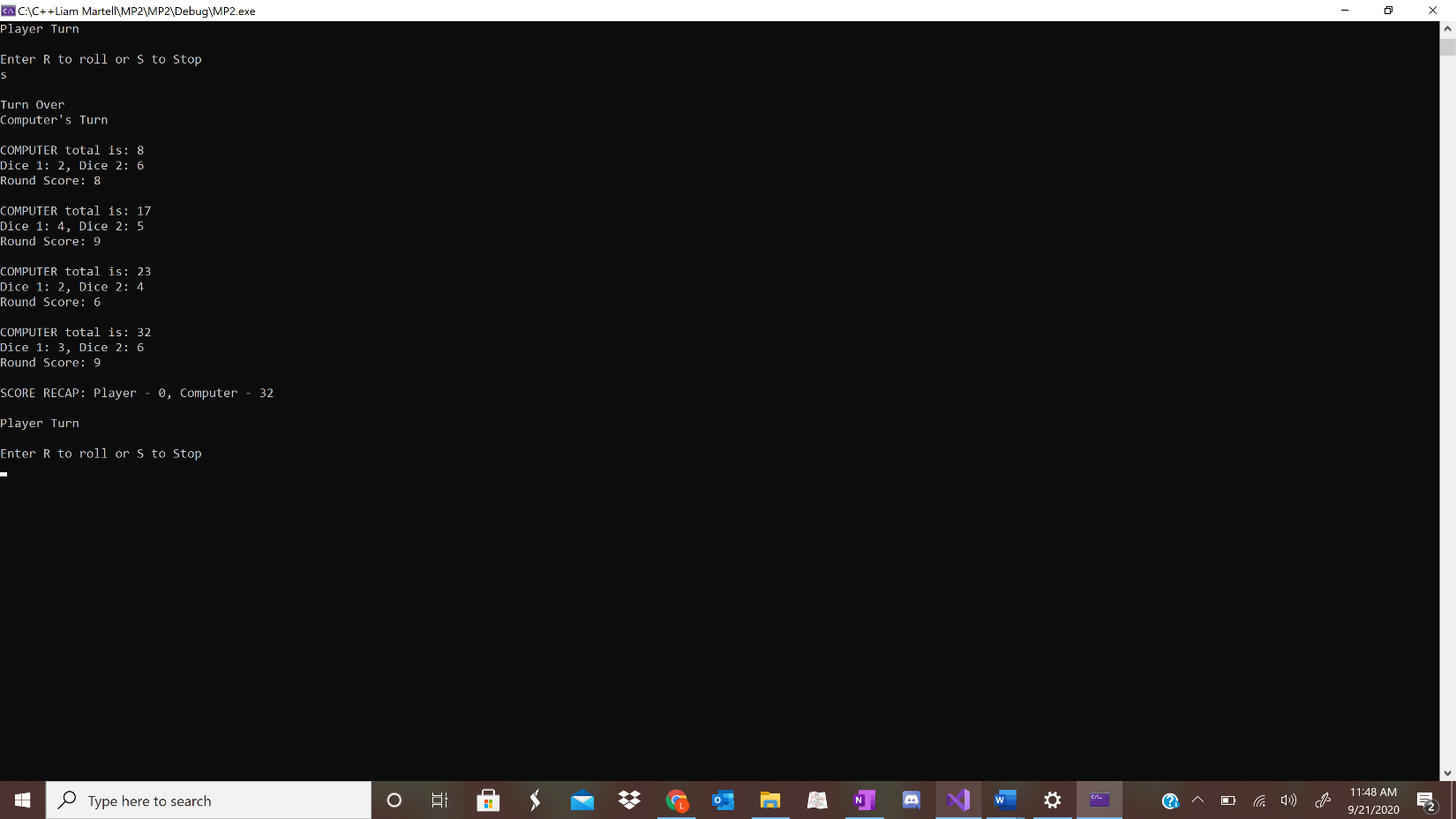
|  |  |  |  |
| --- | --- | --- | --- |
| **Criteria** | **Meets Expectations - 5** | **Needs Minor/Major Improvement – 4/3/2** | **Unacceptable/Missing - 0** |
| Program Header Comments  Score (x1): \_\_\_\_\_\_ | Every program starts with a header comment that contains the name of the file, the date of its writing, the full name of its author, and a description of what the program does. |  |  |
| Function Header Comments  Score (x1): \_\_\_\_\_\_\_ | All function header comments are with the definition (not prototype) and contain preconditions and postconditions as appropriate; each description clearly but succinctly explains purpose of the function. |  |  |
| Section Comments  Score (x1): \_\_\_\_\_\_\_ | Each functional section of code includes a comment describing the goal or purpose that that section is trying to accomplish without being either verbose or parroting. |  |  |
| Code Comments  Score (x1): \_\_\_\_\_\_\_ | Line-oriented comments are used to clarify meaning and/or provide elaboration as needed. |  |  |

**Screenshot 1: Shows Computer and Human loosing turn to a 1 rolled**

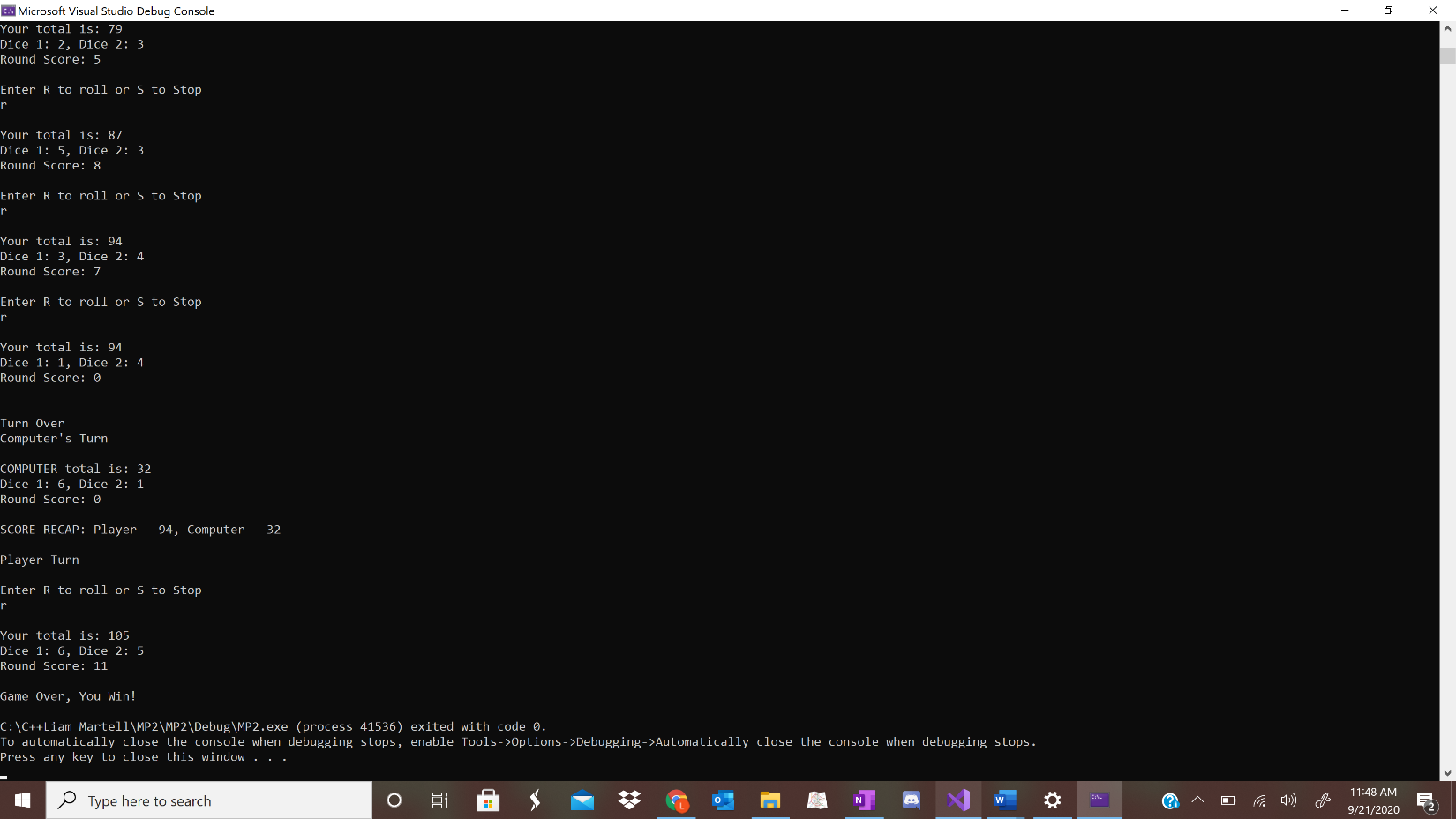
**Screenshot 2: Shows 1 roll then stop for human**

****

**Screenshot 3: Shows computer score end if 25+ points gained**

****

**Screenshot 4: Shows game won if points for 1 side total 1000 or more**

****