TONGOL, GABRIELLE VINCENZA T.

S21A

Word Scramble Program: ton77954.c

USER DEFINED FUNCTIONS

int randomNum() - returns 1 odd number from 1 - 100

No parameters

Variables:

num - will hold the random value generated

@returns num

Pre - condition: the loop will not stop until it encounters an odd number

void revealClues() - prints the letter clue

@param nChoice - player's menu choice

@param cWord - a string of the original word that is randomly generated

Pre - condition: nChoice should be 1 and 2 only

void displayJumbledWord() - prints the jumbled form of the original word

@param cWord - a string of the original word that is randomly generated

@param nWordCnt - how many words are currently answered by the player inside the program

- used only for printing

Pre - condition: cWord is only 1 word

int checkAnswer() - checks the player's answer if it is correct or not

@param cWord - a string of the original word that is randomly generated

@param answer - user input of answer

@returns 1 when the compared cWord and answwer is 0

Pre - condition: Since cWord is from the file, the \n should be removed at the end of the word or else even when both strings are the same it will always return 0.

int computePoints() - computes the points to be added or subracted in the player's existing points

@param cWord - a string of the original word that is randomly generated

@param answer - user input of answer

@param nClueCnt - how many clues are used, either 1 or 2 only

@returns nPnts - number of points depending on what condition is true
will later on be used to subtract or add points of the player

Pre - condition: length of word and checkAnswer function will also be used inside

void correctAnswer() - prints the statement when the player gets correct answer

@param cWord - a string of the original word that is randomly generated

@param answer - user input of answer

@param nClueCnt - how many clues are used, either 1 or 2 only

@param num - value that is inside the array of p struct

@param struct playerTag p - storage of the player's data

Pre - condition: the function ComputePoints will be used inside the function

void wrongAnswer() - prints the statement when the player gets wrong answer

@param cWord - a string of the original word that is randomly generated

@param answer - user input of answer

@param nClueCnt - how many clues are used, either 1 or 2 only

@param num - value that is inside the array of p struct

@param struct playerTag p - storage of the player's data

Pre - condition: the function computePoints will be used inside the function

void displayHighScorers() - displays the top 5 high scorers

@param struct playerTag p - storage of the player's data

Pre - condition: The array p holds 6 players but will only print 5