MIDDLE EAST TECHNICAL UNIVERSITY, NORTHERN CYPRUS CAMPUS
CNG140 C Programming – Programming Assignment 4

Date handed out: **Monday 27 April 2020**
Date submission due: **Monday 11 May 2020 23:55**

Programming Assignment 4: Code Breaker

Purpose:
The main purpose of this programming assignment is to revise functions, pointers, pass by value vs. pass by reference, arrays, strings and dynamic memory allocation.

Description

Code Breaker is a modern game played by two players with pegs. In this assignment you will implement a simple one-player game based on the given specifications below. The aim of the game is to break the secret code by guessing the exact positions of the colors in eight guesses.

Specifications

In this game, the computer will be the codemaker and the player will be the codebreaker. The codemaker chooses a pattern of four code pegs, each one being one of any of five colors (red, green, orange, blue, yellow). The codebreaker tries to guess the pattern, in both order and color, within eight turns.
After each guess, the computer gives you a score of exact and partial matches. Each guess is made by placing a row of code pegs on the decoding board. Once placed, the codemaker provides feedback by giving the following feedback:

- A black key peg is placed for each code peg from the guess which is correct in both color and position.
- A white key peg indicates the existence of a correct color code peg placed in the wrong position.
- Small hole is left empty if the guess is not a correct color.

In your implementation, computer will be the codemaker.

Rules

1. The sequence can contain pegs of colors: red, green, orange, blue, yellow.
2. A color can be used any number of times in the sequence.
3. All four pegs of the secret code will contain a color - no blanks/empties are allowed.
4. Each guess must consist of 4 peg colors - no blanks.
5. The player has 8 guesses to find the secret code.
6. If the player guesses the secret code in eight guesses, then s/he will win the game.

Scoring

- For each of the pegs in your guess that is the correct colour and in the correct position, the computer will give you one small black peg. If you score 4 small black pegs on a guess, you have guessed the secret sequence.
- For each of the pegs in your guess that is a correct colour in an incorrect position, the computer will give you one small white peg to the right of that move. Together, there will be no more than four small black and white pegs for each move.

- If none of the pegs in your guess is of a correct colour, you will see no small pegs to the right of that move.

Requirements:

1. The player should enter four colours as his/her guess. When the player enters the guess, then your program should display the guess and next to the guess it should display the score.

2. In your implementation, you can assign an integer number to each colour: Assume that black (1), white (0), red (2), green (3), orange (4), blue (5), yellow (6).

For example, assume that the secret code is Green, Orange, Yellow, Blue and player guesses the code after three guesses:

Secret Code: Green Orange Yellow Blue		Score
Guess 1:	Orange Green Blue Red	White White White "*Empty*"
Guess 2:	Green Orange Blue Yellow	Black Black White White
Guess 3:	Green Orange Yellow Blue	Black Black Black Black

In above example, Player guesses the secret code after three guesses and wins the game.

3. If the player does not guess the secret code after eight guesses then the computer (codemaker) wins the game.

3. After the player completes playing the game once (after either they win or they had 8 guesses), your program should ask the player if s/he would like to continue. If the player wants to continue then your program should generate a new code.

Your program should have the following functions:

Function Name	**Description**
`Create_code()_`	This function should generate a secret code that will be stored in a two-dimensional character array (an array of strings). It may generate a random number between 0 and 6 to represent colours and then assign the colours to this array of strings according to the randomly generated number.
`player_guess()`	This function will ask the player to input his/her guess and will store the guess in a two-dimensional character array (an array of strings).
`find_score()`	This function will get the guess, the code and will do scoring according to the guess. For each of the pegs in your guess that is the correct colour and in the correct position, the score for that peg will be one black peg. For each of the pegs in your guess that is the correct colour and in the incorrect position, the score will be one white peg. If the user guess does not exist in the secret code, then the score will be nothing (_). This will be stored and will be displayed by another function, see display_score() function below.
`display_score()`	This function will display the score which is calculated by the find_score function. The score can include a sequence of black, white or nothing (_). This function will also decide if the user breaks the code or not. If the user breaks the code, it will display the necessary messages and return true, otherwise it will return false.
display_winner()	This function will get either 1 or 0 from find_score() function and it will display the winner

Grading:

Your program will be graded as follows:

Grading Point	Mark(100)
create_code() Function	15
player_guess() Function	15
find_score() Function	25
display_score() Function	15
display_winner() Function	10
Maintaining 8 guesses	5
Ask the player if he/she wants to play again	5
Code quality (Appropriate comments, variable names, formulation of selection statements and loops, reusability, extensibility etc.)	10

Sample Run:

```
CODEBREAKER IS STARTING!
Secret Code Generated!
1.Enter your guess: Blue Yellow Red Green
Your score: White White _ Black
2.Enter your guess: Yellow Orange Blue Green
Your score: Black White White Black
3. Enter your guess: Yellow Blue Orange Green
Your score: Black Black Black Black
You win the game!
Secret code= Yellow Blue Orange Green!
You found it after 3 guesses!

Do you want to play again (Y/N)?Y
Secret Code Generated!
1.Enter your guess: Green Yellow Orange Blue
Your score: _ Black White White
2.Enter your guess: Red Yellow Blue Orange
Your score: Black Black Black Black
Secret code= Red Yellow Blue Orange
You found it after 2 guesses!

Do you want to play again (Y/N)?Y
Secret Code Generated!
1.Enter your guess: Green Green Blue Blue
Your score: _ _ _ _
2.Enter your guess: Red Red Blue Blue
Your score: White White _ _
3.Enter your guess: Red Blue Red Green
Your score: White _ White _
4. Enter your guess: Green Blue Red Green
Your score: _ _ White _
5. Enter your guess: Green Blue Green Red
Your score: _ _ _ Black
6. Enter your guess: Orange Blue Green Red
Your score: White _ _ Black
7. Enter your guess: Orange Yellow Green Red
Your score: White White _ Black
8. Enter your guess:  Yellow Orange Green Red
```

```
Your score: White Black_ Black
You LOST!
Secret code: Red Orange Yellow Red
CODEMAKER WINS!

Do you want to play again (Y/N)?N
BYEE!!
```

Please make sure that you follow the restrictions for the assignment as follows.

- You are not allowed to use global variables.
- Strictly obey the input output format. Do not print extra things.
- You are not allowed to use goto statement.
- Name your source file “StudentID.c”
- Upload only source file. Do not compress it (zip, rar, …)