MIDDLE EAST TECHNICAL UNIVERSITY, NORTHERN CYPRUS CAMPUS
CNG140 C Programming – Programming Assignment 2

Date handed out: **19 April 2021, Monday**
Date submission due: **02 May 2021, Sunday, 23:55, Cyprus Time**

Programming Assignment 2: MidnightDice

Purpose:

The main purpose of this programming assignment is to revise the topics that we have covered in the first six weeks including fundamentals of C programming, conditional statements, repetitive statements, and functions.

Description:

You will write a program for playing a MidnightDice game between a player and computer. MidnightDice is a dice game which is the variation of Midnight game. This will give you practice with all three control constructs (sequence, selection and repetition). We are including some design constraints in the "programming task" section, for example you will need to use functions. This will give you the experience of decomposing a problem into parts, and then implementing each part in one highly cohesive, loosely coupled parts as functions.

Don't try to compile your entire program in one "**big bang**". Compile it piece by piece. Test each piece that you have compiled to make sure it works correctly before you add the next piece.

MidnightDice Rules:

Equipment: 6 dices and a scoresheet
Number of Players: 2 players

How to play:
At the beginning to decide who is going to start the game, each player rolls one die. The player who rolls the highest die starts the game. The game consists of N rounds – each player will play N turns. The game is played with 6 dices. The first player rolls all six dice to start his/her turn. The player must keep at least two scoring combination each time he/she rolls by setting the scoring dice to the side. He/She then picks up the remaining dice and rolls them. Again, at least two scoring combination must be kept aside, and the remaining dice rolled. This sequence continues until there are no more dice to roll.

Scoring:
At each round, players must have kept aside one dice with a value **1** and one dice with a value **4 (both needs to be kept)**, or their score will be zero for that round. If they have kept one dice with a value **1** and one dice with a value **4**, the other dice are totalled to give the player's score for that round. For example, if the user kept the following dice: 1, 4, 6, 5, 2, 2 then their score is 15. If they kept the following dice: 1, 2, 3, 6, 5, 6 then their score is zero as they do not have both 1 and 4. Therefore, the maximum score for a round is 24 (four 6s.). The procedure is repeated for the remaining rounds. The player with the highest total score wins.

Scoresheet: A Scoresheet looks as follows:

Player 1	Player 2
15	24
38	24

As you can see scores are accumalted from the previous round.

How to Play MidnightDice?

You will write the program that will allow a player to play the MidnightDice game against the computer. The game consists of N rounds. At the end of N rounds, whoever has the highest score will win the game. User will choose the number of rounds, N.

A sample run is as follows where "My Turn" is representing a computer!

```
Welcome to the MidnightDice game.
Lets get started!

How many rounds would you like to play? 2

I have rolled the dice and got 3!
Shall I roll the dice for you (Y/N)? Y
I have rolled the dice for you and you got 1!

Round 1 -- My Turn:
```

```
I got → [Dice 1]: 2 [Dice 2]: 1 [Dice 3]: 4 [Dice 4]: 3 [Dice 5]: 4 [Dice 6]: 4
Kept dice 2 and 3
I got → [Dice 1]: 2 [Dice 4]: 3 [Dice 5]: 6 [Dice 6]: 5
Kept dice 5 and 6
I got → [Dice 1]: 2 [Dice 4]: 2
My score: 15

Round 1 -- Your Turn:
```

```
Are you ready to play!
Shall I roll them for you (Y/N)? Y
You got → [Dice 1]: 6 [Dice 2]: 3 [Dice 3]: 2 [Dice 4]: 2 [Dice 5]: 4 [Dice 6]: 3
Which ones you want to keep?  1 5
Shall I roll the remaining for you (Y/N)? Y
You got → [Dice 2]: 1 [Dice 3]: 2 [Dice 4]: 6 [Dice 6]: 1
Which ones you want to keep?  2 4
Shall I roll the remaining for you (Y/N)? Y
You got → [Dice 3]: 6 [Dice 6]: 6
Your score: 24

Our scoresheet:
====================
My score    Your score
15          24

Round 2 -- My Turn:
```

```
I got → [Dice 1]: 2 [Dice 2]: 1 [Dice 3]: 4 [Dice 4]: 3 [Dice 5]: 4 [Dice 6]: 5
Kept dice 2 and 3
I got → [Dice 1]: 6 [Dice 4]: 5 [Dice 5]: 6 [Dice 6]: 2
Kept dice 1 and 5
I got → [Dice 4]: 6 [Dice 6]: 5
My score: 23

Round 2 -- Your Turn:
```

```
Are you ready to play!
Shall I roll them for you (Y/N)? Y
You got → [Dice 1]: 4 [Dice 2]: 1 [Dice 3]: 2 [Dice 4]:2 [Dice 5]: 6 [Dice 6]: 4
Which ones you want to keep?  1 5
Shall I roll the remaining for you (Y/N)? Y
You got → [Dice 2]: 6 [Dice 3]: 2 [Dice 4]: 6 [Dice 6]: 5
Which ones you want to keep?  2 4
Shall I roll the remaining for you (Y/N)? Y
You got → [Dice 3]: 6 [Dice 6]: 6
Your score: 0
```

```
Our scoresheet:
====================

My score    Your score
38          24

I AM THE WINNER!
```

Programming Requirements:

In order to implement this game, you will need to write at least the following functions, but if you need more functions you can add them.

roll-a-dice () – This function will roll a dice and return the result. The rolling action will give a random value representing a possible dice value.

play_computer() – This function will mainly be responsible from making the computer play the game.

computer_strategy_decider() –This function will decide which strategy will be used for keeping dice. In summary, the computer will always play safe and have a very simple strategy. As soon as a dice shows 1 or 4, computer will keep them (since without keeping them, round will be scored as zero), otherwise computer will always choose the highest dice values to have higher score. For example; if the dice values are 1,5,4,6,6,2 the computer will choose to keep 1 and 4. However, if the dice values are 1,5,3,6,6,2 the computer will choose to keep 1 and 6, since the 6 will be added to score. Similarly, if the dice values are 3,2,2,5,5,6 then the computer will keep 5 and 6.

play-user() – This function will mainly be used to get the player play a turn.

scoresheet() – This function will be used to display the scoresheet on the screen.

Grading Schema:

Your program will be graded as follows:

Grading Point	**Mark (100)**
Maintaining the number of rounds requested by the user and also maintaining the total scores	10
roll-a-dice() function	10
play_computer() function	20
computer_strategy_decider() function	20
play_user() function	20
scoresheet() function	10
Code quality (e.g., formatting, commenting, naming variables, clean use of C constructs such as formulation of selection statements and loops, etc)[1]	10

Rules:

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

- **Strictly obey the input output format. Do not print extra things.**
- **You are not allowed to use global variables.**
- **You are not allowed to use data structures such as arrays to store values as we have not covered them in the class yet.**
- **Add your name/surname and ID at the top of your code as comments and name your source file "Name-Surname-StudentID.c"**
- **Submit your solution as C and PDF to odtuclass. Do not compress it (zip, rar, …).**

[1] See guidelines given here: https://www.gnu.org/prep/standards/html_node/Writing-C.html