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

Date handed out: 10 April 2023, Monday
Date submission due: 27 April 2023, Thursday, 23:55, Cyprus Time

Programming Assignment 2: Drop Dead

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:

Drop Dead is a dice game, created by United States soldiers as an alternative to poker. You will write a program for playing a Drop Dead game between a player and computer. 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.

Drop Dead Game Rules:

Equipment: 5 dice 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 5 dice. The first player rolls all five dice to start his/her turn. A player rolls the five dice and if the throw does not include a 2 or 5, they receive the score of the total numbers added together. That player is also able to roll the dice again. When a player rolls the dice and any of them containing a 2 or 5, they score no points and the dice that includes a 2 or 5 is excluded from any future throws that they make. A player's turn does not stop until their last remaining die shows only a 2 and/or 5. At that point, the player "drops dead" and it becomes the next player's turn. The highest total score wins.

Scoresheet: A Scoresheet looks as follows:

Computer	Player 2
8	28
28	43

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

Scoring:
Below you can see an example turn:

Numbers Thrown	Throw's Score	Total Score
1, 3, 4	8	8
3, 4, 6	13	21
1, 4, 5	0	21
6, 4	10	31
1, 6	7	38
2, 5	0	38

Scores are accumulated from the previous round.

How to Play Drop Dead?

You will write the program that will allow a player to play the Drop Dead 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. The user will choose the number of rounds, N.

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

```
Welcome to the Drop Dead game.
Let's get started!

How many rounds would you like to play? 2

I have rolled the dice and got 3!
I have rolled the dice for you and you got 1!

Round 1 -- My Turn:
```

```
I got → [Dice 1]: 1 [Dice 2]: 2 [Dice 3]: 2 [Dice 4]: 4 [Dice 5]: 5
My score: 0 Total score: 0
Dice 2 Dice 3 Dice 5 are excluded!

I got → [Dice 1]: 3 [Dice 4]: 4
My score: 7 Total score: 7

I got → [Dice 1]: 1 [Dice 4]: 4
My score: 5 Total score: 12

I got → [Dice 1]: 6 [Dice 4]: 4
My score: 10 Total score: 22

I got → [Dice 1]: 1 [Dice 4]: 6
My score: 7 Total score: 29

I got → [Dice 1]: 2 [Dice 4]: 5
Drop Dead!
My score: 0 Total score: 29


Round 1 -- Your Turn:
```

```
Are you ready to play!
I rolled them for you and you got
→ [Dice 1]: 1 [Dice 2]: 6 [Dice 3]: 6 [Dice 4]: 4 [Dice 5]: 4
Your score: 21 Total score: 21
I rolled and you got
→ [Dice 1]: 2 [Dice 2]: 5 [Dice 3]: 1 [Dice 4]: 5 [Dice 5]: 4
Your score: 0 Total score: 21
Dice 1 Dice 2 Dice 4 are excluded
→ [Dice 3]: 5 [Dice 5]: 5
Drop Dead!
Your score: 0 Total score: 21


Our scoresheet after round 1:
============================
My score    Your score
29          21
```

```
Round 2 -- My Turn:
```

```
I got → [Dice 1]: 2 [Dice 2]: 3 [Dice 3]: 1 [Dice 4]: 1 [Dice 5]: 1
My score: 0 Total score: 29
Dice 1 are excluded!
I got → [Dice 2]: 5 [Dice 3]: 2 [Dice 4]: 2 [Dice 5]: 2
Drop Dead!
My score: 0 Total score: 29
```

```
Round 2 -- Your Turn:
```

```
Are you ready to play!
I rolled them for you and you got
→ [Dice 1]: 1 [Dice 2]: 1 [Dice 3]: 1 [Dice 4]:1 [Dice 5]: 1
Your score: 5 Total score: 26

I rolled and you got
→ [Dice 1]: 2  [Dice 2]: 2 [Dice 3]: 2 [Dice 4]: 5 [Dice 5]: 3
Dice 1 Dice 2 Dice 3 Dice 4 are excluded!
Your score: 0 Total score: 26

I rolled and you got
→ [Dice 5]: 2
Drop Dead!

Our scoresheet after round 2:
============================

My score    Your score
29          26

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. The computer will only choose to roll dices again if they are not 2 or 5. When the computer has the last die with 2 or 5. The turn will finish for the computer.

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:

If your code does NOT compile, you will automatically get zero. If your code compiles, you will then be graded based the following scheme:

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	25
play_user() function	35
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

If you do not obey the rules then you will automatically get 0.

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 and goto statements.**
- **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