MIDDLE EAST TECHNICAL UNIVERSITY, NORTHERN CYPRUS CAMPUS
CNG 140 C Programming – Programming Assignment 1

Date handed out: **7 August 2023, Monday**
Date submission due: **18 August 2023, Friday, 23:55, Cyprus Time**

Programming Assignment 1: Cacho Alalay

Purpose:

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

Description:

Cacho Alalay is a popular dice game from Latin America. It is similar to Yahtzee/Yatzy. The purpose of the game is to roll five dice and score points from their combinations. 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.

Cacho Alalay 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. If two players get the same scores then your program should ask the players to roll again. The game consists of N rounds – each player will play N turns. There are 10 categories and players get up to 3 rolls to score in each category. After each roll, the player can set aside any number of dice and roll the remaining dice. The game is played with 5 dice.

The first player rolls all five dice to start his/her turn (this is his/her first turn). The player can keep as many dice as he/she wants and reroll the remaining dice. This is the player's second roll. Again, the player can keep as many dice as he/she wants and reroll the remaining dice. This is player's third roll (final). Then the next player starts. The highest total score wins.

Scoresheet:

A scoresheet looks as follows:

Computer	Player
8	23
28	53
58	63

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

Scoring:
Below you can see the scoring table to calculate scores:

Category	Description	Score
Ones	At least two dices showing the number one	Total of ones rolled
Twos	At least two dices showing the number two	Total of twos rolled
Threes	At least two dices showing the number three	Total of threes rolled
Fours	At least two dices showing the number four	Total of fours rolled
Fives	At least two dices showing the number five	Total of fives rolled
Sixes	At least two dices showing the number six	Total of sixes rolled
Escalera (Straight)	Five sequential dice (1-2-3-4-5 or 2-3-4-5-6)	25 if in first roll 20 for second or third roll
Un Full (Full House)	A three-of-a-kind and a pair (Ex. 3-4-3-3-4 or 5-5-1-1-5)	35 if in first roll 30 for second or third roll
Poker (Four of a kind)	Four dices showing same number (Ex. 1-1-1-6-1 or 3-2-3-3-3)	45 if in first roll 40 for second or third roll
Grande (Five of a kind)	All dices showing same number (Ex. 6-6-6-6-6 or 5-5-5-5-5)	50

The table above shows the scoring table to calculate scores for different dice combinations in each round. It is important that on cases where multiple scoring categories are possible to execute, you should pick the one with the highest score possible. The scores acquired in each round are accumulated in the score sheet according to this table.

How to play Cacho Alalay?
You will write the program that will allow a player to play the Cacho Alalay game against the computer. The game consists of N rounds. At the beginning of the game, the computer and the player will throw a dice to find who will start first and at the end of N rounds, whoever has the highest score will win the game. In case of equality of scores at the end, a new round will be added to break the tie. The user will choose the number of rounds, N.

A sample run is provided below. As the computer plays, "I" refers to the computer and "You" refers to the player. Please check it thoroughly. (The highlighted text is the user input)

Sample Run:

```
Welcome to the Cacho Alalay 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 3!

Rerolling again!

I have rolled the dice and got 4!
I have rolled the dice for you and you got 1!

I go first!
```

```
Round 1 --- Computer:
---------------------
I got -> [Dice 1]: 2  [Dice 2]: 1  [Dice 3]: 5  [Dice 4]: 3  [Dice 5]: 5
I am keeping dice 3 and 5!
I got -> [Dice 1]: 2  [Dice 2]: 2  [Dice 4]: 2
I got Un Full!
I am keeping dice 1, 2, and 4!
My score: 30, Total score: 30

Round 1 --- Player:
-------------------
Get ready to play!
I rolled dices for you and you got
-> [Dice 1]: 6  [Dice 2]: 3  [Dice 3]: 2  [Dice 4]: 2  [Dice 5]: 4
Current round score: 4
Which ones do you want to keep? 3 4
I rolled the remaining and you got:
-> [Dice 1]: 1  [Dice 2]: 2  [Dice 5]: 2
You got Poker!
Current round score: 40
Which ones do you want to keep? 3
You have already excluded Dice 3! Please re-try
Which ones do you want to keep? 5
I rolled the remaining and you got:
-> [Dice 5]: 1
You got Grande!
Your score: 50, Total score: 50

Our scoresheet after round 1:
=============================
Computer    Player
30          50

Round 2 --- Computer:
---------------------
I got -> [Dice 1]: 1  [Dice 2]: 2  [Dice 3]: 3  [Dice 4]: 4  [Dice 5]: 5
I got Escalera!
I am keeping dice 1, 2, 3, 4, and 5!
My score: 25, Total score: 55

Round 2 --- Player:
-------------------
Get ready to play!
I rolled dices for you and you got
-> [Dice 1]: 3  [Dice 2]: 3  [Dice 3]: 3  [Dice 4]: 5  [Dice 5]: 6
Current round score: 9
Which ones do you want to keep? 1 2 3
I rolled the remaining and you got:
-> [Dice 4]: 2  [Dice 5]: 5
Current round score: 9
Which ones do you want to keep? 0
Rerolling remaining dices again!
-> [Dice 4]: 3  [Dice 5]: 1
You got Poker!
Your score: 40, Total score: 90

Our scoresheet after round 2:
=============================
Computer    Player
55          90


PLAYER WINS!
```

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 try to keep the scoring dices and roll the rest.

***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 have similarities with another student or taken from a website you will automatically get zero.
If your code compiles, you will then be graded based on the following scheme:

Grading Point	**Mark (100)**
Maintaining the number of rounds, the total scores and adding extra round if necessary.	10
roll-a-dice() function	5
play_computer() function	25
play_user() function	35
scoresheet() function	10
Input/Output format and gameplay	5
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:

Make sure to follow the restrictions below for the assignment otherwise you will automatically get 0.

- **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 comment and name your source file as "Name-Surname-StudentID.c" (Ex. Zekican-Budin-1234567.c)**
- **Submit your solution as C and PDF to ODTUClass. Do not compress it (no zip, rar, …)**
- **Beware that using code generators may lead to code similarities which may lead to receiving 0.**

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