C Programming Language
#Project – 7

The Monopoly Game

In this assignment, you are asked to develop a well-known board game that is named Monopoly. Monopoly, real-estate board game for two players, in which the player's goal is to remain financially solvent while forcing opponents into bankruptcy by buying and developing pieces of property. Each side of the square board is divided into 6 equal divisions by forming 20 small rectangles representing specific properties, punishments and taxes. At the start of the game, each player is given a fixed amount of money to spend while playing; the players then move around the board by throwing a dice. Each player should obey the rules mentioned below. Any player who lands on an unowned property may buy it, but, if he or she lands on a property owned by another player, rent money must be paid to the owner of the land. A player continues to travel around the board until he or she gets bankrupt. Bankruptcy results in elimination from the game. The last player remaining on the board is the winner.

In the assignment, you need to code several functions to develop the game. The functions must be obtained to the following rules:

- The game is played by two users. Both users in the game should be controlled by a person. The players start the game from the start block.
- Each player can roll one dice on his / her turn and moves according to number that is on the dice.
- If the user arrives to an unowned property, then the user can buy the property if he / she wants it and she / he can construct some houses before the turn pass over to the other user (up to 3 house).
- The user only can buy a property or construct a house on the property when the user has arrived to the block of the property. The user can construct a house on each time he / she arrived to his/her owned property.
- If the user arrives to an owned property, then the user must pay the rent of the property to the owner of the property. The rent can be changed according to number of the houses constructed on the property.
- When the user arrived a tax block, then the user must pay the tax fee of the block.
- The punishment blocks hold the arrived user according to punishment count. In an example; if the punishment says: “Wait 2 turn” then the user cannot roll the dice before the against user roll the dice three times.

- When a user doesn't have enough money to pay his / her rent or tax then the bank asks him / her to sell one or more properties which are owned by the user. If the user can't pay his / her payment even all the owned properties are sold, then the user lost the game.

- When a user pass through the start block, then the user gains ten thousand dollars on each time.

In this assignment, we strongly expect from you to use structures. For that reason, you need to declare two structures for blocks and players. The fields of the structures are explained below:

struct player:

- player_type type : Holds type of the player. The player_type must be defined as enumerated type which consist of *noone, cap* and *car* values. int current_block_id : Holds
- player location as block id.
- int owned_block_ids[11] : Holds property block ids that are owned by the user.
- int account : Holds amount of the current money of the player.
- int turn_to_wait : Holds number of turns to wait if the player got a punishment.
- char * name : Holds the name of the player.

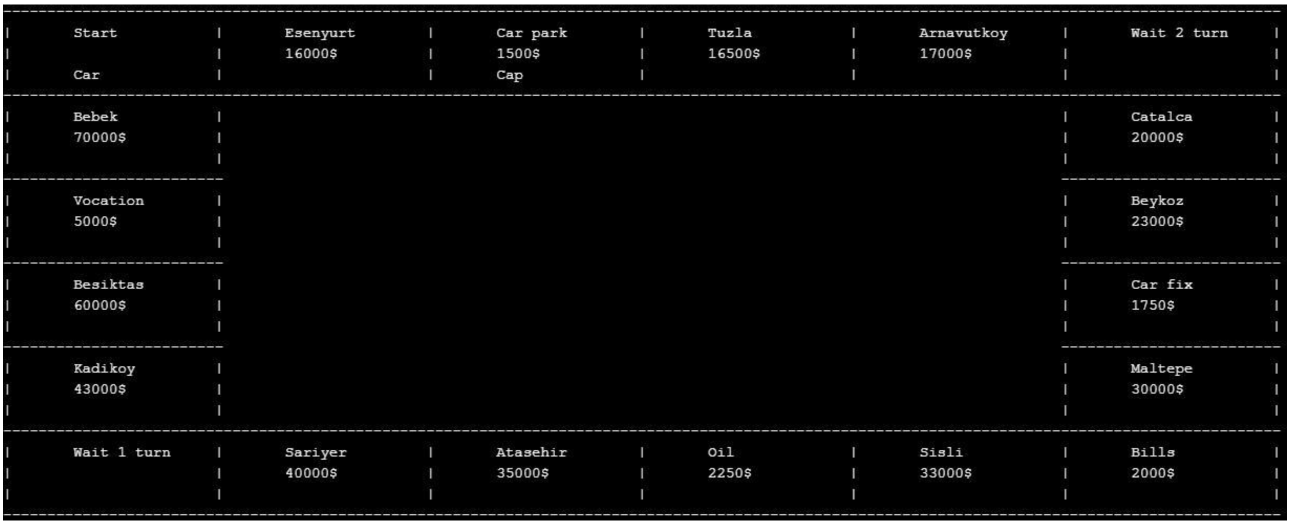
struct block:

- int block_id : Holds id of the block.
- char * name : Holds text of the block that is shown on the top of the block.
- int price : If the block is property, holds the price of property. If the block is a tax block than holds the tax fee and if the block is punishment block, then holds the number of turns to wait.
- int rent : Holds the default rent of a property (no house).
- int rent_1 : Holds the rent of a property with one house.
- int rent_2 : Holds the rent of a property with two houses.
- int rent_3 : Holds the rent of a property with three houses.
- int house_price : Holds price of building a house on the block.
- int house_count : Holds the number of the houses on the block that are already built.
- struct player owner : Holds the owner of the block.
- block_type type : Holds type of the block. The block_type must be defined as enumerated type which consist of *start, property, tax, punish* values.

Part 1. Write a function that takes board as an input and initializes the board with the blocks. The information of the blocks is given in file blocks.xlsx. Do not try to read blocks programmatically, just read and prepare the board according to data in the Excel file. Remember that, block ids represent the order of the blocks in the board. The function prototype is:

```
void init_the_board(struct block board[20])
```

Part 2. Write a function that prints the board as shown below. The blocks consist of three lines. The first line of the blocks prints text of the block, the second line print's the price of the block and the last line prints the users that are on the block.

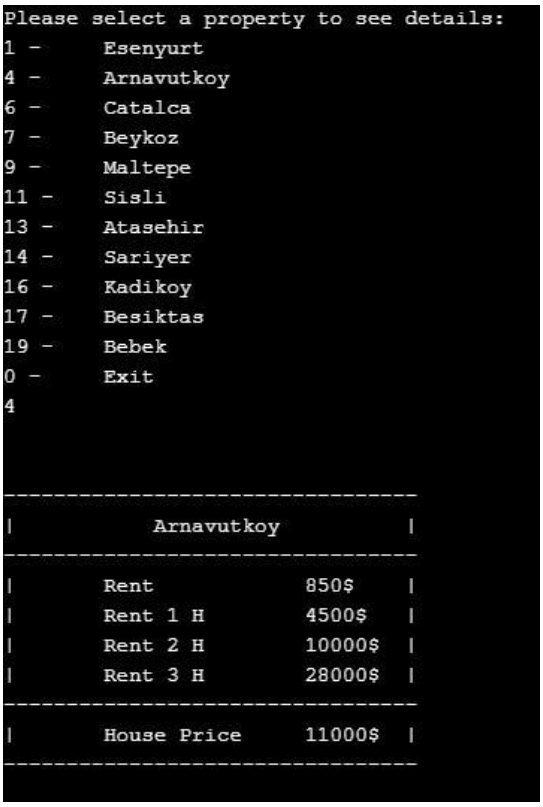


```
-------------------------------------------------------------------------------------------------------------------------------------------------
|       Start           |       Esenyurt        |       Car park        |       Tuzla           |       Arnavutkoy      |       Wait 2 turn     |
|                       |       16000$          |       1500$           |       16500$          |       17000$          |                       |
|       Car             |                       |       Cap             |                       |                       |                       |
-------------------------------------------------------------------------------------------------------------------------------------------------
|       Bebek           |                                                                                               |       Catalca         |
|       70000$          |                                                                                               |       20000$          |
|                       |                                                                                               |                       |
-------------------------                                                                                               -------------------------
|       Vocation        |                                                                                               |       Beykoz          |
|       5000$           |                                                                                               |       23000$          |
|                       |                                                                                               |                       |
-------------------------                                                                                               -------------------------
|       Besiktas        |                                                                                               |       Car fix         |
|       60000$          |                                                                                               |       1750$           |
|                       |                                                                                               |                       |
-------------------------                                                                                               -------------------------
|       Kadikoy         |                                                                                               |       Maltepe         |
|       43000$          |                                                                                               |       30000$          |
|                       |                                                                                               |                       |
-------------------------------------------------------------------------------------------------------------------------------------------------
|       Wait 1 turn     |       Sariyer         |       Atasehir        |       Oil             |       Sisli           |       Bills           |
|                       |       40000$          |       35000$          |       2250$           |       33000$          |       2000$           |
|                       |                       |                       |                       |                       |                       |
-------------------------------------------------------------------------------------------------------------------------------------------------
```

The function prototype is:

```
void show_board(struct block board[20],struct player player_one, struct
player player_two)
```

Part 3. Write a function that allows a user to select a property and to see details of its property deed. The function takes the board as an input and shows all properties to the user on the board. The user selects one of them and the function shows the details of the deed as shown below:



```
Please select a property to see details:
1 -     Esenyurt
4 -     Arnavutkoy
6 -     Catalca
7 -     Beykoz
9 -     Maltepe
11 -    Sisli
13 -    Atasehir
14 -    Sariyer
16 -    Kadikoy
17 -    Besiktas
19 -    Bebek
0 -     Exit
4



---------------------------------
|           Arnavutkoy          |
---------------------------------
|       Rent            850$    |
|       Rent 1 H        4500$   |
|       Rent 2 H        10000$  |
|       Rent 3 H        28000$  |
---------------------------------
|       House Price     11000$  |
---------------------------------
```

The function prototype is:

```
void show_properties(struct block board[20])
```

Part 4. Write a function that takes the block of a property and a user as input and if the user is available to buy the property, the function allows the user to buy the property. All controls (is the property unowned, is the user on the block of the property, do the user have enough money to buy this property etc.) must be done in the function. Remember that if the user buys the property, the function must ask the user to learn if he / she wants to build one or more house on the property before he / she lose his / her turn. The prototype of the function is:

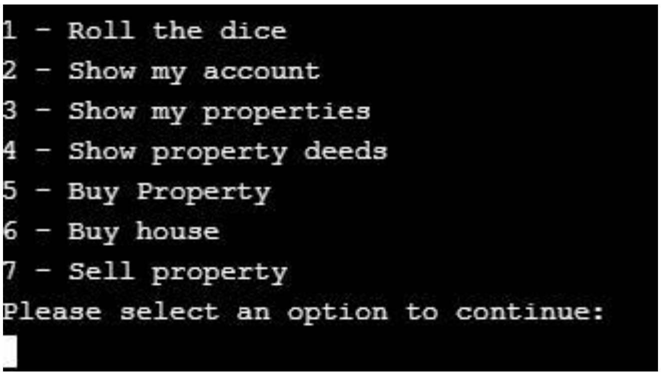
```
void buy_property(struct block* current_block, struct player*
                          current_player)
```

Part 5. Write a function that takes the board and a user as input and allows the user to sell the properties that are owned from the user to the bank. Remember that bank only pays back half of the price of the property and the houses that are on the property. The user can sell one or more properties until he / she wants to exit the function or he / she sold all his / her properties. The function prototype is:

```
void sell_property(struct block board[20], struct player* current_player)
```

Part 6. Write a function that takes board and users as an input to allow two users to play the Monopoly game. The function must provide:

- Keep playing until the one of the players get bankrupt. If a player can't pay his payment even after selling all his / her properties, then the player can be acknowledged as bankrupt.
- Show a menu to the players when it is their turn, to help them to play the game. The menu is shown below.
- If the player arrives to an unowned property, the program must ask if he / she wants to buy the property or not. If he / she buys the property, then the program asks if he / she wants to build one or more house on the property or not.
- If the player arrives an owned property, then the he / she must pay the rent of the property.
- If a user doesn't have enough money in his / her account when he / she needs to pay for something, then the program must request him / her to sell some property.



```
1 - Roll the dice
2 - Show my account
3 - Show my properties
4 - Show property deeds
5 - Buy Property
6 - Buy house
7 - Sell property
Please select an option to continue:
```

```
void gameplay (struct block board[20], struct player player_one, struct
                        player player_two)
```