# C Programming Language
# #Project - 2

**Part 1.** Write a menu function which displays the menu as exactly as below. You must use while loop and switch case to implement this menu. If user enters an invalid choice (such as entering a negative number, 0, and more than 4), then you must warn user by saying “This is an invalid choice. Try again!”.

```
***** MENU *****
 1. Play Lucky Number
 2. Draw Hourglass
 3. Draw Mountain Road
 4. Exit
 Choice:
```

**Function:** void menu()

**Part 2.** Program randomly picks a lucky number (LN) that is between 1 and 1024 (inclusive). You make series of guesses (GN) to find LN. If the distance between LN and GN is:

512 – 1023, program says “Distance is 10.”
256 – 511, program says “Distance is 9.”
128 – 255, program says “Distance is 8.” 64
– 127, program says “Distance is 7.”
32 – 63, program says “Distance is 6.”
16 – 31, program says “Distance is 5.”
8 – 15, program says “Distance is 4.”
4 – 7, program says “Distance is 3.”
2 – 3, program says “Distance is 2.”
1, program says “Distance is 1.”
0, then you win the game.

If you do not successfully guess LN with T trials, program wins. T is determined by you. You must use the following functions to implement this game.

**Random Number Generation:**

1. Include time.h library
2. Type srand(time(NULL)) at the beginning of your main program
3. Use rand() function to get a random value

**Function:** int make_a_guess (int trial, int min, int max)
This function helps you to make better guesses by warning you about limits of your guess. Limit values are min and max. At first, this function prints min and max values. Then, it gets the input guess from you. Finally, returns your guess.

**Function:** void show_scores(int score_human, int score_program)
This function stores scores (number of wins) of both human and the computer through the execution of whole program. For instance, user can play this game first. Then, game ends and user goes back to menu. After using other parts of this project, user can go back to this game and play more. At each play, score must be updated cumulatively. Scores are displayed after playing this game as exactly as below:

**Gameplay Example:** (With having T=10 & LN = 918)

```
(Trial: 1) Make a guess between 1 and 1024: 200
Distance: 10
(Trial: 2) Make a guess between 200 and 1024: 500
Distance: 9
(Trial: 3) Make a guess between 500 and 1024: 750
Distance: 8
(Trial: 4) Make a guess between 750 and 1024: 900
Distance: 5
(Trial: 5) Make a guess between 900 and 1024: 930
Distance: 4
(Trial: 6) Make a guess between 900 and 930: 915
Distance: 2
(Trial: 7) Make a guess between 915 and 930: 918
Distance: 0. You win!
Your Score: 5 Program Score: 3 (This means; user wins 5 out of 8 game play.)
```

**Part 3.** In this part, you take height of the hourglass from user and print it on screen using "*" character. Height of the hourglass must be odd number. If user types an even number, you must warn him/her, and ask again for valid height value.

**Function:** void draw_hourglass (int height)

**Output Example:**

```
              Height = 1          Height = 3          Height = 5
                    *                ***                *****
                                      *                  ***
                                     ***                  *
                                                         ***
                                                        *****
```

**Part 4.** In this part, you draw a mountain road using "/", "\", and "|" characters. Adjacent half circles form a mountain road. To do so, you take length N and the maximum radius (R) of half circles from user. Then, your program generates N half circles with randomly generated radius between 0 and R. Half circles are adjacent. They follow each other with left-right basis. Please examine the example for better understanding.

Function: void draw_mountain_road (int length, int max_radius)

**Output Example:** (N = 3, R = 5. Let’s say randomly generated radius values are 2,3,1.)

```
      /
     /
    |          R1 = 2
     \
      \
       \
        \
         \
          |    R2 = 3
         /
        /
       /
      /
     |         R3 = 1
      \
```