# Time execution

-In this paper we will try to check the time execution of our game and see how changing our variables will affect on the time taken to execute the game.

The function used to calculate the execution time is:

To calculate time taken by a process, we can use **clock()** function which is available time.h. We can call the clock function at the beginning and end of the code for which we measure time,

We divide by CLOCKS_PER_SEC (the number of clock ticks per second) to get processor time, like following.

the algorithm:

```
#include <time.h>

    clock_t start, end;

    double cpu_time_used;


    start = clock();

    ... /* Do the work. */

    end = clock();

    cpu_time_used = ((double) (end - start)) / CLOCKS_PER_SEC;
```

We already inserted this function in our game implementation as shown in the next page:

```c
void startGame() {
 clock_t t;
 t = clock(); //start counting time


 char gameBoard[BOARD_HORIZONTAL][BOARD_VERTICAL];
 memset(gameBoard, ' ', BOARD_HORIZONTAL * BOARD_VERTICAL); // Allocate memory to the board using memset

 int thePlayerChoice = 0; // Player choice flag

 do {
     printf("\n\n Welcome to Connect 4");
     printf("\n\n Please choose an option below\n\n");

     printf("\n 0. Show Instructions");
     printf("\n 1. Show Help");
     printf("\n 2. Play Game");

     printf("\n 3. Play vs Computer");
     printf("\n 4. Load Game");
     printf("\n 5. Quit");

t = clock() - t; //end counting time
    double time_taken = ((double)t)/CLOCKS_PER_SEC; // in seconds

    printf(" \n the game took %f seconds to execute \n", time_taken);
```

And after executing with the default game variables which are:

```c
#define PLAYER_NAME_SIZE 128
#define INPUT_SIZE 256
#define BOARD_HORIZONTAL 7 //horizontal slots for the game board
#define BOARD_VERTICAL 6 // Vertical slots for the game board
#define PLAYER_MOVES 42 // Number of player moves
```

The result is:

```
Welcome to Connect 4

Please choose an option below


0. Show Instructions
1. Show Help
2. Play Game
3. Play vs Computer
4. Load Game
5. Quit
the game took 0.009000 seconds to execute
```

## Increasing the size of the gameboard:

After changing game variables as following: (increasing the size of the gameboard from 7x6 to 9x8)

```
#define PLAYER_NAME_SIZE 128
#define INPUT_SIZE 256
#define BOARD_HORIZONTAL 9   //horizontal slots for the game board
#define BOARD_VERTICAL 8 // Vertical slots for the game board
#define TRUE 1
#define FALSE 0
#define PLAYER_MOVES 72 // Number of player moves
```

The execution time is:

```
Welcome to Connect 4

Please choose an option below


0. Show Instructions
1. Show Help
2. Play Game
3. Play vs Computer
4. Load Game
5. Quit
the game took 0.016000 seconds to execute
```

Conclusion: execution time increased .

## Decreasing the size of the gameboard:

By changing the game variables as following: (decreasing the size of the gameboard from 7x6 to 6*5)

```
#define PLAYER_NAME_SIZE 128
#define INPUT_SIZE 256
#define BOARD_HORIZONTAL 6   //horizontal slots for the game board
#define BOARD_VERTICAL 5// Vertical slots for the game board
#define TRUE 1
#define FALSE 0
#define PLAYER_MOVES 30 // Number of player moves
```

The execution time is:

```
Welcome to Connect 4

Please choose an option below


0. Show Instructions
1. Show Help
2. Play Game
3. Play vs Computer
4. Load Game
5. Quit
the game took 0.007000 seconds to execute
```

Conclusion: execution time decreased.