Tic-Tac-Toe Game

A PROJECT REPORT

Submitted by

PRACHI (RA2111026010111)

CH MALVIKA REDDY (RA2111026010125)

ANURADHA KRISHNAN (RA2111026010134)

Under the guidance of

Dr. M. SINDHUJA

In partial fulfillment for the award of the degree of

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE ENGINEERING

With specialization in Artificial Intelligence and Machine Learning

Of

FACULTY OF ENGINEERING AND TECHNOLOGY

SRM

INSTITUTE OF SCIENCE & TECHNOLOGY
(Deemed to be University u/s 3 of UGC Act, 1956)

S.R.M. Nagar, Kattankulathur, Kancheepuram District

ACKNOWLEDGEMENT

We would like to express our deepest gratitude to our guide, Dr. M. Sindhuja for her valuable guidance, consistent encouragement, personal caring, timely help and providing us with an excellent atmosphere for doing our project. All through the work, in spite of her busy schedule,she has extended cheerful and cordial support to us for completing this project work.

INDEX

INTRODUCTION TO C

C is a powerful general-purpose programming language. C programming is an excellent language to learn, to program for beginners. It was originally developed at Bell Labs by Dennis Ritchie between 1972 and 1973 to construct utilities running on UNIX. C++ is a superset of C, which means it is based upon C.

It is used for developing:

- Database systems
- Graphics packages
- Word processors
- Spread sheets
- Operating system
- Compilers and Assemblers
- Network drivers
- Interpreters

DID YOU KNOW?

Interpreted languages like Python, Ruby, and PHP have their primary implementations written in C.

INTRODUCTION TO VISUAL STUDIO CODE

Visual Studio Code is a source-code editor made by Microsoft for Windows, Linux and macOS. Features include support for debugging, syntax highlighting, intelligent code completion, snippets, code refactoring, and embedded Git. It can be used with a variety of programming languages, including Java, JavaScript, Go, Node.js,Python, C and C++. It is a code editor redefined and optimized for building and debugging modern web and cloud applications. C/C++ support for Visual Studio Code is provided by a Microsoft C/C++ extension to enable cross-platform C and C++ development.

REQUIREMENTS FOR PROJECT

Hardware Required:

- 4 GB free RAM
- 3.5 GB disk space

Software Required:

- Windows Operating System
- Visual Studio Code (with C/C++ extension)
- MS Word
- Google Chrome

INTRODUCTION OF PROJECT

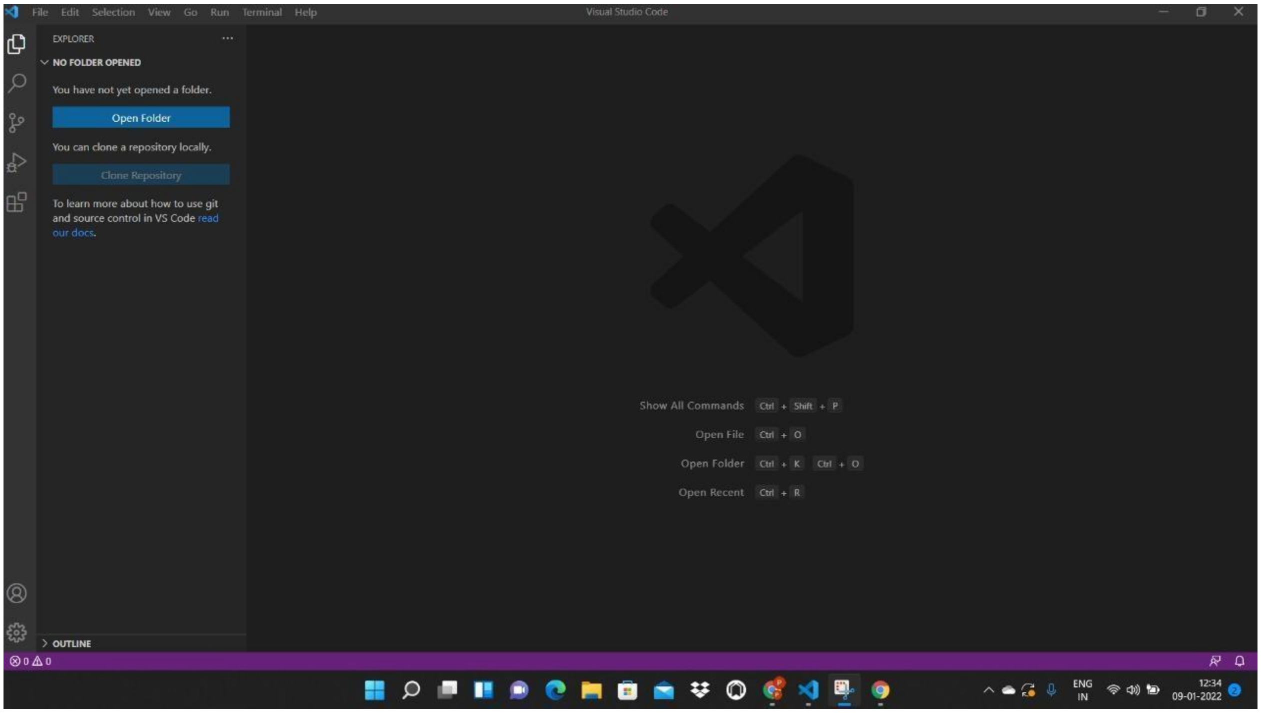
Tic-tac-toe (American English), noughts and crosses (Commonwealth English), or Xs and Os (Irish English) is a paper-and-pencil game for two players who take turns marking the spaces in a three-by-three grid with *X* or *O*. The player who succeeds in placing three of their marks in a horizontal, vertical, or diagonal row is the winner. It is a solved game, with a forced draw assuming best play from both players.

Objectives achieved from the project-

In this project, we have created a simple tic-tac-toe game, which is a player v/s computer game. While making a Tic Tac Toe game using C language, it is important to make use of arrays. The Xs and Os are kept in different arrays, and they are passed between several functions in the code to keep track of how the game goes. With the code here you can play the game choosing either X or O against the computer.

SOURCE CODES

Visual Studio Code

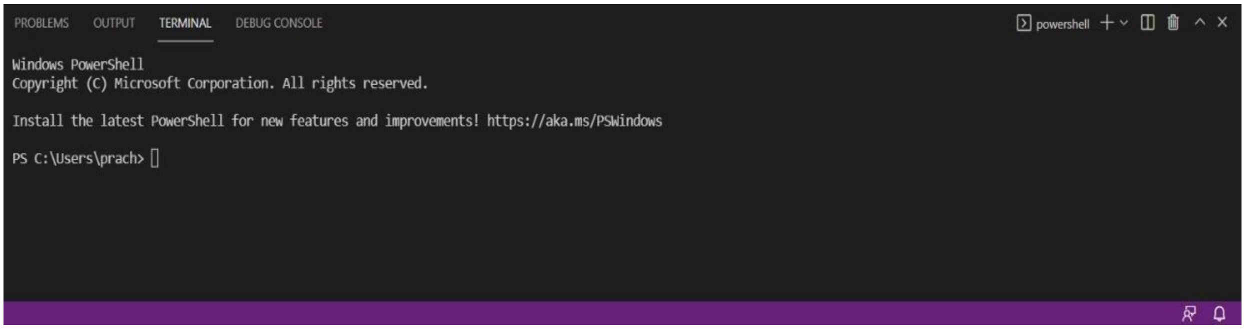


You have not yet opened a folder.

You can clone a repository locally.

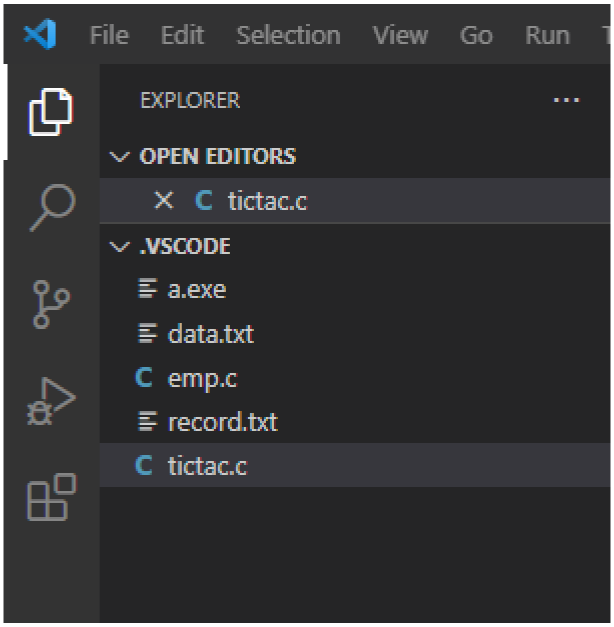
To learn more about how to use git and source control in VS Code read our docs.

Terminal



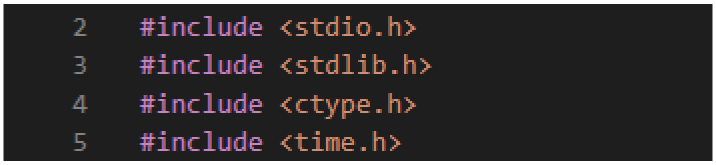
```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\prach>
```

First, we have to create a new file with .c extension. The file name here is tictac.c. Following changes occur on the screen.



After creating the file, following coding part has been done for the project.

Header files-



```
#include <stdio.h>
#include <stdlib.h>
#include <ctype.h>
#include <time.h>
```

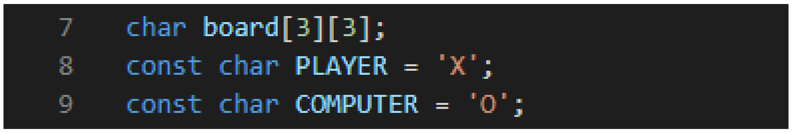
- ➢ **#include <stdio.h>-** This header file has the necessary information to include the input/output related functions in our program. Example printf(), scanf().

- ➢ **#include <stdlib.h>-** This header file is for Standard Library. <stdlib.h>" contains declaration of malloc() and free(). "<stdlib.h>" contains header information for „Memory Allocation/Freeing‟ functions. Functions such as srand() & rand() are used from the above header file.

- **#include <ctype.h>-**This header file of the C Standard Library declares several functions that are useful for testing and mapping characters. All the functions accepts **int** as a parameter, whose value must be EOF or representable as an unsigned char. Function such as toupper() is used from the above header file.

- **#include <time.h>-** This header defines four variable types, two macro and various functions for manipulating date and time. Function such as time() is used from the above header file.

Global Variables-

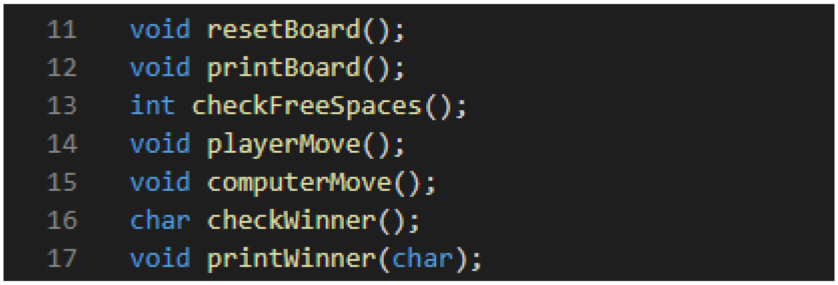


```
char board[3][3];
const char PLAYER = 'X';
const char COMPUTER = 'O';
```

An array named “board” of datatype char is defined for creating the overall structure of tic-tac-toe game and for giving the values of X & O.

Two variables “PLAYER” and “COMPUTER” are declared with datatype char and initialized with values X and O respectively, and these values remain constant throughout the program.

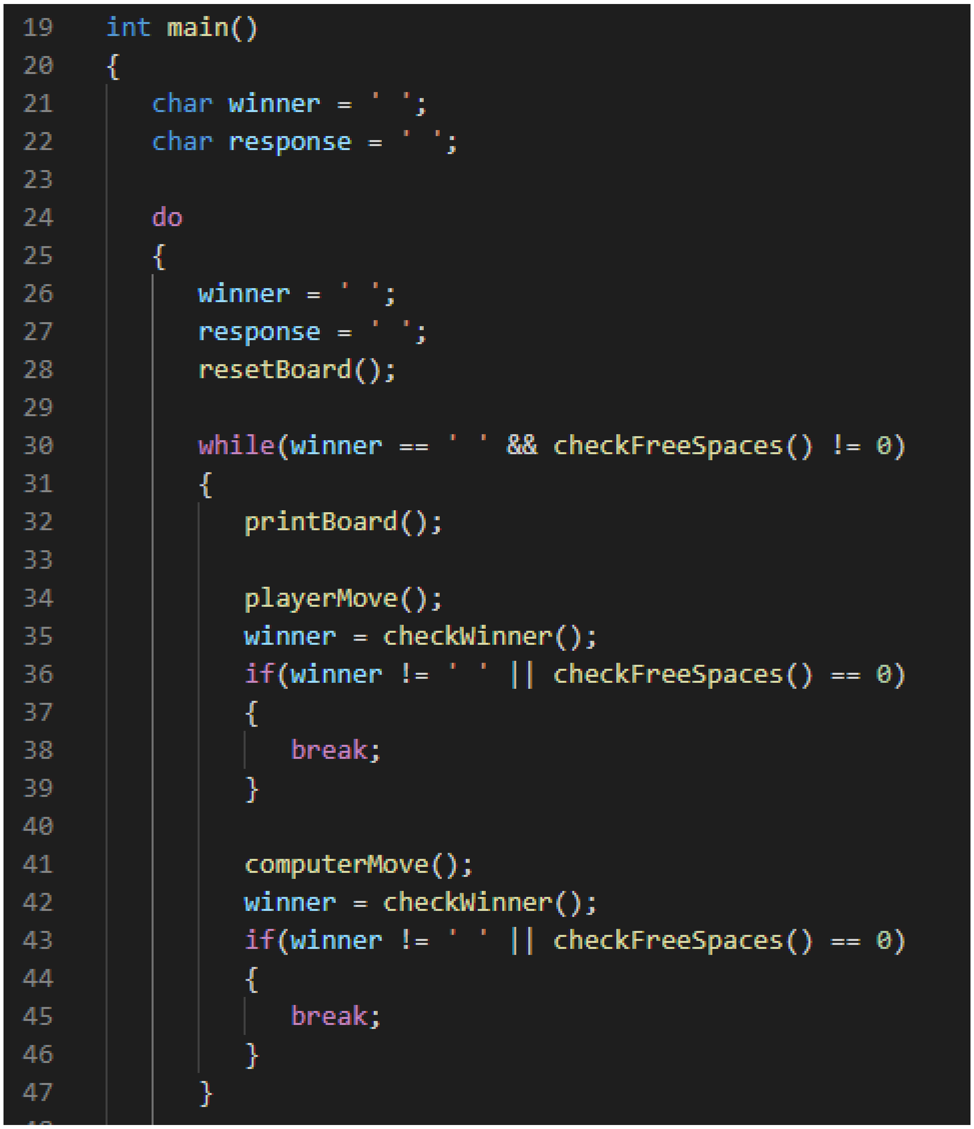
User defined functions-



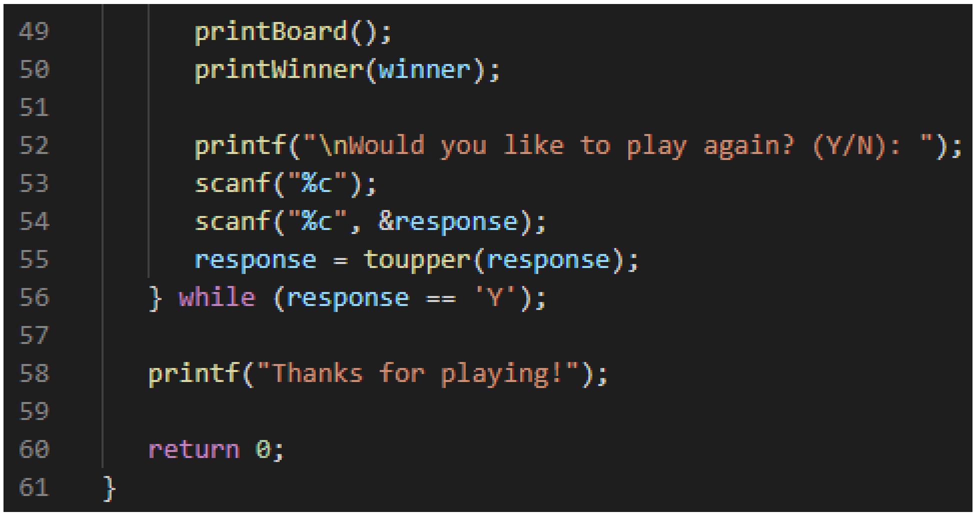
```
void resetBoard();
void printBoard();
int checkFreeSpaces();
void playerMove();
void computerMove();
char checkWinner();
void printWinner(char);
```

We will see these functions in detailed form later.

main() function-



```
int main()
{
    char winner = ' ';
    char response = ' ';

    do
    {
        winner = ' ';
        response = ' ';
        resetBoard();

        while(winner == ' ' && checkFreeSpaces() != 0)
        {
            printBoard();

            playerMove();
            winner = checkWinner();
            if(winner != ' ' || checkFreeSpaces() == 0)
            {
                break;
            }

            computerMove();
            winner = checkWinner();
            if(winner != ' ' || checkFreeSpaces() == 0)
            {
                break;
            }
        }
```



```
        printBoard();
        printWinner(winner);

        printf("\nWould you like to play again? (Y/N): ");
        scanf("%c");
        scanf("%c", &response);
        response = toupper(response);
    } while (response == 'Y');

    printf("Thanks for playing!");

    return 0;
}
```

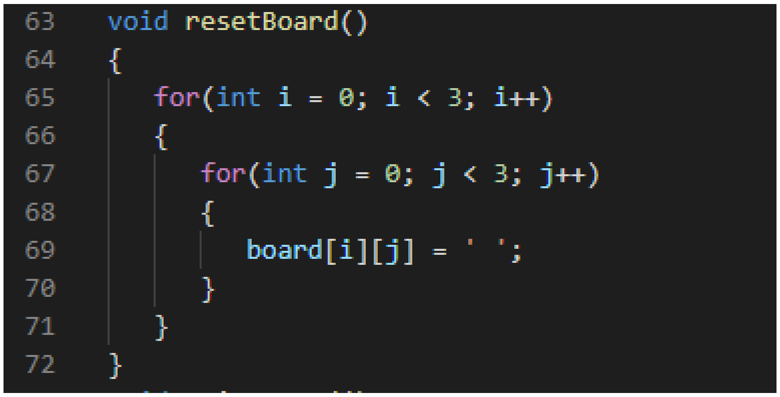
The main function is used to call the user defined functions.

toupper()- This function is used to convert lowercase alphabet to uppercase. i.e. If the character passed is a lowercase alphabet then the toupper() function converts a lowercase alphabet to an uppercase alphabet.

Here we have also asked the user if he/she wishes to play again, if yes then the response is Y.

do…..while loop- A do while loop is similar to while loop with one exception that it executes the statements inside the body of do-while before checking the condition. On the other hand in the while loop, first the condition is checked and then the statements in while loop are executed. So you can say that if a condition is false at the first place then the do while would run once, however the while loop would not run at all.

void resetboard()-



```c
void resetBoard()
{
    for(int i = 0; i < 3; i++)
    {
        for(int j = 0; j < 3; j++)
        {
            board[i][j] = ' ';
        }
    }
}
```

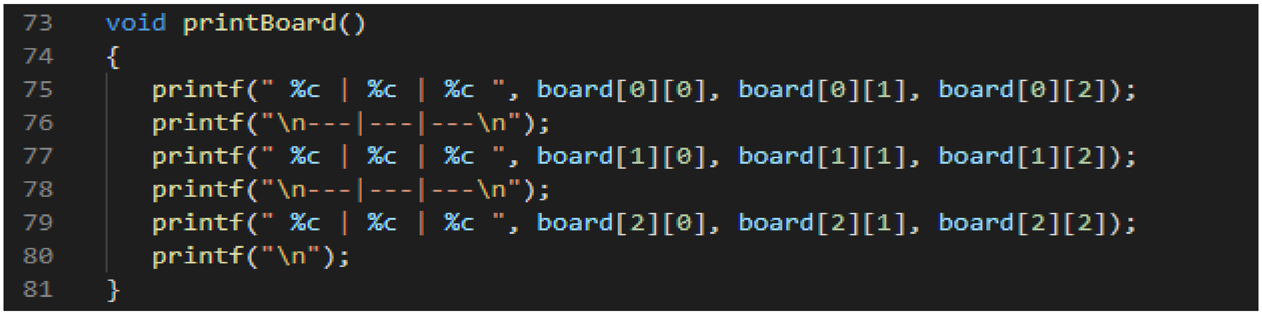
This function is used to reset the board to the original state.

This is done by replacing all the values of the board[][] array to ‘ ‘, a blank space.

Here a nested for loop is used inorder to access all the elements of the 2D array

This is a void function, hence there is no return value.

void printboard()-



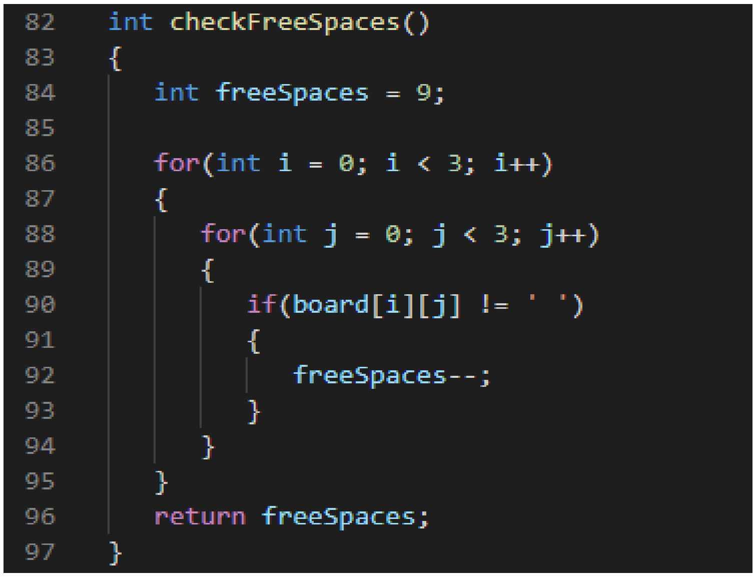
```c
void printBoard()
{
    printf(" %c | %c | %c ", board[0][0], board[0][1], board[0][2]);
    printf("\n---|---|---\n");
    printf(" %c | %c | %c ", board[1][0], board[1][1], board[1][2]);
    printf("\n---|---|---\n");
    printf(" %c | %c | %c ", board[2][0], board[2][1], board[2][2]);
    printf("\n");
}
```

This function is used to create the specific board for tic-tac-toe game.

This function is used to print the lines on the screen within which the ‘X’ and ‘O’s wil get placed. During the printing, the elements of the 3*3 char array ,board[][], gets printed, each of them separated by | and - - - .

As this too is a void function it does not return any value.

int checkFreeSpaces()-



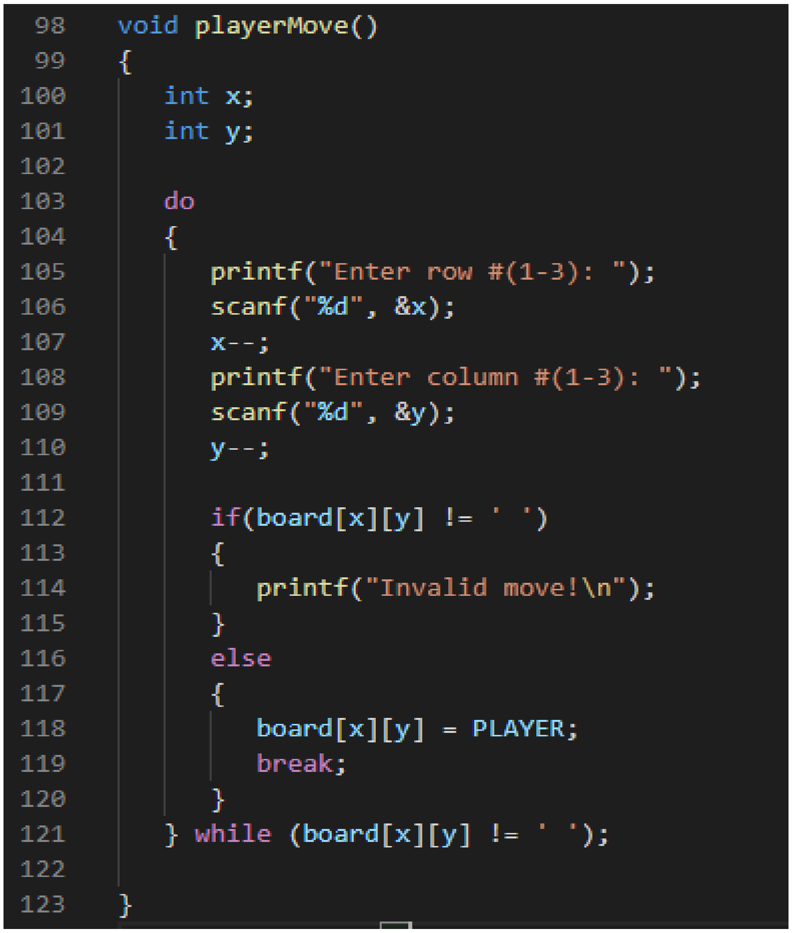
```
int checkFreeSpaces()
{
    int freeSpaces = 9;

    for(int i = 0; i < 3; i++)
    {
        for(int j = 0; j < 3; j++)
        {
            if(board[i][j] != ' ')
            {
                freeSpaces--;
            }
        }
    }
    return freeSpaces;
}
```

This function is used to check for given free spaces so that if a pre-filled space is being assigned with some value then it will give invalid move.
This has been done by counting the number of elements of the board[][] array that have an ‘ ‘ value.

As before, a nested for loop is used to access the elements of the 2D array. And if an element contains a blank space, the variable freeSpaces is decremented.

This is an integer function and the integer freeSpaces is returned.

void playerMove()-



```c
void playerMove()
{
    int x;
    int y;

    do
    {
        printf("Enter row #(1-3): ");
        scanf("%d", &x);
        x--;
        printf("Enter column #(1-3): ");
        scanf("%d", &y);
        y--;

        if(board[x][y] != ' ')
        {
            printf("Invalid move!\n");
        }
        else
        {
            board[x][y] = PLAYER;
            break;
        }
    } while (board[x][y] != ' ');

}
```

This function is used for the player's moves against the computer.

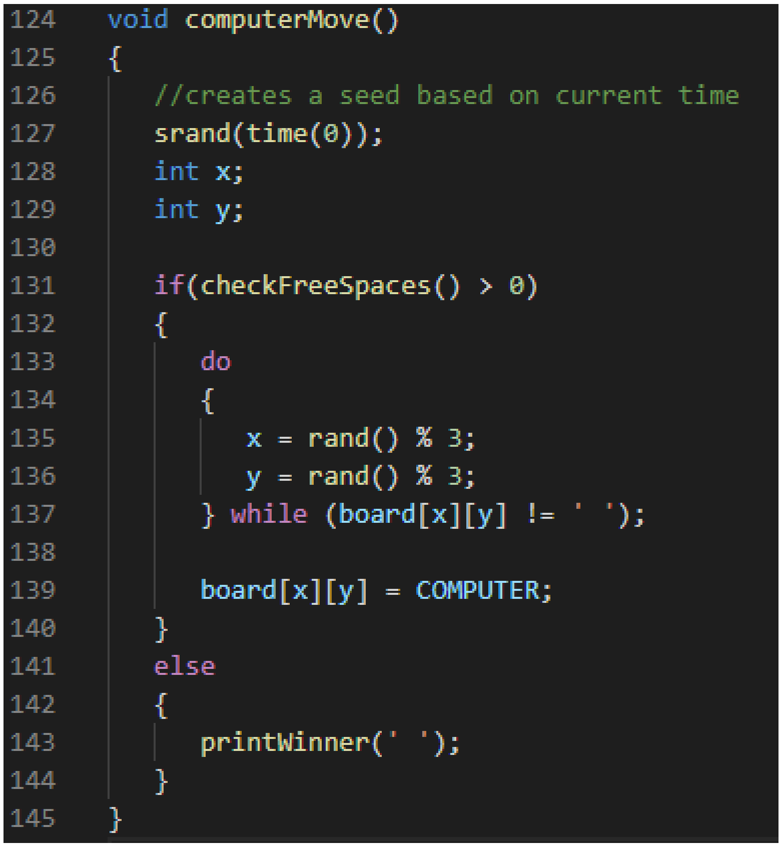
The user is asked to enter x and y values (the coordinates) in which they want to place their cross.

Then an if loop is used to test whether the user given position is free by checking whether board[x][y] has a blank space.
If the position has a blank space then board[x][y] will be assigned to the char PLAYER ('X')

If the position does not have a blank space then "invalid move" gets printed and the user will get to enter a valid position.
This has been facilitated by the do while loop condition

This is a void function and does not return any values.

void computerMove()-



```
void computerMove()
{
    //creates a seed based on current time
    srand(time(0));
    int x;
    int y;

    if(checkFreeSpaces() > 0)
    {
        do
        {
            x = rand() % 3;
            y = rand() % 3;
        } while (board[x][y] != ' ');

        board[x][y] = COMPUTER;
    }
    else
    {
        printWinner(' ');
    }
}
```

This function is used for the moves of the computer.

srand()- The srand() function sets the starting point for producing a series of pseudo-random integers.

time()-The time() function is defined in time.h (ctime in C++) header file. This function returns the time since 00:00:00 UTC, January 1, 1970 (Unix timestamp) in seconds.

rand()- The rand() function is used in C/C++ to generate random numbers in the range [0, RAND_MAX).

srand(time(0)) is used in the generation of random numbers.

An if condition is used to check for free spaces on the board using the checkFreeSpaces() function.

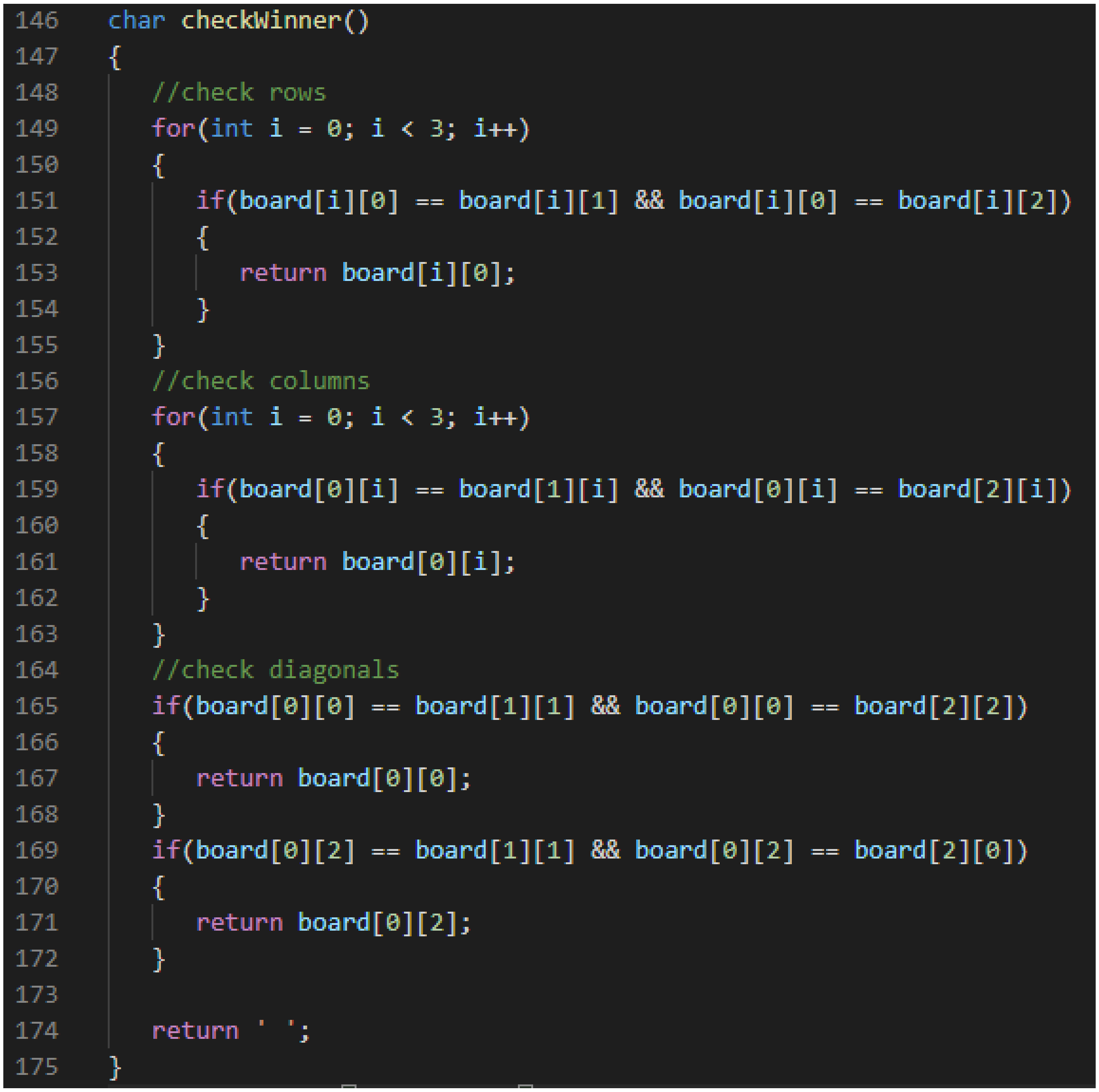
Then x and y are assigned random values ranging between 0-2, using rand() % 3.

A do while loop is given to repeat the block if the position at board[x][y] is not free.
If it is free then it gets assigned to the COMPUTER value ‘O’.

Also if the first if condition is not satisfied then the printWinner(‘ ‘) function is called.

This is a void function and does not return any value.

char checkWinner()-



```
char checkWinner()
{
    //check rows
    for(int i = 0; i < 3; i++)
    {
        if(board[i][0] == board[i][1] && board[i][0] == board[i][2])
        {
            return board[i][0];
        }
    }
    //check columns
    for(int i = 0; i < 3; i++)
    {
        if(board[0][i] == board[1][i] && board[0][i] == board[2][i])
        {
            return board[0][i];
        }
    }
    //check diagonals
    if(board[0][0] == board[1][1] && board[0][0] == board[2][2])
    {
        return board[0][0];
    }
    if(board[0][2] == board[1][1] && board[0][2] == board[2][0])
    {
        return board[0][2];
    }

    return ' ';
}
```

This function is used to compare the rows, columns and diagonals so as to check whether it forms a straight line of Xs or Os.

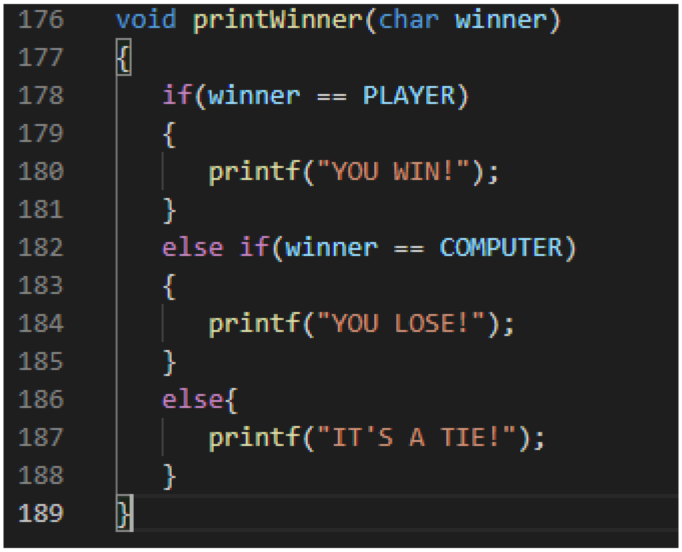
Checking happens in three categories: rows,columns and diagonals. Each one of them is tested using for and if control statements.

This is a character function and will return a char.

If any of the rows, columns or diagonals contain the same character then that character gets returned.

Else ‘ ‘ will get returned.

void printWinner()-



```
void printWinner(char winner)
{
    if(winner == PLAYER)
    {
        printf("YOU WIN!");
    }
    else if(winner == COMPUTER)
    {
        printf("YOU LOSE!");
    }
    else{
        printf("IT'S A TIE!");
    }
}
```

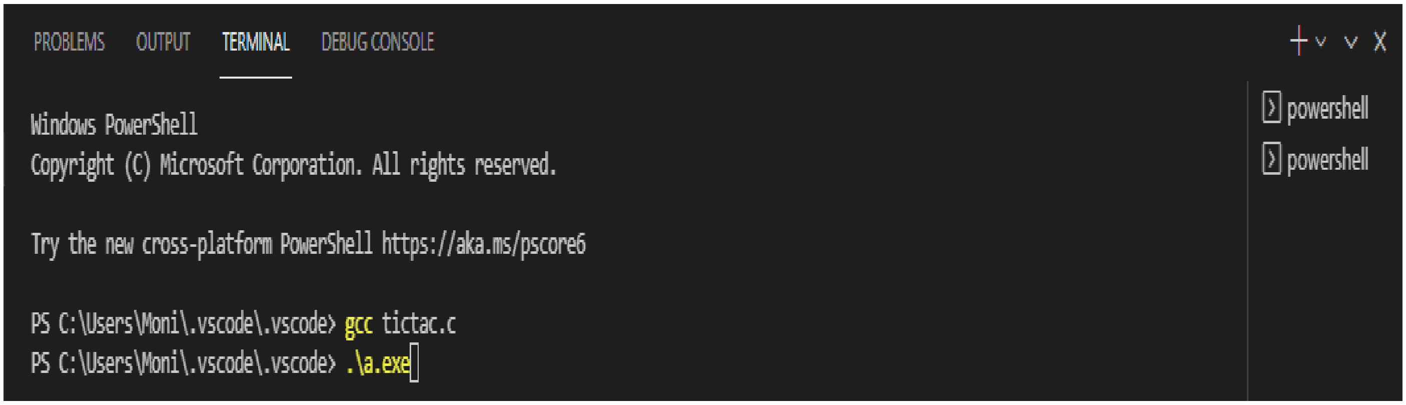
This function is used to print whether we have won, lost, or the match has tied.
If conditions are used to check who the winner is.

This a void function so it has no return value .

char winner is the parameter used here.

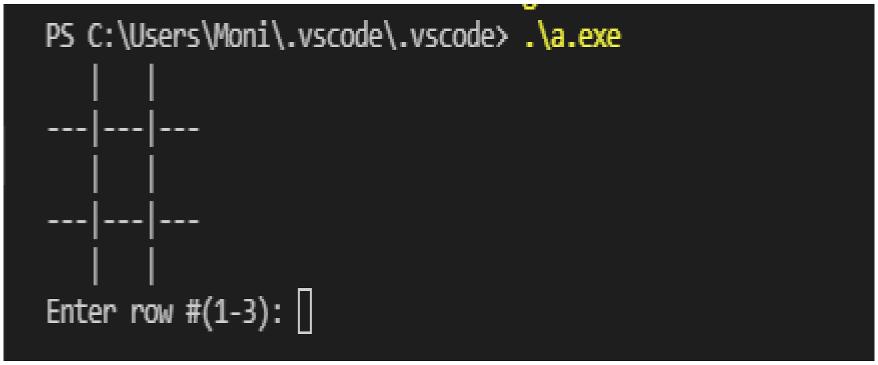
OUTPUT

For executing the program, given commands need to be entered in the terminal.



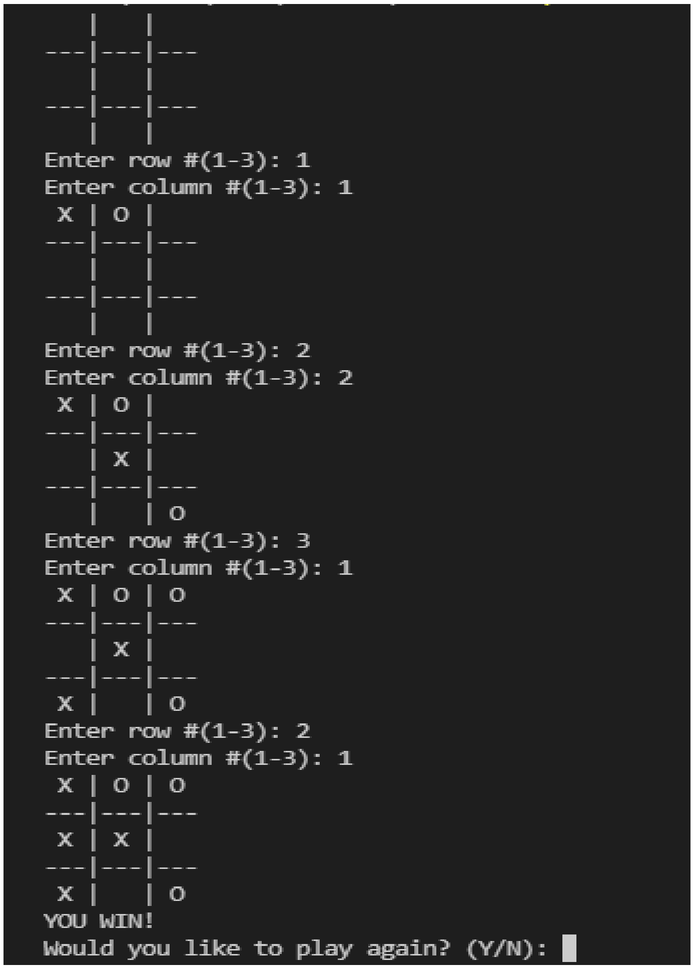
```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\Moni\.vscode\.vscode> gcc tictac.c
PS C:\Users\Moni\.vscode\.vscode> .\a.exe
```

After the execution of the commands, the terminal will look like this.



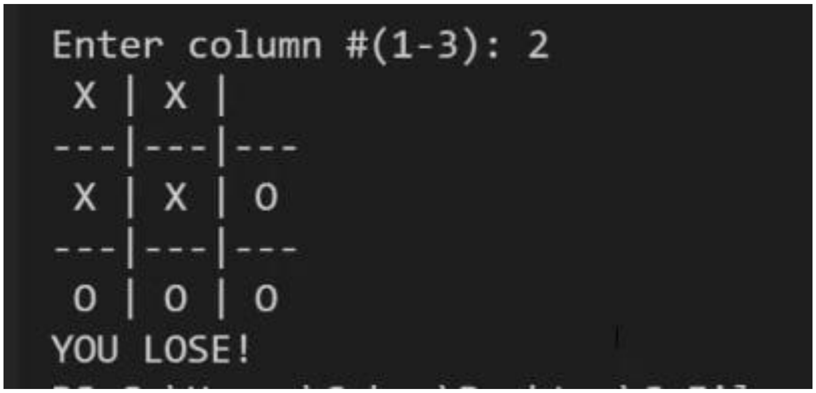
```
PS C:\Users\Moni\.vscode\.vscode> .\a.exe
   |   |
---|---|---
   |   |
---|---|---
   |   |
Enter row #(1-3):
```

Case 1- If we win.



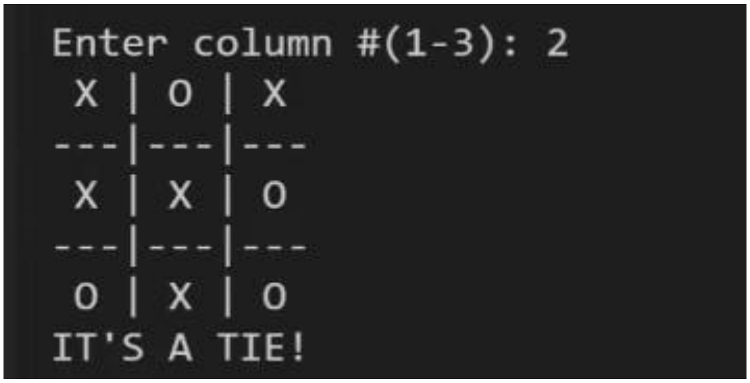
```
   |   |
---|---|---
   |   |
---|---|---
   |   |
Enter row #(1-3): 1
Enter column #(1-3): 1
 X | O |
---|---|---
   |   |
---|---|---
   |   |
Enter row #(1-3): 2
Enter column #(1-3): 2
 X | O |
---|---|---
   | X |
---|---|---
   |   | O
Enter row #(1-3): 3
Enter column #(1-3): 1
 X | O | O
---|---|---
   | X |
---|---|---
 X |   | O
Enter row #(1-3): 2
Enter column #(1-3): 1
 X | O | O
---|---|---
 X | X |
---|---|---
 X |   | O
YOU WIN!
Would you like to play again? (Y/N):
```

Case 2- If we lose.



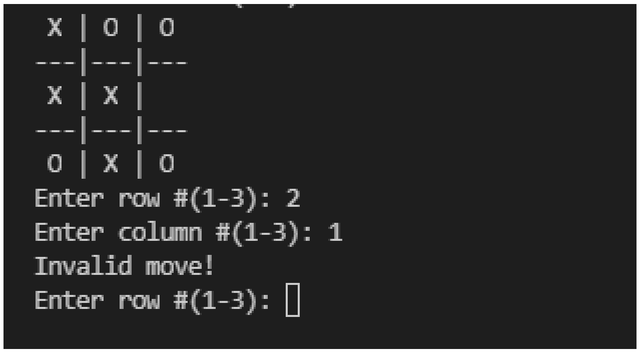
```
Enter column #(1-3): 2
 X | X |
---|---|---
 X | X | O
---|---|---
 O | O | O
YOU LOSE!
```

Case 3- If the match gets tied.



```
Enter column #(1-3): 2
 X | O | X
---|---|---
 X | X | O
---|---|---
 O | X | O
IT'S A TIE!
```

In case of invalid entry-



```
 X | O | O
---|---|---
 X | X |
---|---|---
 O | X | O
Enter row #(1-3): 2
Enter column #(1-3): 1
Invalid move!
Enter row #(1-3):
```

CONCLUSION

The Tic Tac Toe In C Programming With Source Code is developed using C programming language. This Tic Tac Toe in C game is played on a 3×3 grid by two players.

A tic tac toe game c program first player uses a circle to mark his movements, while the second uses a cross. The tic tac toe using c player who has made a three-mark horizontal, vertical, or diagonal series wins.

In this tic tac toe on c, both players must enter a particular number one to nine based on the grid position in order to make a mark X or O. The tic tac toe program in c 3×3 winner will be the first player to successfully position three of their marks in a horizontal, vertical, or diagonal) row.

REFERENCES

https://www.youtube.com/watch?v=_889aB2D1KI

https://www.geeksforgeeks.org/

CERTIFICATE

This is to certify that “Prachi”, “Ch. Malvika Reddy” and “Anuradha Krishnan”, students of department Computer Science and Engineering with specialization in Artificial Intelligence and Machine Learning section T1 (Batch-1) has successfully completed their Programming and Problem Solving Project on “Tic-Tac-Toe Game” under the guidance of their professor “Dr. M Sindhuja”.

________________ ________________

Professor’s Signature (Prachi (RA2111026010111))

____

(Ch. Malvika Reddy (RA2111026010125))

____

(Anuradha Krishnan (RA2111026010134))