**Introductory Programming UESTC1005**

**STUDENT NAME**: 张立澄

---

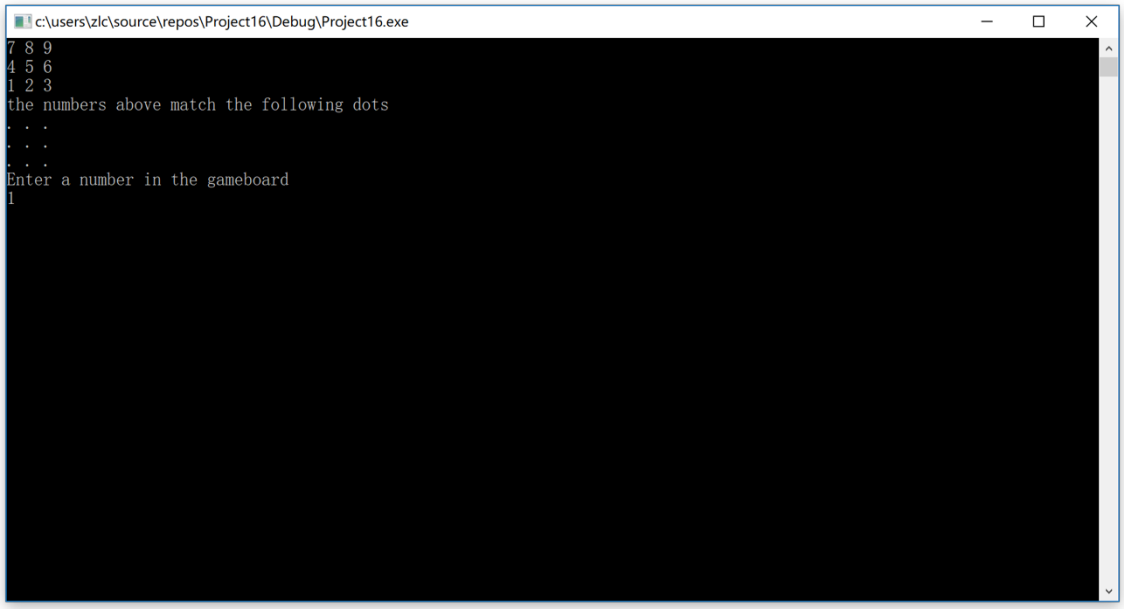
**STUDENT NUMBER**: 2017200602011

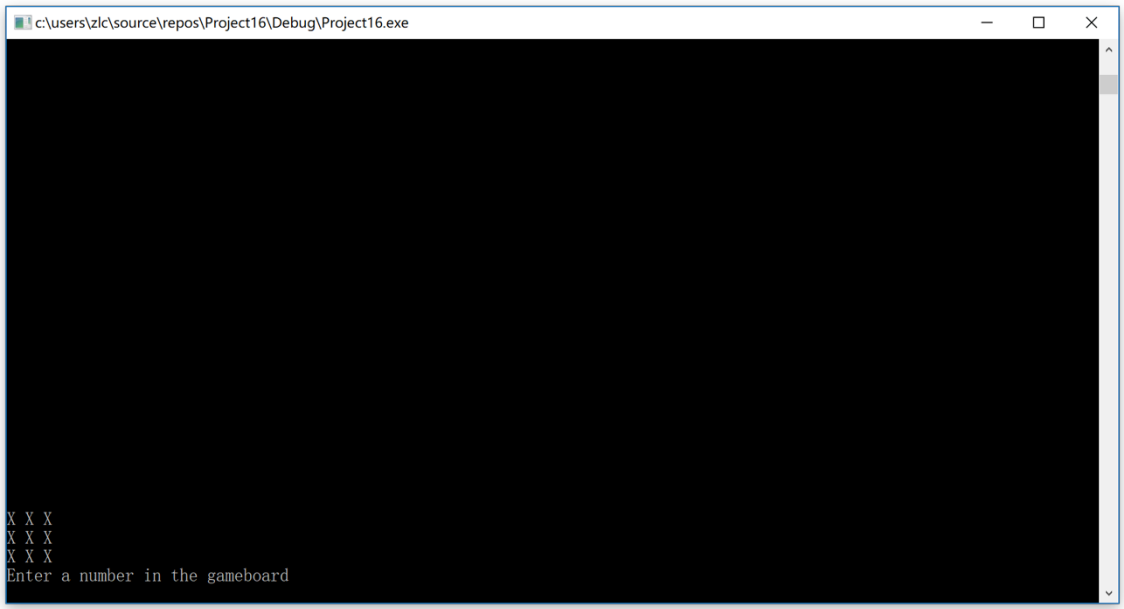
---

You will need to complete the demonstration of this week lab and submit this report into BB （it will be added into your portfolio of work).

## Exercise 10: Two-Dimensional Array

Task_A: Follow the instructions in Week 11 Lab manual and past your screen short of your result here.





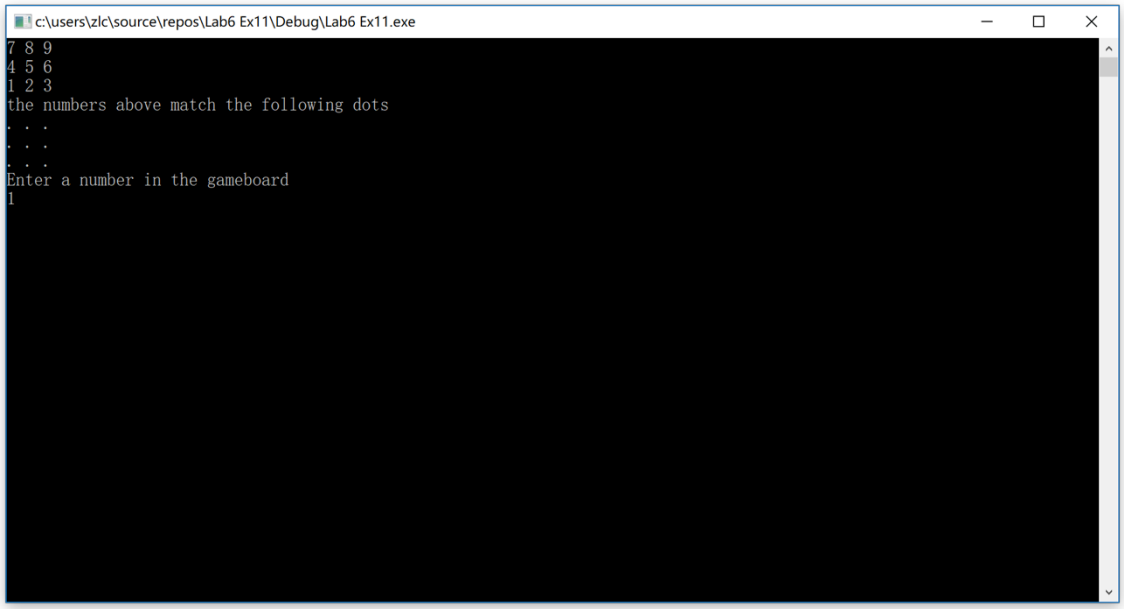

Task_B: Please attach your source code here.

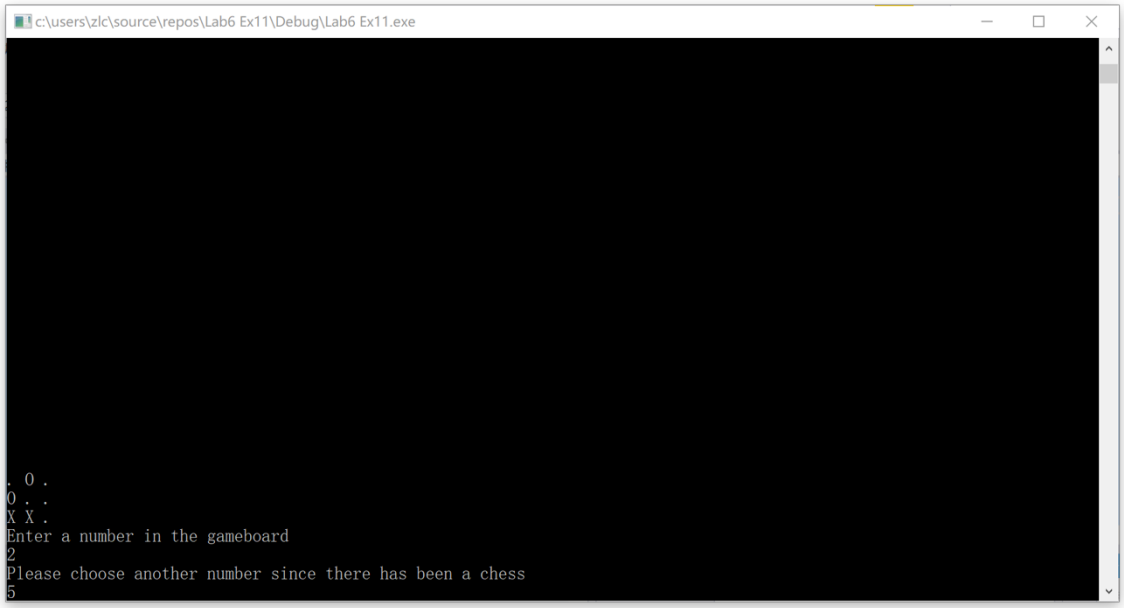
```
//This is a Tic-Tac-Toe
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

void Gameboard(int row, int column, char square[3][3], char c)
{
	square row  column     c
	for (int i = 0; i<3; i++)
	{
		for (int j = 0; j<3; j++)
		{
			printf("%c ", square[i][j]);
		}
		printf("\n");
	}
}

int	int n //locate the row
{
	return 2 - (n - 1) / 3;
}

int		int n //locate the column
{
	return (n - 1) % 3;
}

int main()
{

	char square[3][3];
	int chess[3][3];
	int Human_number, Human_row, Human_column;
```

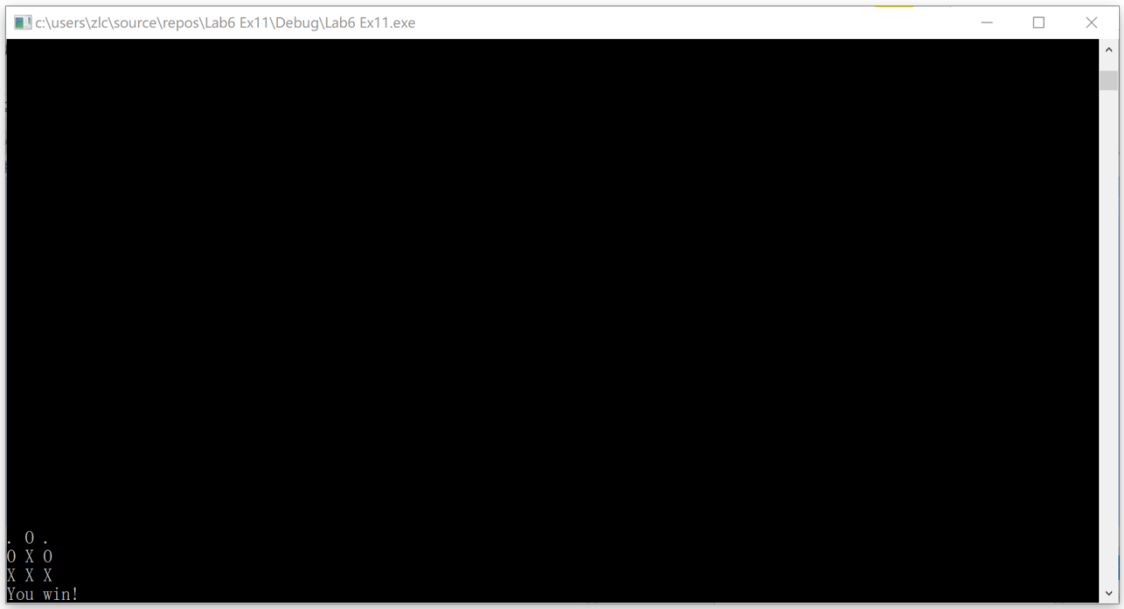
```
        printf("7 8 9\n");
        printf("4 5 6\n");
                "1 2 3\n"  //using the third gameboard
        for  int                        //get the 3*3 gameboard
        {
                for (int j = 0; j < 3; j++)//get a row with 3 "."
                {
                        square[i][j] = '.';
                        chess[i][j] = 0;
                }
        }
               "the numbers above match the following dots\n"
        for  int                        //get three rows and each of them with 3 "."
        {
                for (int j = 0; j < 3; j++)
                {
                        printf("%c ", square[i][j]);
                }
                printf("\n");
        }

        int n = 1;
        char x = 'X', o = 'O';
        while (n <= 9) {
                        "Enter a number in the gameboard\n"
                scanf_s("%d", &Human_number);
                getchar();
                Human_row = locate_row(Human_number);
                Human_column = locate_column(Human_number);
                while (chess[Human_row][Human_column] != 0)
                {
                                "Please choose another number since there has been a chess\n"
                        scanf_s("%d", &Human_number);
                        Human_row = locate_row(Human_number);
                        Human_column = locate_column(Human_number);
                   //warning for repeat

                chess[Human_row][Human_column] = 1;//locate the chess
                        "\n\n\n\n\n\n\n\n\n\n"
                        "\n\n\n\n\n\n\n\n\n\n"
                        "\n\n\n\n\n\n\n\n\n\n"
                Gameboard(Human_row, Human_column, square, x);
        }
}
```

## Exercise 11: Tic-Tac-Toe

Task_A: Follow the instructions in Week 11 Lab manual and past your screen short of your result here.






Task_B: Please attach your source code here.

```
//This is a Tic-Tac-Toe
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

void Gameboard(int row, int column, char square[3][3], char c)
{
	square row  column     c
	for (int i = 0; i<3; i++)
	{
		for (int j = 0; j<3; j++)
		{
			printf("%c ", square[i][j]);
		}
		printf("\n");
	}
}

int check_horizontal(int row, int chess[3][3])//check horizontally
{
	if (chess[row][0] == chess[row][1] && chess[row][0] == chess[row][2] && chess[row][0] != 0)
		return chess[row][0];
	else return
}
int check_vertical(int column, int chess[3][3])//check vertically
{
	if  chess    column      chess    column      chess    column      chess    column      chess    column
		return chess    column
	else return
}
int check_diagonal(int chess[3][3])//check diagonally
{
	if (chess[0][0] == chess[1][1] && chess[0][0] == chess[2][2] && chess[0][0] != 0)
		return chess[0][0];
	else if (chess[0][2] == chess[1][1] && chess[0][2] == chess[2][0] && chess[0][2] != 0)
		return chess[0][2];
	else return
}

int	int chess	//check three situations above
{
	for (int i = 0; i<3; i++)
```

```
        {
                if (check_horizontal(i, chess) != 0)
                        return check_horizontal(i, chess);
                else if (check_vertical(i, chess) != 0)
                        return check_vertical(i, chess);
        }
        if (check_diagonal(chess) != 0)
                return check_diagonal(chess);
        else return
}

int     int n //locate the row
{
        return 2 - (n - 1) / 3;
}

int             int n //locate the column
{
        return (n - 1) % 3;
}

float getRand() {
        return rand() / (RAND_MAX + 1.0);
 //get a random number for computer to play a chess

int main()
{
        srand(time(NULL));
        char square[3][3];
        int chess[3][3];
        int Human_number, Human_row, Human_column;
        int Computer_number, Computer_row, Computer_column;
        printf("7 8 9\n");
        printf("4 5 6\n");
                "1 2 3\n"  //using the third gameboard
        for  int                //get the 3*3 gameboard
        {
                for (int j = 0; j < 3; j++)//get a row with 3 "."
                {
                        square[i][j] = '.';
                        chess[i][j] = 0;
                }
        }
                "the numbers above match the following dots\n"
        for  int                //get three rows and each of them with 3 "."
        {
                for (int j = 0; j < 3; j++)
                {
                        printf("%c ", square[i][j]);
                }
                printf("\n");
        }

        int n = 1;
        char x = 'X', o = 'O';
        while (n <= 9) {
                        "Enter a number in the gameboard\n"
                scanf_s("%d", &Human_number);
                getchar();
                Human_row = locate_row(Human_number);
                Human_column = locate_column(Human_number);
                while (chess[Human_row][Human_column] != 0)
                {
                                "Please choose another number since there has been a chess\n"
                        scanf_s("%d", &Human_number);
                        Human_row = locate_row(Human_number);
                        Human_column = locate_column(Human_number);
                   //warning for repeat

                chess[Human_row][Human_column] = 1;//locate the chess
                        "\n\n\n\n\n\n\n\n\n\n"
                        "\n\n\n\n\n\n\n\n\n\n"
                        "\n\n\n\n\n\n\n\n\n\n"
                Gameboard(Human_row, Human_column, square, x);

                if (check(chess) == 1) //win
                {
```

```
                                printf("You win!");
                                getchar();
                                getchar();
                                return 0;
                        }
                        Computer_number = getRand() * 9 + 1;
                        Computer_row = locate_row(Computer_number);
                        Computer_column = locate_column(Computer_number);

                        while                                                   //Let computer play chess randomly in the loop
                        {
                                Computer_number = getRand() * 9 + 1;
                                Computer_row = locate_row(Computer_number);
                                Computer_column = locate_column(Computer_number);
                        }

                        chess[Computer_row][Computer_column] = 2;
                                "\n\n\n\n\n\n\n\n\n\n"
                                "\n\n\n\n\n\n\n\n\n\n"
                                "\n\n\n\n\n\n\n\n\n\n"
                        Gameboard(Computer_row, Computer_column, square, o);

                        if (check(chess) == 2)
                        {
                                printf_s("You lose");
                                getchar();
                                return 0;
                        }
                }
}
```