

VASAVI COLLEGE OF ENGINEERING
(AUTONOMOUS)

(Affiliated to Osmania University)

Hyderabad-500031

DEPARTMENT OF : INFORMATION TECHNOLOGY

PROGRAMMING FOR
PROBLEM SOLVING

(ASSIGNMENT -3)
TANT FANT GAME

DONE BY:

1.Akshitha Doodala

2.K Sai Shruthi

HT NO: 1602-20-737-003

1602-20-737-036

VASAVI COLLEGE OF ENGINEERING

(AUTONOMOUS)

(Affiliated to Osmania University)

Hyderabad-500031

DEPARTMENT OF: INFORMATION TECHNOLOGY

NAME OF THE LABORATORY: PROGRAMMING FOR PROBLEM SOLVING

TANT FANT GAME

1. INTRODUCTION TO THE PROJECT:

In the [Tant Fant Game](#) code we have used loops, conditional statements, functions, arrays, string, pointers, structures and files.

TANT FANT is a two-player abstract strategy game from India. It is related to tic-tac-toe, but more closely related to three men's morris, Nine Holes, Achi, shisima, and Dara, because pieces are moved on the board to create the 3 in-a-row. It is an alignment game.

RULES FOR THE GAME:

1. Each player will be getting three tokens of their selected symbol respectively.
2. Player 2 cannot choose the token selected by Player1. Both players should have different token symbols.
3. During first three turns of each players, they need to mark 3 tokens on different given positions of the board respectively.
4. At the time of first three turns of each player, no player should overlap the other players token position.
5. After playing first three turns players can move their respective token one space at a time following the pattern on the board.
6. Only one token can be moved per turn and you need to choose your token only during your turn.
7. Player can only move their token to the selected position, if there is a vacant space available at that position.
8. If the player's tokens are 3 in a row either horizontally, vertically or diagonally then that player will win the match.
9. Once the game is started, Player can only exit if the match is over.

2. DESCRIPTION ABOUT THE FUNCTIONS USED IN THE PROJECT :

USER DEFINED FUNCTIONS:

Although graphics has not been used in this project, the application of user defined functions have been effectively used here. The code for Tant Fant contains nine(9) major user defined functions. Description of the following functions are named and discussed below:

- `void token_change(char *,char *,char [],char []);`
- `void rules();`
- `void game();`
- `void display_board();`
- `void marking_tokens (int ,char ,int);`
- `int moving_positions (int ,int *,int *);`
- `int moving_condition (int ,int ,int);`
- `void moving_tokens (int ,int ,int);`
- `int check_win();`
- `void coders_name();`

i. void game():

This is a void function without passing parameters. The function `game()` takes the input from the user about the player's name and player's choice of token. It further calls all the major user defined functions. It also displays the winner of the game.

ii. void token_change(char *, char *, char [], char []);

This is a void function with passing parameters of char type with help of pointers and arrays. The function `token_change` checks if the token selected by the player is already selected or not.

iii. void rules():

This is a void function without passing parameters. The function `rules()` displays rules of the game 'TANT FANT'.

iv. void display_board():

This is a void function without passing parameters. The function `display_board()` displays the board with certain numbers required for the game.

v. void marking_tokens (int ,char ,int) :

This is a void function with passing parameters for int and char type. The function `void marking_tokens(int ,char ,int)` allows the player to mark their token in the given set of choices and some test case have been added.

vi. int moving_positions(int ,int * ,int *)

This is an integer function with passing parameters of integer type with the help of pointers. The function moving_positions(int, int *,int *) asks the player to enter the positions of token i.e. from where to where they would like to move their tokens and returns a values of “from” and “to”.

vii. int moving_condition (int ,int ,int):

This is an integer function with passing parameters of integer type. The function moving_condition() helps in checking all the conditions required to play the game without any discrepancy and returns a value.

viii. void moving_tokens (int ,int ,int) :

This is a void function with passing parameters of int type. The function moving_tokens (int, int ,int) helps the player to move their token only once in the given set of “from” and “to” positions.

ix. int check_win();

This is an integer function without passing parameters. The function check_win() checks the winning conditions of the game and returns a value.

PRE-DEFINED STANDARD LIBRARY FUNCTIONS :

Standard library functions are also known as built-in functions. Functions such as printf(), scanf() etc are standard library functions. These functions are already defined in header files (files with .h extensions are called header files as <stdio.h>, <stdlib.h>)so we just call them whenever there is a need to use them.

3. SOURCE CODE :

Program:

The source code has been attached. (NOTEPAD)

In putty server:

```
[ it20003, it20036 ]
```

```
$ cd GAME
```

```
$ vi tantfant.c
```

Compilation :

```
$ gcc tantfant.c -o tantfant.o
```

```
$ ./tantfant.o
```

Output:

```
[it20036@vasavi ~]$ cd GAME
[it20036@vasavi GAME]$ clear
[it20036@vasavi GAME]$ ./tantfant.o

      *-----*
      |               |
      |   CODE BY IT-A   |
      | AKSHITHA DOODALA (1602-20-737-003) |
      | K SAI SHRUTHI(1602-20-737-036)    |
      |               |
      *-----*

*****TANT FANT*****
*
*   TANT FANT is a two-player abstract strategy game from India. It is related to tic-tac-toe, but
*   more closely related to three men's morris, Nine Holes, Achi, shisima, and Dara, because pieces are
*   moved on the board to create the 3 in-a-row. It is an alignment game.
*
*
*   RULES FOR THE GAME:
*
*   1. Each player will be getting three tokens of their selected symbol respectively.
*   2. During first three turns of each players, they need to mark 3 tokens on different given positions of the board respectively.
*   3. At the time of first three turns of each player, no player should overlap the other players token position.
*   4. After playing first three turns players can move their respective token one space at a time following the pattern on the board.
*   5. Only one token can be moved per turn and you need to choose your token only during your turn.
*   6. Player can only move their token to the selected position, if there is a vacant space available at that position.
*   7. If the player's tokens are 3 in a row either horizontally, vertically or diagonally then that player will win the match.
*   8. Once the game is started, Player can only exit if the match is Won
*
*****
Enter Player 1's name: Akshitha
Akshitha,Choose any character token : A
Enter Player 2's name: Shruthi
Shruthi,Choose any character token : S
ALL THE BEST Akshitha and Shruthi
      GAME STARTS NOW!!

      1      2      3
```

```

Enter Player 1's name: Akshitha
Akshitha,Choose any character token : A
Enter Player 2's name: Shruthi
Shruthi,Choose any character token : S
ALL THE BEST Akshitha and Shruthi
      GAME STARTS NOW!!

```

1	2	3
*		*
	*	*
4	5	6
	*	*
		*
		*
7	8	9

```

Akshitha, Enter position of your token : 5

```

1	2	3
*		*
	*	*
4	A	6
	*	*
		*
		*
7	8	9

```

Shruthi, Enter position of your token : 6

```

1	2	3
*		*
	*	*
4	A	S
	*	*
		*
		*
7	8	9

Akshitha, Enter position of your token : 3

1	2	A
*		*
*		*
*		*
4	A	S
*		*
*		*
*		*
7	8	9

Shruthi, Enter position of your token : 7

1	2	A
*		*
*		*
*		*
4	A	S
*		*
*		*
*		*
S	8	9

Akshitha, Enter position of your token : 1

A	2	A
*		*
*		*
*		*
4	A	S
*		*
*		*
*		*
S	8	9

Shruthi, Enter position of your token : 2

A	S	A
*		*
	*	
		*
4	A	S
	*	*
		*
	*	
S	8	9

```
Akshitha,Enter from where to where you want to move your token
From & To : 5 8
```

A		S		A
*				*
	*			
		*		
			*	
4		5		S
		*	*	
	*		*	
S		A		9

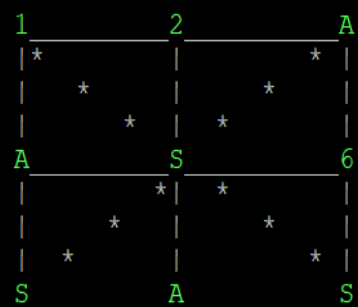
```
Shruthi,Enter from where to where you want to move your token
From & To : 2 5
```

A	2	A
*		*
	*	
	*	*
4	S	S
	*	*
	*	*
S	A	9

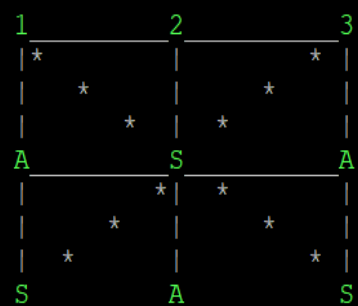
Akshitha,Enter from where to where you want to move your token
 From & To : 1 4



Shruthi,Enter from where to where you want to move your token
 From & To : 6 9



Akshitha,Enter from where to where you want to move your token
 From & To : 3 6



```

Shruthi,Enter from where to where you want to move your token
From & To : 5 7

INVALID!!!
The "To" position you have entered is occupied by your token or your opponet's.

Shruthi,Enter from where to where you want to move your token
From & To : 5 6

INVALID!!!
The "To" position you have entered is occupied by your token or your opponet's.

Shruthi,Enter from where to where you want to move your token
From & To : 1
11

INVALID!!!
You cannot move from vacant space
Maybe
1. The "To" position you have entered is invalid.
2. The "To" position you have entered is occupied by your token or by your opponet's.

Shruthi,Enter from where to where you want to move your token
From & To : 5 1

```

S	2	3
*		*
*		*
*		
A	5	A
*		*
*		*
*		*
S	A	S

```

Akshitha,Enter from where to where you want to move your token
From & To : 8 5

```


S	2	3
*		*
*		*
*		
A	A	A
*		*
*		*
*		*
S	8	S

```


Akshitha won this match
Congragulations🎉🎉🎉🎉
Hope you enjoyed playing this game
Want to play again??
Then enter 1.
If wanted to exit then enter 5.
5
[it20036@vasavi GAME]$ █

```

FILE STORING THE WINNERS DATA:

 it20036@vasavi:~/GAME

```
[it20036@vasavi ~]$ cd GAME
[it20036@vasavi GAME]$ clear
[it20036@vasavi GAME]$ ls
tantfant.c  tantfant.o  winners.txt
[it20036@vasavi GAME]$ vi winners.txt
```

 it20036@vasavi:~/GAME

```
Akshitha has won the match
Shruthi has won the match
```

~~~~~

```
"winners.txt" 2L, 53C
```

#### 4. TEST CASES :

- a. To ensure that player2 doesn't selected the same token selected by player1.

```
*****
Enter Player 1's name: Akki
Akki,Choose any character token : A
Enter Player 2's name: Shru
Shru,Choose any character token : A
CHOOSE ANOTHER TOKEN.
The token you have selected is taken by Akki.
Shru,Choose any character token : A
CHOOSE ANOTHER TOKEN.
The token you have selected is taken by Akki.
Shru,Choose any character token : s
ALL THE BEST Akki and Shru
      GAME STARTS NOW!!
```

|   |   |   |
|---|---|---|
| 1 | 2 | 3 |
| * |   | * |
|   |   |   |
|   |   |   |
| 4 | 5 | 6 |
|   |   |   |
|   |   |   |
|   |   |   |
| 7 | 8 | 9 |

```
Akki, Enter position of your token : █
```

b. **Marking Positions :**

- To ensure the choice of the token is not marked in an occupied position.
- To ensure the given choice of number is between the range  $1 \leq \text{choice} \leq 9$ .

|   |   |   |
|---|---|---|
| 1 | 2 | 3 |
| * |   | * |
|   |   |   |
|   |   |   |
| 4 | A | 6 |
|   |   |   |
|   |   |   |
|   |   |   |
| 7 | 8 | 9 |

```
Shru, Enter position of your token : 5

INVALID!!!
No vacant position available in your selected position.
Shru, Enter position of your token : 11

INVALID!!!!
Enter Valid position from 1-9.
Shru, Enter position of your token : █
```

c. To ensure the given “from” position is occupied with the respective player token and “to” position is vacant.

```

From & To : 5 5

INVALID!!!
Maybe
1. You have entered same position for both "From" and "To"
2. The "From" and "To" position you have entered is occupied by your token or your opponet's.

Akki,Enter from where to where you want to move your token
From & To : 1 3

INVALID!!!
Maybe
1. Both the "From" and "To" positions are vacant.
2. You made a double move or wrong pattern move.n
Akki,Enter from where to where you want to move your token
From & To : 1 1

INVALID!!!
The entered positions of "From" and "To" are same and vacant.

Akki,Enter from where to where you want to move your token
From & To : 6 5

INVALID!!!
Maybe
1. The "To" position you have entered is occupied by your token or your opponet's.
2. You cannot move other player token.
3. The "To" position you have entered is not valid.

Akki,Enter from where to where you want to move your token
From & To : 8 1

INVALID!!!
Your Token cannot move to the given "To" position as it can only take one move.

Akki,Enter from where to where you want to move your token
From & To : 5 5

INVALID!!!
Maybe
1. You have entered same position for both "From" and "To"
2. The "From" and "To" position you have entered is occupied by your token or your opponet's.

Akki,Enter from where to where you want to move your token

```

d. To ensure player can exit or play the game after completion of one game.

```

Shru won this match
Congragulations🎉🎉🎉🎉
Hope you enjoyed playing this game
Want to play again??
Then enter 1.
If wanted to exit then enter 5.
1
Enter Player 1's name: akki
akki,Choose any character token : A
Enter Player 2's name: Shru
Shru,Choose any character token : S
ALL THE BEST akki and Shru
      GAME STARTS NOW!!

1 2 3
|*| |*| |
| |*| |*|
| |*| |*|
4 5 6
|*| |*| |
| |*| |*|
| |*| |*|
7 8 9

akki, Enter position of your token : █

```

```
Shru won this match
Congragulations🎉🎉🎉🎉
Hope you enjoyed playing this game
Want to play again??
Then enter 1.
If wanted to exit then enter 5.
-1
Enter Valid number.
5
[it20003@vasavi ~]$
```

## 5. WHAT HAVE WE LEARNT DOING THIS PROJECT ?

Coding for '**TANT FANT GAME**' was a challenging at the same time it is an another great way to improve our coding skills. This project with a lot of variables and characters has thought us how to look for test cases of different types. And sometimes its difficult to crack the test cases yet we have managed to check them successfully.

Creating our own game gives an hands on experience in manipulating variants and input/output and more.

We have learnt to use colours to the source code using few printf statements as given below in our source code.

- *BIRed = "\033[1;91m\" # Red*
- *BIGreen = "\033[1;92m\" # Green*
- *BIYellow = "\033[1;93m\" # Yellow*
- *BIBlue = "\033[1;94m\" # Blue*
- *BIPurple = "\033[1;95m\" # Purple*
- *BICyan = "\033[1;96m\" # Cyan*
- *BIWhite = "\033[1;97m\" # White*

This process has also made our concepts for loops, condition statements, functions, arrays, strings, pointers, structures and files more precise and clear. We also have learnt to work in a group, sharing equal parts of the work. On a whole this project has been both exciting and challenging.

## 6. CONCLUSION :

The C program "TANT FANT" is compiled without any errors and completed successfully.

## 7. FUTURE SCOPE :

The code of this game can be improved using more advanced topics like graphic card and use of more functions to make our code look easier.