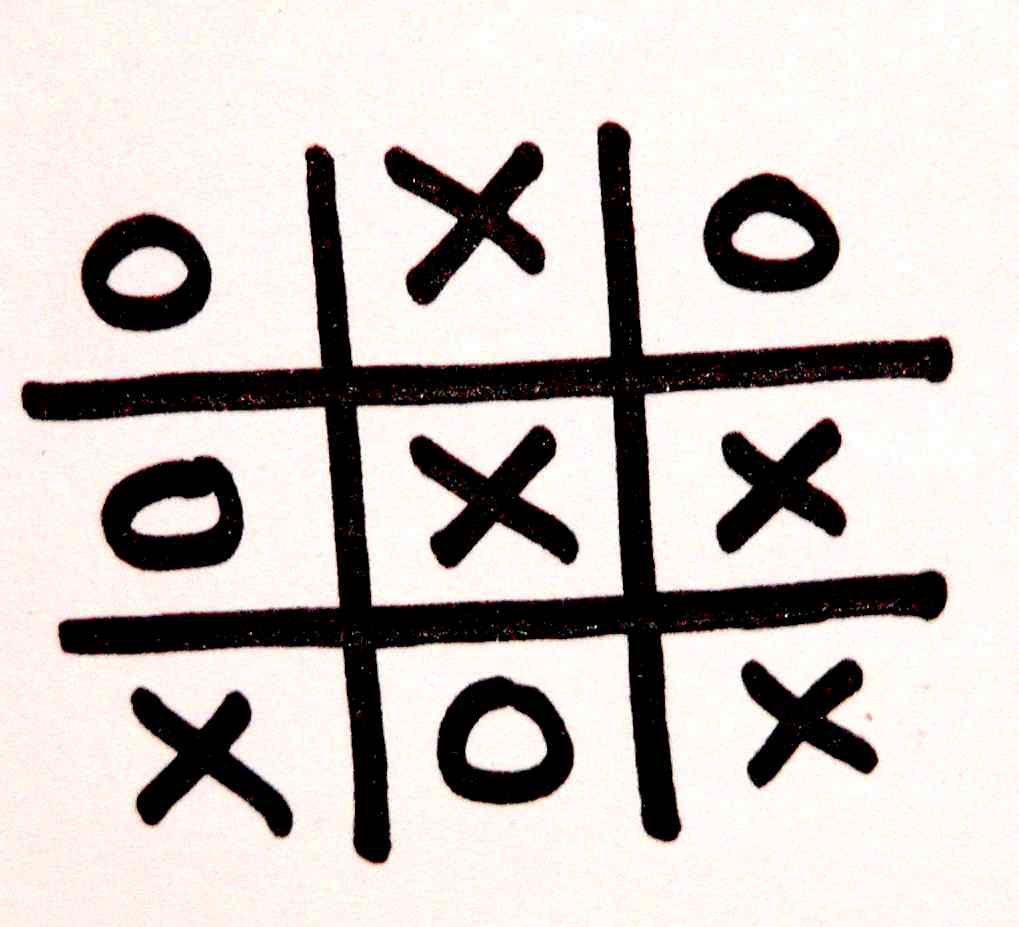
**1. TIC-TAC-TOE**



*Tic-tac-toe is a* [*paper-and-pencil game*](https://en.wikipedia.org/wiki/Paper-and-pencil_game) *for two players, X and O, who take turns marking the spaces in a 3×3 grid. The player who succeeds in placing three of their marks in a horizontal, vertical, or diagonal row wins the game.*

The program for the game TIC TAC TOE is described in the program 1

Program1.: tic\_tac\_toe.c

/\* System Input is 1 and user input is 2 \*/

/\* We need two header files to include \*/

|  |
| --- |
| #include "/usr/include/stdio.h"  #include "/usr/include/stdlib.h" |

/\* Only main function contains all the operations for the computer as well as users \*/

/\* first of all we need to take the users choices which is described \*/

|  |
| --- |
| int main ()  {  fprintf (stderr, "\tSystem input is 1.\n\tUser input is 2.\n");  int choice;  fprintf (stderr, "\tWho wish to go first?\n");  fprintf (stderr, "\t1. System\n");  fprintf (stderr, "\t2. You\n");  fprintf (stderr, "\t---------\t");  scanf ("%d", &choice); |

/\* We need two dimensional array (3 X 3 matrix). Memory is allocated from heap segment \*/

|  |
| --- |
| int \*\* Boxes;  int i, j;    Boxes = NULL;  Boxes = (int \*\*) malloc (sizeof (int \*) \* 3);  if (Boxes == NULL)  {  perror ("malloc");  }  for (i = 0; i < 3; i++)  {  Boxes[i] = (int \*) malloc (sizeof (int) \* 3);  if (Boxes[i] == NULL)  {  perror ("malloc");  }  }  for (i = 0; i < 3; i++)  {  for (j = 0; j < 3; j++)  {  Boxes[i][j] = 0;  }  } |

/\* according to the choices if the user goes first then following action must be taken \*/

|  |
| --- |
| int count = 0;  if (choice == 2)  {  printf ("\tGame status is :\n");  for (i = 0; i < 3; i++)  {  for (j = 0; j < 3; j++)  {  printf ("\t%d\t", Boxes[i][j]);  if (j == 2)  {  printf ("\n");  }  }  }  again: fprintf (stdout, "\tEnter the box:\t");  scanf ("%d", &i);  scanf ("%d", &j);  if (i > 2 || j > 2)  {  printf ("\tInvalid input.\n");  goto again;  }  Boxes[i][j] = 2;  count += 1;  } |

/\* Now all works will be done in a while loop for the condition count, that is all boxes are filled then break the loop \*/

/\* Now it's computer chance \*/

/\* First of all computer will check that by a single move computer can win or not if it can win then give the input to the winning position \*/

|  |
| --- |
| while (count <= 9)  {  /\*Checking for Computer's Victory\*/  for (i = 0; i < 3; i++)  {  for (j = 0; j < 3; j++)  {  if (Boxes[i][j] == 0)  {  if (i == 0 && j == 0)  {  if ((Boxes[0][1] == 1 && Boxes[0][2] == 1) || (Boxes[1][1] == 1 && Boxes[2][2] == 1) ||(Boxes[1][0] == 1 && Boxes[2][0] == 1))  {  Boxes[i][j] = 1;  fprintf (stdout, "\tComputer won.\n");  goto free;  }    }  else if (i == 0 && j == 1)  {  if ((Boxes[0][0] == 1 && Boxes [0][2] == 1) || (Boxes[1][1] == 1 && Boxes[2][1] == 1))  {  Boxes[i][j] = 1;  fprintf (stdout, "\tComputer won.\n");  goto free;  }  }  else if (i == 0 && j == 2)  {  if ((Boxes[0][0] == 1 && Boxes [0][1] == 1) || (Boxes[1][1] == 1 && Boxes[2][0] == 1) || (Boxes[1][2] == 1 && Boxes[2][2] == 1))  {  Boxes[i][j] = 1;  fprintf (stdout, "\tComputer won.\n");  goto free;  }  }  else if (i == 1 && j == 0)  {  if ((Boxes[0][0] == 1 && Boxes[2][0] == 1) || (Boxes[1][1] == 1 && Boxes[1][2] == 1))  {  Boxes[i][j] = 1;  fprintf (stdout, "\tComputer Won.\n");  goto free;  }  }  else if (i == 1 && j == 1)  {  if ((Boxes[0][0] == 1 && Boxes[2][2] == 1) || (Boxes[0][1] == 1 && Boxes[2][1] == 1) || (Boxes[0][2] == 1 && Boxes[2][0] == 1) || (Boxes[1][0] == 1 && Boxes[1][2] == 1))  {  Boxes [i][j] = 1;  fprintf (stdout, "\tComputer Won.\n");  goto free;  }  }  else if (i == 1 && j == 2)  {  if ((Boxes[0][2] == 1 && Boxes[2][2] == 1) || (Boxes[1][0] == 1 && Boxes[1][1] == 1))  {  Boxes[i][j] = 1;  fprintf (stdout, "\tComputer Won.\n");  goto free;  }  }  else if (i == 2 && j == 0)  {  if ((Boxes[0][0] == 1 && Boxes[1][0] == 1) || (Boxes[0][2] == 1 && Boxes[1][1] == 1) || (Boxes[2][1] == 1 && Boxes[2][2] == 1))  {  Boxes[i][j] = 1;  fprintf (stdout, "\tComputer Won.\n");  goto free;  }  }  else if (i == 2 && j == 1)  {  if ((Boxes[2][0] == 1 && Boxes[2][2] == 1) || (Boxes[0][1] == 1 && Boxes[1][1] == 1))  {  Boxes[i][j] = 1;  fprintf (stdout, "\tComputer Won.\n");  goto free;  }  }  else if (i == 2 && j == 2)  {  if ((Boxes[0][0] == 1 && Boxes[1][1] == 1) || (Boxes[0][2] == 1 && Boxes[1][2] == 1) || (Boxes[2][0] == 1 && Boxes[2][1] == 1))  {  Boxes[i][j] = 1;  fprintf (stdout, "\tComputer Won.\n");  goto free;  }  }  }  }  } |

/\* If computer can not win in a single move as described above then check whether user may win in the next move or not. If the user is going to win then give the input to user's winning position, Also set the flag to 1 so that computer have one chance \*/

|  |
| --- |
| /\*Checking if user may win\*/  for (i = 0; i < 3; i ++)  {  for (j = 0; j < 3; j++)  {  if (Boxes[i][j] == 0)  {  if (i == 0 && j == 0)  {  if ((Boxes[0][1] == 2 && Boxes[0][2] == 2) || (Boxes[1][1] == 2 && Boxes[2][2] == 2) || (Boxes[1][0] == 2 && Boxes[2][0] == 2))  {  Boxes[i][j] = 1;  flag = 1;  count++;  break;  }  }  else if ( i == 0 && j == 1)  {  if ((Boxes[0][0] == 2 && Boxes[0][2] == 2) || (Boxes[1][1] == 2 && Boxes[2][1] == 2))  {  Boxes[i][j] = 1;  flag = 1;  count++;  break;  }  }  else if (i == 0 && j == 2)  {  if ((Boxes[0][0] == 2 && Boxes[0][1] == 2) || (Boxes[1][1] == 2 && Boxes[2][0] == 2) || (Boxes[1][2] == 2 && Boxes[2][2] == 2))  {  Boxes[i][j] = 1;  flag = 1;  count++;  break;  }  }  else if (i == 1 && j == 0)  {  if ((Boxes[0][0] == 2 && Boxes[2][0] == 2) || (Boxes[1][1] == 2 && Boxes[1][2] == 2))  {  Boxes[i][j] = 1;  flag = 1;  count++;  break;  }  }  else if (i == 1 && j == 1)  {  if ((Boxes[0][0] == 2 && Boxes[2][2] == 2) || (Boxes[0][1] == 2 && Boxes[2][1] == 2) || (Boxes[0][2] == 2 && Boxes[2][0] == 2) || (Boxes[1][0] == 2 && Boxes[1][2] == 2))  {  Boxes[i][j] = 1;  flag = 1;  count++;  break;  }  }  else if (i == 1 && j == 2)  {  if ((Boxes[0][2] == 2 && Boxes[2][2] == 2) || (Boxes[1][0] == 2 && Boxes[1][1] == 2))  {  Boxes[i][j] = 1;  flag = 1;  count++;  break;  }  }  else if (i == 2 && j == 0)  {  if ((Boxes[0][0] == 2 && Boxes[1][0] == 2) || (Boxes[1][1] == 2 && Boxes[0][2] == 2) || (Boxes[2][1] == 2 && Boxes[2][2] == 2))  {  Boxes[i][j] = 1;  flag = 1;  count++;  break;  }  }  else if (i == 2 && j == 1)  {  if ((Boxes[1][1] == 2 && Boxes[0][1] == 2) || (Boxes[2][0] == 2 && Boxes[2][2] == 2))  {  Boxes[i][j] = 1;  flag = 1;  count++;  break;  }  }  else if (i == 2 && j == 2)  {  if ((Boxes[2][0] == 2 && Boxes[2][1] == 2) || (Boxes[0][2] == 2 && Boxes[1][2] == 2) || (Boxes[1][1] == 2 && Boxes[0][0] == 2))  {  Boxes[i][j] = 1;  flag = 1;  count++;  break;  }  }  }  }  if (flag)  {  break;  }  }  if (count == 9)  {  break;  } |

/\* if flag is still 0 then check for the available boxes for the users in the current chance and according to the user's entry take decisions and make a suitable move \*/

/\* if it is first chance is for computer, that is count == 0 then \*/

|  |
| --- |
| if (count == 0)  {  Boxes[0][0] = 1;  count++;  } |

/\* if computer chance is second, then count == 1 and in this condition check 9 possible moves of user\*/

|  |
| --- |
| else if (count == 1)  {  if (Boxes[0][0] == 2)  {  Boxes[1][1] = 1;  count++;  }  else if (Boxes[0][1] == 2)  {  Boxes[0][0] = 1;  count++;  }  else if (Boxes[0][2] == 2)  {  Boxes[1][1] = 1;  count++;  }  else if (Boxes[1][0] == 2)  {  Boxes[0][2] = 1;  count++;  }  else if (Boxes[1][1] == 2)  {  Boxes[0][0] = 1;  count++;  }  else if (Boxes[1][2] == 2)  {  Boxes[0][0] = 1;  count++;  }  else if (Boxes[2][0] == 2)  {  Boxes[1][1] = 1;  count++;  }  else if (Boxes[2][1] == 2)  {  Boxes[0][0] = 1;  count++;  }  else if (Boxes[2][2] == 2)  {  Boxes[1][1] = 1;  count++;  }  } |

/\* if count == 2 then surely first chance was system's chance, now check the suitable position and make a move \*/

|  |
| --- |
| else if (count == 2)  {  if (Boxes[2][2] == 0)  {  Boxes[2][2] = 1;  count++;  }  else  {  Boxes[0][2] = 1;  count++;  }  } |

/\* if three boxes are full then in this condition user's had 7 empty boxes in the previous move \*/

/\* according to these moves system will make a move \*/

|  |
| --- |
| else if (count == 3)  {  if (Boxes[0][0] == 2 && Boxes[1][1] == 1 && Boxes[1][2] == 2)  {  Boxes[0][2] = 1;  count++;  }  else if (Boxes[0][0] == 2 && Boxes[1][1] == 1 && Boxes[2][1] == 2)  {  Boxes[2][0] = 1;  count++;  }  else if (Boxes[0][0] == 2 && Boxes[1][1] == 1 && Boxes[2][2] == 2)  {  Boxes[1][2] = 1;  count++;  }  else if (Boxes[0][0] == 1 && Boxes[0][1] == 2 && Boxes[0][2] == 2)  {  Boxes[2][2] = 1;  count++;  }  else if (Boxes[0][0] == 1 && Boxes[0][1] == 2 && Boxes[1][0] == 2)  {  Boxes[2][2] = 1;  count++;  }  else if (Boxes[0][0] == 1 && Boxes[0][1] == 2 && Boxes[1][2] == 2)  {  Boxes[0][2] = 1;  count++;  }  else if (Boxes[0][0] == 1 && Boxes[0][1] == 2 && Boxes[2][0] == 2)  {  Boxes[2][2] = 1;  count++;  }  else if (Boxes[0][0] == 1 && Boxes[0][1] == 2 && Boxes[2][2] == 2)  {  Boxes[2][0] = 1;  count++;  }  else if (Boxes[0][2] == 2 && Boxes[1][1] == 1 && Boxes[1][0] == 2)  {  Boxes[2][2] = 1;  count++;  }  else if (Boxes[0][2] == 2 && Boxes[1][1] == 1 && Boxes[2][0] == 2)  {  Boxes[2][1] = 1;  count++;  }  else if (Boxes[0][2] == 2 && Boxes[1][1] == 1 && Boxes[2][1] == 2)  {  Boxes[2][2] = 1;  count++;  }  else if (Boxes[1][0] == 2 && Boxes[0][2] == 1 && Boxes[0][1] == 2)  {  Boxes[2][0] = 1;  count++;  }  else if (Boxes[1][0] == 2 && Boxes[0][2] == 1 && Boxes[2][1] == 2)  {  Boxes[0][0] = 1;  count++;  }  else if (Boxes[1][0] == 2 && Boxes[0][2] == 1 && Boxes[2][2] == 2)  {  Boxes[0][0] = 1;  count++;  }  else if (Boxes[0][0] == 1 && Boxes[1][2] == 2 && Boxes[0][1] == 2)  {  Boxes[2][0] = 1;  count++;  }  else if (Boxes[0][0] == 1 && Boxes[1][2] == 2 && Boxes[2][0] == 2)  {  Boxes[2][2] = 1;  count++;  }  else if (Boxes[0][0] == 1 && Boxes[1][2] == 2 && Boxes[2][1] == 2)  {  Boxes[2][0] = 1;  count++;  }  else if (Boxes[1][1] == 1 && Boxes[2][0] == 2 && Boxes[0][1] == 2)  {  Boxes[2][2] = 1;  count++;  }  else if (Boxes[1][1] == 1 && Boxes[2][0] == 2 && Boxes[0][2] == 2)  {  Boxes[1][0] = 1;  count++;  }  else if (Boxes[1][1] == 1 && Boxes[2][0] == 2 && Boxes[1][2] == 2)  {  Boxes[2][2] = 1;  count++;  }  else if (Boxes[0][0] == 1 && Boxes[2][1] == 2 && Boxes[0][2] == 2)  {  Boxes[2][0] = 1;  count++;  }  else if (Boxes[0][0] == 1 && Boxes[2][1] == 2 && Boxes[1][0] == 2)  {  Boxes[2][2] = 1;  count++;  }  else if (Boxes[0][0] == 1 && Boxes[2][1] == 2 && Boxes[1][2] == 2)  {  Boxes[2][0] = 1;  count++;  }  else if (Boxes[1][1] == 1 && Boxes[2][2] == 2 && Boxes[0][0] == 2)  {  Boxes[0][1] = 1;  count++;  }  else if (Boxes[1][1] == 1 && Boxes[2][2] == 2 && Boxes[0][1] == 2)  {  Boxes[0][2] = 1;  count++;  }  else if (Boxes[1][1] == 1 && Boxes[2][2] == 2 && Boxes[1][0] == 2)  {  Boxes[0][2] = 1;  count++;  }  } |

/\* if four boxes are full then check for the previous moves of the system and make a move \*/

|  |
| --- |
| else if (count == 4)  {  if (Boxes[2][2] == 1)  {  if (Boxes[2][0] == 0)  {  Boxes[2][0] = 1;  count++;  }  }  else if (Boxes[0][2] == 1)  {  if (Boxes[2][0] == 0)  {  Boxes[2][0] = 1;  count++;  }  }  } |

/\* if five boxes are full then surely user had five empty boxes in the previous move then check all these five moves and take a decision \*/

|  |
| --- |
| else if (count == 5)  {  if (Boxes[0][0] == 2 && Boxes[0][1] == 1 && Boxes[0][2] == 2 && Boxes[1][1] == 1 && Boxes[2][1] == 2)  {  Boxes[1][2] = 1;  count++;  }  else if (Boxes[0][0] == 2 && Boxes[1][0] == 1 && Boxes[1][1] == 1 && Boxes[1][2] == 2 && Boxes[2][0] == 2)  {  Boxes[2][1] = 1;  count++;  }  else if (Boxes[0][0] == 1 && Boxes[0][1] == 2 && Boxes[1][1] == 2 && Boxes[2][1] == 1 && Boxes[2][2] == 2)  {  Boxes[2][0] = 1;  count++;  }  else if (Boxes[0][0] == 1 && Boxes[0][1] == 2 && Boxes[1][2] == 2 && Boxes[0][2] == 1 && Boxes[2][0] == 2)  {  Boxes[1][1] = 1;  count++;  }  else if (Boxes[0][0] == 1 && Boxes[0][1] == 2 && Boxes[1][2] == 2 && Boxes[0][2] == 1 && Boxes[2][2] == 2)  {  Boxes[2][0] = 1;  count++;  }  else if (Boxes[0][0] == 1 && Boxes[0][1] == 2 && Boxes[1][0] == 2 && Boxes[2][0] == 1 && Boxes[2][2] == 2)  {  Boxes[0][2] = 1;  count++;  }  else if (Boxes[0][0] == 2 && Boxes[0][1] == 1 && Boxes[0][2] == 2 && Boxes[1][1] == 1 && Boxes[2][1] == 2)  {  Boxes[1][0] = 1;  count++;  }  else if (Boxes[0][2] == 2 && Boxes[1][0] == 2 && Boxes[1][1] == 1 && Boxes[1][2] == 1 && Boxes[2][2] == 2)  {  Boxes[0][1] = 1;  count++;  }  else if (Boxes[0][0] == 2 && Boxes[0][1] == 2 && Boxes[0][2] == 1 && Boxes[1][0] == 2 && Boxes[2][0] == 2)  {  Boxes[2][2] = 1;  count++;  }  else if (Boxes[0][0] == 1 && Boxes[0][1] == 2 && Boxes[0][2] == 1 && Boxes[1][0] == 2 && Boxes[2][1] == 2)  {  Boxes[2][2] = 1;  count++;  }  else if (Boxes[0][0] == 1 && Boxes[0][1] == 2 && Boxes[0][2] == 1 && Boxes[1][0] == 2 && Boxes[2][2] == 2)  {  Boxes[2][0] = 1;  count++;  }  else if (Boxes[0][0] == 1 && Boxes[0][1] == 2 && Boxes[2][2] == 2 && Boxes[1][1] == 2 && Boxes[2][1] == 1)  {  Boxes[2][0] = 1;  count++;  }  else if (Boxes[0][0] == 1 && Boxes[1][0] == 2 && Boxes[1][1] == 2 && Boxes[1][2] == 1 && Boxes[2][2] == 2)  {  Boxes[0][2] = 1;  count++;  }  else if (Boxes[0][0] == 1 && Boxes[0][1] == 2 && Boxes[0][2] == 1 && Boxes[1][2] == 2 && Boxes[2][2] == 2)  {  Boxes[2][0] = 1;  count++;  }  else if (Boxes[0][0] == 2 && Boxes[1][0] == 1 && Boxes[1][1] == 1 && Boxes[1][2] == 2 && Boxes[2][0] == 2)  {  Boxes[2][1] = 1;  count++;  }  else if (Boxes[0][1] == 2 && Boxes[1][1] == 1 && Boxes[2][0] == 2 && Boxes[2][1] == 1 && Boxes[2][2] == 2)  {  Boxes[1][0] = 1;  count++;  }  else if (Boxes[0][0] == 1 && Boxes[0][2] == 2 && Boxes[1][0] == 2 && Boxes[2][0] == 1 && Boxes[2][1] == 2)  {  Boxes[0][1] = 1;  count;  }  else if (Boxes[0][0] == 1 && Boxes[1][0] == 2 && Boxes[2][0] == 1 && Boxes[2][1] == 2 && Boxes[2][2] == 2)  {  Boxes[0][2] = 1;  count++;  }  else if (Boxes[0][2] == 2 && Boxes[1][0] == 2 && Boxes[1][1] == 1 && Boxes[1][2] == 1 && Boxes[2][2] == 2)  {  Boxes[2][1] = 1;  count++;  }  else if (Boxes[0][1] == 2 && Boxes[1][1] == 1 && Boxes[2][0] == 2 && Boxes[2][1] == 1 && Boxes[2][2] == 2)  {  Boxes[1][0] = 1;  count++;  }  } |

/\* we don't need to do anything for the case in which six boxes are full because this can be managed by the previous two cases of computer's and user's victory \*/

/\* now in the case in which seven boxes are full, user had 3 empty boxes for moves take decisions according to these moves \*/

|  |
| --- |
| else if (count == 7)  {  flag = 0;  for (i = 0; i < 3; i++)  {  for (j = 0; j < 3; j++)  {  if (Boxes[i][j] == 0)  {  Boxes[i][j] = 1;  count++;  flag = 1;  break;  }  }  if (flag)  {  flag = 0;  break;  }  }  } |

/\* if eight boxes are full, we don't need to do anything because either this can be managed by two previous conditions of computer's and user's victory, or arbitrarily give the input \*/

/\* in the last case, arbitrarily give the input anywhere game will be drawn \*/

|  |
| --- |
| else if (count == 8)  {  flag = 0;  for (i = 0; i < 3; i++)  {  for (j = 0; j < 3; j++)  {  if (Boxes[i][j] == 0)  {  Boxes[i][j] = 1;  count++;  flag = 1;  break;  }  }  if (flag)  {  flag = 0;  break;  }  }  }  } |

/\* now it's user chance take input by the user \*/

|  |
| --- |
| fprintf (stdout, "\tGame Status is.\n");  for (i = 0; i < 3; i++)  {  for (j = 0; j < 3; j++)  {  printf ("\t%d\t", Boxes[i][j]);  if (j == 2)  {  printf("\n");  }  }  }  again1: fprintf (stdout, "\tEnter the Box.\t");  scanf ("%d%d", &i, &j);  if ((Boxes[i][j] != 0) || (i > 2) || (j > 2))  {  fprintf (stdout, "\tInvalid Box.\n");  goto again1;  }  Boxes[i][j] = 2;  count++; |

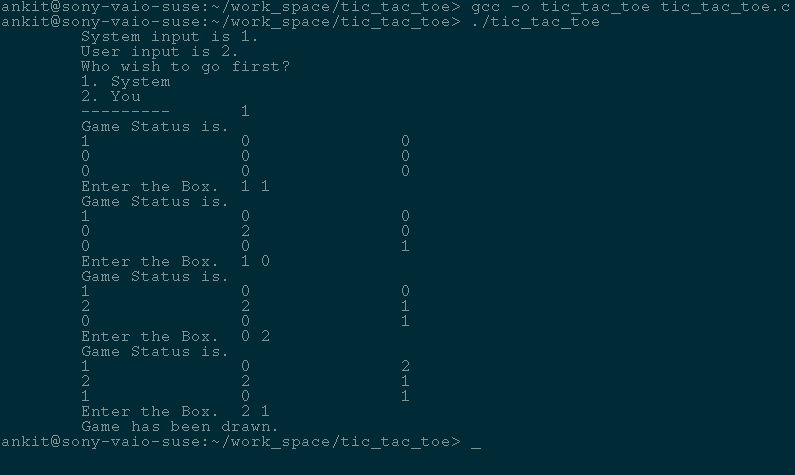
/\* After every move of user check it's victory \*/

|  |
| --- |
| for (i = 0; i < 3; i++)  {  for (j = 0; j < 3; j++)  {  if (Boxes[i][j] == 2)  {  if (i == 0 && j == 0)  {  if ((Boxes[1][1] == 2 && Boxes[2][2] == 2) || (Boxes[0][1] == 2 && Boxes[0][2] == 2) || (Boxes[1][0] == 2 && Boxes[2][0] == 2))  {  printf ("\tYou won.\n");  goto free;  }  }  else if (i == 0 && j == 1)  {  if ((Boxes[0][0] == 2 && Boxes[0][2] == 2) || (Boxes[1][1] == 2 && Boxes[2][1] == 2))  {  printf ("\tYou won.\n");  goto free;  }  }  else if (i == 0 && j == 2)  {  if ((Boxes[1][1] == 2 && Boxes[2][0] == 2) || (Boxes[0][0] == 2 && Boxes[0][1] == 2) || (Boxes[1][2] == 2 && Boxes[2][2] == 2))  {  printf ("\tYou won.\n");  goto free;  }  }  else if (i == 1 && j == 0)  {  if ((Boxes[0][0] == 2 && Boxes[2][0] == 2) || (Boxes[1][1] == 2 && Boxes[1][2] == 2))  {  printf ("\tYou won.\n");  goto free;  }  }  else if (i == 1 && j == 1)  {  if ((Boxes[0][0] == 2 && Boxes[2][2] == 2) || (Boxes[0][1] == 2 && Boxes[2][1] == 2) || (Boxes[0][2] == 2 && Boxes[2][0] == 2) || (Boxes[1][0] == 2 && Boxes[1][2] == 2))  {  printf ("\tYou won.\n");  goto free;  }  }  else if (i == 1 && j == 2)  {  if ((Boxes[0][2] == 2 && Boxes[2][2] == 2) || (Boxes[1][0] == 2 && Boxes[1][1] == 2))  {  printf ("\tYou won.\n");  goto free;  }  }  else if (i == 2 && j == 0)  {  if ((Boxes[1][1] == 2 && Boxes[0][2] == 2) || (Boxes[0][0] == 2 && Boxes[1][0] == 2) || (Boxes[2][1] == 2 && Boxes[2][2] == 2))  {  printf ("\tYou won.\n");  goto free;  }  }  else if (i == 2 && j == 1)  {  if ((Boxes[1][1] == 2 && Boxes[0][1] == 2) || (Boxes[2][0] == 2 && Boxes[2][2] == 2))  {  printf ("\tYou won.\n");  goto free;  }  }  else if (i == 2 && j == 2)  {  if ((Boxes[0][0] == 2 && Boxes[1][1] == 2) || (Boxes[0][2] == 2 && Boxes[1][2] == 2) || (Boxes[2][0] == 2 && Boxes[2][1] == 2))  {  printf ("\tYou won.\n");  goto free;  }  }  }  }  }  } |

/\* After all go for the match drawn condition and free all the allocated memory \*/

|  |
| --- |
| fprintf (stdout, "\tGame has been drawn.\n");  free: for (i = 0; i < 3; i++)  {  free (Boxes[i]);  }  free (Boxes);  return 0;  } |

If system go first



If user go first

