algorithms and techniques used in the program:

The program maintains the state of the game board as an array of 9 integers, with each integer representing a cell on the board. The picksX represents a cell marked by the human player with 'X', picksO represents a cell marked by the human player with 'O'. The program consists of several functions, including displayboard to print the current state of the game board, check\_win to check if a player has won the game, and the main function to handle player turns and game logic.

In each turn, the program prompts the current player to pick a space with its corresponding number and mark on the board. If the player is human, the program waits for input from the user using the syscall function. If the player is the computer, the program uses a simple algorithm to select an empty cell on the board at random. After a player marks a cell, the program calls the check\_win function to check if the player has won the game. If a player has won, the program prints a message declaring the winner and exits. If the game is tied, the program prints a message declaring a tie and exits.

The displayboard function prints the current state of the game board by looping through the array of board cells and printing the corresponding symbol for each value (picksX for ‘x’ , picksO for ‘O’). The check\_win function checks if a player has won the game by iterating over the rows, columns, and diagonals of the board and checking if the sum of the values in any of those groups is either 3 (indicating that the human player has won) or 6 (indicating that the computer player has won)

Overall, the programme implements the game logic using fundamental programming constructs including loops, conditional statements, and function calls. Additionally, it shows how to use arrays to represent the state of the game board and the syscall function to communicate with the user.