I'm not a robot

This snakes and ladders game is primarily designed as a semester project. It particularly aims at imparting general workable and practical knowledge about C and lets the beginners have adequate concept on programming small to big projects. It is developed with a hope to increase the programming logic and also to provide substitution to the classical game“Snake and Ladder”. Unlike other projects, there are many files, besides the source code files, that make up thissnakes and ladders project in C. You can directly download the program files of this project from the link below. Download Snakes and Ladders Game Project in C The game includes the facilities such as saving and loading the game. The saved games are password protected so that only the authorized user can load the previously saved game. Thesnakes and ladders game project comprises of multiple functions (scroll down to view the description for each) which can be dealt at ease. Attempt has been made to present the program in user friendly environment. Classical Snakes and Ladders OBJECTIVES To use different user defined function, to break a program into many simplified parts to deal with and to make it easier to understand the codes too. To promote the use of array in general programming for simplified version of the program. To apply the file handling concepts in order to retrieve the player’s data files that contains the data of the players. To use minimum graphics as far as possible to make the program execute fast but also user friendly side by side. To use general concept of c language to develop a simple snakes and ladders game that as a whole entertains the user. FEATURES Compatible as multi-player game. Facility to save the game. Open the saved game using the password pre-defined by the user. Use minimum graphics in snakes and ladders as far as possible avoiding complex codes. SCOPE Users preferring the classical games can switch to the game. Beginners in computer programming can take hints via the mini project to boost up their programming techniques. Can be brought in practice in training centers that offers basic programming courses. Can be the best choice as stress relieving game in the busy life pattern of the people in present life. Simplicity and efficient is main fact that we get using the c language, so user can execute snakes and ladders in each and every computer almost. Can be milestone for the programmers trying to learn to build projects in C. Here is a block diagram for this Snakes and Ladders game labelled as “The Viper Buzz”. Block Diagram Main() This function is the main function. This function contains the control of the whole game. User can switch to resume, new game, load game, save game, tutorial and exit from this function. It calls the respective function as user’s input. In this function user can use pre-defined short cut keys too. Save game() This part of the program is called from the main function when user wishes to save the game. The player can provide his own password to protect the game from other users. In snakes and ladders, this function calls the password function to get the password from the user. After getting the password the name of game, the name of players, and the value of the player is stored in a file named save.dat that can be reopened further. Loadgame() This function enables the users to load the initially saved snakes and ladders game. In order to prevent the unauthorized use of the loaded game, there is a password system that user has to go through before he opens or loads the saved game. In order to get game loaded user has to input the game name and the password per defined or per entered. Password functions is called for password input and for matching of the password and the game name data is read from the data file save.dat. Tutorial() This function in the program prints the tutorial of the snakes and ladders i.e. the help of the game, color indication and so on. The function uses a data file that contains the tutorial of the game. After printing the tutorial the control is transferred to main function. Firstscr() This function generates the first screen appeared when the program is executed. This function includes some animations and the snakes and ladders title. This function gets control until unless any key of the key board is hit. If any key of the key board is hit the control flows to main function. Main game() This is the function from where all the game is executed . From this function firstly the layout of the board is called. Then the vipers and the ladders are printed. Then the dice function is called and returned value is added to respective player’s position. Then after the viper and ladder function is called to check the viper and ladder position. Then after the players are displayed at the new position. At last eofgame function is called in this snakes and ladders if the player’s value equals to 100. Layout() In this function, the general layout of the board is drawn as like in classical game. This function does not get control for long. As soon as it get the control it prints the layout of the board and the control goes to the main game function. Drawviperandladder() In this function, the vipers and ladders are drawn on the board. This function is called from the main game function of this viper buzz project. The viper are represented by the red color and the ladders are represented by the green color. Dice() This function is for the value of the dice as per user’s reaction. In snakes and ladders, his function returns an integer value if any key of the keyboard is hit. The value is returned to the main game function. In addition to the dice rolling, this function holds the screen and also responds to the escape key pressed. If escape key is pressed, it flows control to the main menu else the control flows to main menu. Position() This is the function locating the player in the board of snakes and ladders. This function prints a white space in the old position and it prints the representation character to the new location as the value of the player. The main aim of this function is to prints the player’s indicator in the respective position. Viperandladder() In this function the player’s position is compared with the initial position of the ladder and respective value of the player is changed only if the initial value matches. In addition to that this function checks the hit condition also. The control flows to the main game after execution ofsnakes and ladders. Locate(int , int) This function is used in our program to set the cursor position on the different location of the screen. It takes two arguments and sets the cursor position according to the arguments. Draw() In this snakes and ladders game project, draw() function helps to print the dice box and the data of the player on the box. It is called each time loop executes so that it overlaps the initial mesh created. This function displays the player’s current position, player’s name and turn. Eofgame(char , int) This is the final screen after any one of the player reaches to the position of 100. The main loop of the main game function in this snakes and ladders gets break and this function is called. In this function two arguments are passed i.e. firstly the player’s name and the player’s code. This function has the control until unless a key is pressed. Password(char) In this function, an address is passed as argument and entered password is returned for the pointer. In this snakes and ladders game project, the character that user input is take and the asterisks are printed in place of the character to avoid the identification. This function is called from save game and the load game. Playfilesync(char) This function is used in the program to play the sound of the game. This function enables the programmer to embed the sound in their console based program. This function takes the file name to be played. The file to be played should be inwav format. Download Snakes and Ladders Game Project in C Here, individual algorithms and flowcharts have been present in a sequential manner for the functions that make up this gaming project. 1. main () Step 1: StartStep 2 : Display the menu items.Step 3: Check the user input and go to respective functions. 3(a): If 1 go to main game() function 3(b): if 2 go to new game 3(c): If 3 go to load game 3(d): If 4 go to save game 3(e): If 5 go to tutorial 3(f): If 6 go to step 4. Step 4: Stop 2. main game() Step 1: StartStep 2: Display the layout of the board of snakes and ladders from layout function.Step 3: Call the dice function and store returned value to a variable(c)Step 4: Check whether the p1 and p2 value is zero or not 4(a): If yes, check the dice value if it is 1 or 6 change respective player’s value to 1. 4(b): If no increase the value of turn, call position function and go to step 3. Step 5: Check whether sum of player’s value and dice(c) greater than 100 or not 5(a): If yes increase value of turn and go to step 3, Step 6: Check whether sum player’s value and dicer equals 100 or not 6(a): if yes, display the player name as a winner got main. 6(b): if no, add the players value and dicer. Step 8: Call the function position to locate the player on the board.Step 9: Check whether dice value is 1 or 6. 9(a): if yes, decrease turn by 1. Step 10: Increase the turn by 1.Step 11: Go to step 3.Step 12: Step 3. New game() Step 1: StartStep 2: Ask user whether to start a new game or not?Step 3: check the input character 3(a): if yes, Assign player1’s value, player2’s value and turn to 0 Ask for player1’s name and player’s name and store it to global structure of snakes and ladders. Call the function main game and change the flag value. 3(b): If no call the function main menu. Step 4: Stop 4. Save game() Step 1: StartStep 2: check the flag value initialized as global is equal to 46. 2(a): if matches to start up value (46) request user to start a new game and change the flag value. 2(b): If does not matches to startup value (46) Open a fie to store the data in append mode. Ask user for the input of game name and password. Write, player’s data, position , game name and password to the file. Display game saved Close the file and return to main menu. Step 3: Stop. 5. Load game () Step 1: startStep 2: Ask user for game name and password.Step 3: Open initially saved fie from save game function in read mode.Step 4: Compare the game name and password from user and data file. 4(a): If both game name and password matches Display match found. Assign player’s data and name to respective global variables Call the main game function 4(b) If match is not found go to main menu of snakes and ladders Step 5: Stop 6. Layout () Step 1: StartStep 2 : Initialize a variable (j) as 0.Step 3: Find the quotient when j is divided by 10.Step 4: Check the quotient value is even or not? 4(a): if yes, Assign the value of i as 1. Display the sum of i and j. Increase the value of i by 1. Check the value of i whether it is less than equals to 10 or not? If yes, goto step 4(a).ii If no goto step 5. 4(b): If no, Assign the value of I as 10 Display the sum of I and j. Decrease the value of I by 1 Check whether the value of i is greater than 0 or not? If yes, goto step 4(b).ii If no, go to step 5. Step 5: Increase the value of j by 10.Step 6: Check whether j if less than 100 or not? 6(a): If yes, goto step 3. 6(b): if no, goto step 7. Step 7: Go to main() function of snakes and ladders.Step 8: Stop. 7. Drawviperandladder () This flowchart is continued below: Step 1: StartStep 2: Open the data file containing the initial and final position of viper and the ladder.Step 3: Read initial and final position of viper and ladder.Step 4: Find the coordinates of intial and final position of the snakes and ladders and store to initial.x, initial.y, final.x and final.y.Step 5: Initialize a variable (a) with y coordinate of initial position.a=initial.yStep 6: Locate the Cursor position as initial.x and a.Step 7: Display asterisk (*).Step 8 : Increase the value of a by 1.Step 9: Check whether the value of a is less than final.y or not? 9(a): If yes, go to step 6. 9(b): If no, go to step 10. Step 10: Initialize a variable (b) with x coordinate of initial position. b=initial.xStep 11: Locate the Cursor position as b and initial.y.Step 12: Display asterisk (*).Step 13 : Increase the value of b by 1.Step 14: Check whether the value of b is less than final.x or not? 14(a): If yes, go to step 11. 14(b): If no, go to step 15. Step 15: Check whether it is end of file or not? 15(a): If yes, go to step 17. 15(b): if no, go to step 3. Step 16: Go to main() function of snakes and ladders.Step 17: Stop. 8. Dice () Step 1: Start.Step 2: Initialize a variable as 1.Step 3: Increase the value of the variable by 1.Step 4: Check whether there is keyboard input or not? 4(a): if yes, return the value of the variable as the dice value and go to main(). 4(b): if no, go to step 5. Step 5: Check whether counter variable equals 6 or not? 5(a): if yes, assign the value of counter as 1 and goto Step 3. 5(b): if no, goto Step 3. Step 6: Stop. 9. Position () Step 1: Start.Step 2: Display white space on the initial position of the players.Step 3: Get the new player’s position.Step 4: Locate the X and Y coordinate of the player’s position.Step 5: Display the respective player to the position.Step 6: Go to the main.Step 7: Stop. 10. Viperandladder () Step 1: Start.Step 2: Open the file containing initial and the final position of the viper and ladder.Step 3: Read initial and final position of viper and ladders.Step 4: Check the player’s position with the initial position of the viper and ladder, whether it matches or not? 4(a): If yes, Assign the value of player with the final value of viper or ladder. 4(b): if no, go to Step 5. Step 5: Check whether the position of the player 1 and the position of player 2 matches or not? 5(a): If yes, Check the turn whether it is odd or not? If yes, assign player 1’s position equal to 1. If no, assign player 2’s position equal to 1. 5(b): if no, go to step 6. Step 6: Go to main().Step 7: Stop. The Snakes And Ladders Game project in C With Source Code is developed using C programming language.This simple Game is good for the students or beginners who wants to learn programming especially in C Programming Language.A snake and ladder game in c language was created specifically for a semester project.It aims to give beginners a good understanding of programming small to large projects by imparting general workable and practical information about C.It was created in the hopes of improving programming logic and serving as a replacement for the classic game Snake and Ladder.This snake and ladder game in c programming design is straightforward and clean, making it easy for users to learn, use, and navigate.In this project is also includes a downloadable snake and ladder game using c for free, just find the downloadable source code below and click to start downloading.Anyway if you want level up your knowledge in programming especially C/C++ Programming Language, try this new article I’ve made for you Best C Projects with Source Code for Beginners Free Download.To run this code for snake and ladder game in c make sure that you have a Code Blocks or any platform of C installed in your computer.ABOUT PROJECTPROJECT DETAILSProject Name :snake and ladder gameProject Platform :C/C++Programming Language Used:C Programming LanguageDeveloper Name :itsourcecode.comIDE Tool (Recommended):Dev-C++/CodeblocksProject Type :Desktop ApplicationDatabase:Stores data in .DAT fileSnakes And Ladders Game In C Language With Source Code InformationTime needed: 5 minutesThese are the steps on how to run Snakes And Ladders Game In C With Source CodeStep 1: DownloadFirst, download the source code given below.Step 2: Extract file.Second, after you finished download the source code, extract the zip file.Step 3: Open CodeBlocksThird, open “CodeBlocks IDE”.Step 4: Open Project.Fourth, open file tab and Open File after that open folder SnakesAndLadders then click the “snake-n-ladder.c”.Step 5: Run ProjectFifth, run the project.Step 6: The actual code.You are free to copy the given source code below or download the downloadable source code given.Click Here to Download the Source CodeThis snake and ladder game in c graphics is only a project made for school requirement purposes only.You can download this program and modify it to suit your client requirements, since this is a student project it means you cannot really expect 100% functionality from this.If you have any questions or suggestions about snake and ladder game in c using graphics, please feel free to leave a comment below. In this game project, I have presented the source code along with the algorithm and flowchart for the project under the name “Viper Buzz”. ‘The Viper Buzz” is a reflection of the famous classic game ‘Snakes and Ladders’. Designed as a computer version of the South Asian game, this final year project is written on C language using GNU GCC compiler on Code::Blocks IDE. It is an entertaining game that has been made user friendly with multiple user compatibility. This snakes and ladders game is primarily designed as a semester project. It particularly aims at imparting general workable and practical knowledge about C and lets the beginners have adequate concept on programming small to big projects. It is developed with a hope to increase the programming logic and also to provide substitution to the classical game “Snake and Ladder”. Unlike other projects, there are many files, besides the source code files, that make up this snakes and ladders project in C. You can directly download the program files of this project from the link below. [sociallocker] Download Snakes and Ladders Game Project in C with Source Code (Updated: Sound Files) [/sociallocker] The game includes the facilities such as saving and loading the game. The saved games are password protected so that only the authorized user can load the previously saved game. The snakes and ladders game project comprises of multiple functions (scroll down to view the description for each) which can be dealt at ease. Attempt has been made to present the program in user friendly environment. Classical Snakes and Ladders OBJECTIVES To use different user defined function, to break a program into many simplified parts to deal with and to make it easier to understand the codes too. To promote the use of array in general programming for simplified version of the program. To apply the file handling concepts in order to retrieve the player’s data files that contains the data of the players. To use minimum graphics as far as possible to make the program execute fast but also user friendly side by side. To use general concept of c language to develop a simple snakes and ladders game that as a whole entertains the user. FEATURES Compatible as multi-player game. Facility to save the game. Open the saved game using the password pre-defined by the user. Use minimum graphics in snakes and ladders as far as possible avoiding complex codes. SCOPE Users preferring the classical games can switch to the game. Beginners in computer programming can take hints via the mini project to boost up their programming techniques. Can be brought in practice in training centers that offers basic programming courses. Can be the best choice as stress relieving game in the busy life pattern of the people in present life. Simplicity and efficient is main fact that we get using the c language, so user can execute snakes and ladders in each and every computer almost. Can be milestone for the programmers trying to learn to build projects in C. Here is a block diagram for this Snakes and Ladders game labelled as “The Viper Buzz”. Block Diagram Main() This function is the main function. This function contains the control of the whole game. User can switch to resume, new game, load game, save game, tutorial and exit from this function. It calls the respective function as user’s input. In this function user can use pre-defined short cut keys too. Save game() This part of the program is called from the main function when user wishes to save the game. The player can provide his own password to protect the game from other users. In snakes and ladders, this function calls the password function to get the password from the user. After getting the password the name of game, the name of players, and the value of the player is stored in a file named save.dat that can be reopened further. Loadgame() This function enables the users to load the initially saved snakes and ladders game. In order to prevent the unauthorized use of the loaded game, there is a password system that user has to go through before he opens or loads the saved game. In order to get game loaded user has to input the game name and the password per defined or per entered. Password functions is called for password input and for matching of the password and the game name data is read from the data file save.dat. Tutorial() This function in the program prints the tutorial of the snakes and ladders i.e. the help of the game, color indication and so on. The function uses a data file that contains the tutorial of the game. After printing the tutorial the control is transferred to main function. Firstscr() This function generates the first screen appeared when the program is executed. This function includes some animations and the snakes and ladders title. This function gets control until unless any key of the key board is hit. If any key of the key board is hit the control flows to main function. Main game() This is the function from where all the game is executed . From this function firstly the layout of the board is called. Then the vipers and the ladders are printed. Then the dice function is called and returned value is added to respective player’s position. Then after the viper and ladder function is called to check the viper and ladder position. Then after the players are displayed at the new position. At last eofgame function is called in this snakes and ladders if the player’s value equals to 100. Layout() In this function, the general layout of the board is drawn as like in classical game. This function does not get control for long. As soon as it get the control it prints the layout of the board and the control goes to the main game function. Drawviperandladder() In this function, the vipers and ladders are drawn on the board. This function is called from the main game function of this viper buzz project. The viper are represented by the red color and the ladders are represented by the green color. Dice() This function is for the value of the dice as per user’s reaction. In snakes and ladders, his function returns an integer value if any key of the keyboard is hit. The value is returned to the main game function. In addition to the dice rolling, this function holds the screen and also responds to the escape key pressed. If escape key is pressed, it flows control to the main menu else the control flows to main menu. Position() This is the function locating the player in the board of snakes and ladders. This function prints a white space in the old position and it prints the representation character to the new location as the value of the player. The main aim of this function is to prints the player’s indicator in the respective position. Viperandladder() In this function the player’s position is compared with the initial position of the ladder and respective value of the player is changed only if the initial value matches. In addition to that this function checks the hit condition also. The control flows to the main game after execution of snakes and ladders. Locate(int , int) This function is used in our program to set the cursor position on the different location of the screen. It takes two arguments and sets the cursor position according to the arguments. Draw() In this snakes and ladders game project, draw() function helps to print the dice box and the data of the player on the box. It is called each time loop executes so that it overlaps the initial mesh created. This function displays the player’s current position, player’s name and turn. Eofgame(char , int) This is the final screen after any one of the player reaches to the position of 100. The main loop of the main game function in this snakes and ladders gets break and this function is called. In this function two arguments are passed i.e. firstly the player’s name and the player’s code. This function has the control until unless a key is pressed. Password(char) In this function, an address is passed as argument and entered password is returned for the pointer. In this snakes and ladders game project, the character that user input is take and the asterisks are printed in place of the character to avoid the identification. This function is called from save game and the load game. Playfilesync(char) This function is used in the program to play the sound of the game. This function enables the programmer to embed the sound in their console based program. This function takes the file name to be played. The file to be played should be inwav format. Here, individual algorithms and flowcharts have been present in a sequential manner for the functions that make up this gaming project. 1. main () Step 1: Start Step 2 : Display the menu items. Step 3: Check the user input and go to respective functions. The Snakes And Ladders Game project in C With Source Code is developed using C programming language.This simple Game is good for the students or beginners who wants to learn programming especially in C Programming Language.A snake and ladder game in c language was created specifically for a semester project.It aims to give beginners a good understanding of programming small to large projects by imparting general workable and practical information about C.It was created in the hopes of improving programming logic and serving as a replacement for the classic game Snake and Ladder.This snake and ladder game in c programming design is straightforward and clean, making it easy for users to learn, use, and navigate.In this project is also includes a downloadable snake and ladder game using c for free, just find the downloadable source code below and click to start downloading.Anyway if you want level up your knowledge in programming especially C/C++ Programming Language, try this new article I’ve made for you Best C Projects with Source Code for Beginners Free Download.To run this code for snake and ladder game in c make sure that you have a Code Blocks or any platform of C installed in your computer.ABOUT PROJECTPROJECT DETAILSProject Name :snake and ladder gameProject Platform :C/C++Programming Language Used:C Programming LanguageDeveloper Name :itsourcecode.comIDE Tool (Recommended):Dev-C++/CodeblocksProject Type :Desktop ApplicationDatabase:Stores data in .DAT fileSnakes And Ladders Game In C Language With Source Code InformationTime needed: 5 minutesThese are the steps on how to run Snakes And Ladders Game In C With Source CodeStep 1: DownloadFirst, download the source code given below.Step 2: Extract file.Second, after you finished download the source code, extract the zip file.Step 3: Open CodeBlocksThird, open “CodeBlocks IDE”.Step 4: Open Project.Fourth, open file tab and Open File after that open folder SnakesAndLadders then click the “snake-n-ladder.c”.Step 5: Run ProjectFifth, run the project.Step 6: The actual code.You are free to copy the given source code below or download the downloadable source code given.Click Here to Download the Source CodeThis snake and ladder game in c graphics is only a project made for school requirement purposes only.You can download this program and modify it to suit your client requirements, since this is a student project it means you cannot really expect 100% functionality from this.If you have any questions or suggestions about snake and ladder game in c using graphics, please feel free to leave a comment below. End 99 S(2) 97 96 95 94 93 92 9181 82 83 84 85 86 87 88 89 9080 79 78 77 76 L(99) 74 73 72 7161 62 S(22) 64 65 66 67 68 69 7060 59 58 57 56 55 54 53 52 5141 42 43 44 L(68) 46 47 48 49 5040 39 S(6) 37 36 35 34 33 32 3121 22 23 24 25 26 27 28 29 3020 19 18 17 S(9) 15 14 13 12 11Start 2 3 4 5 6 7 8 9 10Constraints :1 MAX_PLAYERS) { printf("Invalid number of players. Please try again."); return; } // Initialize the random number generator srand(time(NULL)); // Game loop while (1) { // Roll the dice diceValue = generateRandomNumber(1, 6); // Update the current player's position newPosition = players[currentPlayer] + diceValue; // Check if the new position exceeds the board size if (newPosition > BOARD_SIZE) { printf("Player %d rolled %d, but cannot move beyond position %d.", currentPlayer + 1, diceValue, BOARD_SIZE); } else { // Check if the new position is a snake or ladder for (int i = 0; i < 7; i++) { if (newPosition == snakes[i][0]) { newPosition = snakes[i][1]; printf("Player %d landed on a snake! Moved to position %d.", currentPlayer + 1, newPosition); break; } else if (newPosition == ladders[i][0]) { newPosition = ladders[i][1]; printf("Player %d landed on a ladder! Moved to position %d.", currentPlayer + 1, newPosition); break; } } // Update the player's position players[currentPlayer] = newPosition; // Check if the player has won if (newPosition == BOARD_SIZE) { printf("Player %d has won the game!", currentPlayer + 1); break; } // Print the current player's position printf("Player %d rolled %d and moved to position %d.", currentPlayer + 1, diceValue, newPosition); } // Move to the next player currentPlayer = (currentPlayer + 1) % numPlayers; // Wait for the player to press a key getch(); } } int main() { // Initialize the graphics mode int gd = DETECT, gm; initgraph(&gd, &gm, ""); // Call the snake and ladder game function playSnakeAndLadder(); // Close the graphics mode closegraph(); return 0; } The Snake game is a simple arcade game where the player controls a snake that moves around to eat food. The goal is to keep eating food without crashing into the walls. The game ends when the snake crashes, with the player's score based on how much food was collected.In this article, we will learn how to create snake game using C programming language.PrerequisiteWe can create a console-based snake game using basic knowledge of C along with the following prerequisites:Working C compiler and IDE.For windows, we need . For linux, we need . These libraries are generally installed by default.We might need to install library as it does not comes bundled with the standard C compiler.GameplayThe gameplay of the snake game is simple. There are three game objects:Snake: The snake automatically moves into given direction till we change it. It has a mouth and tail.Fruit: The object that snake has to eat to grow.Walls: They are the boundary of the game area.These objects work according to the game rules which are as follows:The snake moves at a constant speed in one direction, and we can change its direction using the movement keys. However, the snake can only turn 90° at a time.The objective is to eat as many fruits as possible, which will make the snake to grow in length.The game ends if the snake collides with the boundary or its own body.Implementation of Snake Game in COur objective is to create an interactive console-based snake game using C. The whole console screen can be visualized as the grid where each point has two coordinates, x-axis and y-axis coordinates. The position of each object the game will be described by these coordinates. All the game will be based on the continuous updating of these coordinates. The below is the intuition for the implementation of different game mechanics:Snake Implementation LogicThe snake can be represented by the coordinates of its body stored in two arrays:One Array for x-axis coordinates.One Array for y-axis coordinates.The head of the snake moves according to the user input:W for upS for downA for leftD for rightwhile the rest of the body follows. When the snake eats some fruit, its length increases, and the coordinate array is updated. Collision detection is implemented to checking if the snake coordinates overlap (self-collision) or goes over the boundary limit (boundary collision).Fruit Implementation LogicThe fruit is represented as a single coordinate within the game area. It is randomly placed inside the boundary, ensuring it doesn't spawn outside the playable area. The fruit have the following functions:Random Positioning: The fruit's position is generated using random coordinates within the height and width of the game area, excluding the boundary.Snake Eating: If the snake's head reaches the fruit's position, the fruit is "eaten", the snake grows in length, and the score increases.Boundary Implementation LogicThe boundary defines the playing area of the game. It is created by drawing a box or rectangle that surrounds the playing field using some characters (usually # or | for the sides and - for the top and bottom), and it acts as a collision zone. We can define macros for setting the boundary limits.Game Motion Implementation LogicThe motion in game is generated by an infinite loop that continuously updates the position of the snake and checks for game events such as collisions, fruit consumption or user input. In each iteration of the loop, the snake moves one step in the current direction, and the screen is redrawn in place or previous one to reflect the new position. It is similar to creating static images as frames and move them fast enough to create a motion.Real Time Response LogicGenerally, we need to press the Enter key for the input buffer to supply data to the program. But this does not provide real time response. For real time interactive response, we use the kbhit() function provided by . This function returns the pressed character immediately as soon as the key is pressed on the keyboard.Snake Game Code in C C // C program to implement an interactive // snake game #include #include #include #include #include // Macros variable (HEIGHT, WIDTH) #define HEIGHT 20 #define WIDTH 40 // Array to store the coordinates of snake // tail (x-axis, y-axis) int snakeTailX[100], snakeTailY[100]; // Variable to store the length of the // snake's tail int snakeTailLen; // Score and signal flags int gameover, key, score; // Coordinates of snake's head and fruit int x, y, fruitx, fruity; // Function to generate the fruit // within the boundary void setup() { // Flag to signal the gameover gameover = 0; // Initial coordinates of the snake x = WIDTH / 2; y = HEIGHT / 2; // Initial coordinates of the fruit fruitx = rand() % WIDTH; fruity = rand() % HEIGHT; while (fruitx == 0) fruitx = rand() % WIDTH; while (fruity == 0) fruity = rand() % HEIGHT; // Score initialzed score = 0; } // Function to draw the game field, snake // and fruit void draw() { system("cls"); // Creating top wall for (int i = 0; i < WIDTH + 2; i++) printf("-"); printf(""); for (int i = 0; i < HEIGHT; i++) { for (int j = 0; j