```
import json
def main():
    runs the main loop
    calls a menu
    sends control to other parts of the program
    handles invalid input from menu
    menuChoice = getMenuChoice()
    if menuChoice == “0”:
        exit
    if menuChoice == “1”:
        loadGame()
    if menuChoice == “2”:
        loadGame()
    if menuChoice == “3”:
        saveGame()
    if menuChoice == “4”:
        editNode()
    if menuChoice == “5”:
        playGame()
    else:
        print(f”Invalid input.”)
def getMenuChoice():
    print(f””””
        0) exit
        1) load default game
        2) load a game file
        3) save the current file
```

4) edit or add a node
5) play the current game

“””)

menuChoice = input(“What will you do? “)

return menuChoice

def playGame():

call getDefaultGame to get defaultGame

set currentKey to “start”

use while keepGoing to iterate through defaultGame dictionary

set keepGoing to True

while keepGoing:

if currentKey gets “quit”,

set keepGoing to False (exiting loop)

else:

send game and currentKey to playNode function

currentKey gets result of playNode

def playNode():

parameters: defaultGame, currentKey

create var “currentNode”; “currentNode” looks up the node from the currentKey (defaultGame[currentKey])

(description, menuA, nodeA, menuB, nodeB) = currentNode

print the menu

call {description}

1) {menuA}
2) {menuB}

choice = input(what will you do?)

if choice == “1”:

```
            nextKey == nodeA
        if choice == “2”:
            nextKey == nodeB
        else:
            nextKey == currentKey
        return nextKey
def getDefaultGame():
        creates a single-node default game
        returns that data structure
        defaultGame = {
            “start”: [“Do you want to win or lose?”, “I’m a winner!”, “win”, “I’m a big
            loser”, “lose”]
            “win”: [“You win! I knew you could do it.”, “Start over”, “start”, “Quit”, “quit”]
            “lose”: [“You lose! I’m so disappointed in you.”, “Start over”, “start”, “Quit”,
            “quit”]
        generates dictionary (defaultGame)
        return defaultGame
def editNode():
        given the current game structure,
        list all of the current content print(json.dumps(defaultGame, indent = 2))
        nodeName = get node name input(“what node do you want to edit or create?”)
            if node exists,
                copy that node to newNode
            else (IE doesn’t exist yet)
                create newNode with empty data

        use editField() to allow user to edit each node
```

```
        return the now edited newNode
def editField():
        gets a field name
        print the field's current value
        if the user presses "enter" immediately without changing anything
                retain the current value
        else
                use the new value
def saveGame():
        save the game to a data file
        preset the file name
        print the current game dictionary in human-readable format (pretty-printing)
        save the file in JSON format
        defaultGame = getDefaultGame()
        outFile = open("defaultGame.json", "w")
        json.dump(defaultGame, outFile, indent=2)
        outFile.close
        print(json.dumps(defaultGame, indent=2))
        print("saved defaultGame data to defaultGame.json")
def loadGame():
        presume there is a data file named *whatever you preset it as in saveGame()*
        open that file
        load the data into the game object
        return that game object
        inFile = open("defaultGame.json", "r")
        defaultGame = json.load(inFile)
```

```
inFile.close

return defaultGame
```