Post Mortem Review Question	Response
What was the purpose of your program?	To let the user play an adventure game. Where their goal is to collect items and increase their score.
How could your program be useful in the real world?	It could give the user some entertainment once more features are implemented into the game.
What is a problem you ran into, and how did you fix it?	I didn’t really run into any problems while writing/compiling my code.
Describe one thing you would do differently the next time you write a program.	Add a score based off of the item's config. Add a crafting part of the game. Make the game have a UI.
How could your program be generalized and useful in other areas?	It could be used for a demonstration as to what coding can do.

PSUDOCODE
START

```
❖ Import random, time and builtins
❖ Print "Type quit at any time to quit the program!"
❖ Initialize the items set in a list of dicts or a database
❖ Initialize score itemsCollected and the collect count
❖ Initialize the min and max searching time for the program
❖ Define addItem(item):
    ➢ Global itemsCollected
    ➢ Set itemName to the item.name
    ➢ Set itemQuantity to item.quantity
    ➢ Set itemFound to tell the program that the item was found in the database or
      list
    ➢ Foreach in range length of itemsCollected
        ▪ Collected equals the index of itemsCollected
        ▪ If (collected.name == itemName)
            • Currentquantity  = item.quantity or default 0
            • Break
    ➢ If the item isnt found
        ▪ Append the item to itemsCollected
❖ Define input
    ➢ Initialize input as the result of builtins#input asking prompt
    ➢ If input lower and stip equals quit
        ▪ Quit the program
    ➢ Set input to input lower and strip
    ➢ If input is blank return the default value
    ➢ Return input if not blank
❖ Define printWelcomeMessage
    ➢ Wait 1 sec
    ➢ Print "Welcome to the Adventure Game!"
    ➢ Wait 1 sec
    ➢ Print "Your goal is to collect {itemsCollectCount} items before
      completing the level."
    ➢ Wait 1 sec
❖ Define collectItems
    ➢ Initialize teh globals items, itemsCollected, score,itemsCollectCount, min
      and max searchTime
```

```
        ➢ While score less than itemsCollectCount
            ▪ Print "Searching..."
            ▪ Wait random range of min and max searchTime
            ▪ Initialize ItemToFind as index of items by randomrange of length of
              items
            ▪ Ask the user if then wnt to collect itemToFind
            ▪ If the input is equal to y or yes
                • Call addItem with params {name,quantity}
                • Increase score by 1
                • Print "You collected an item! Current score: {score}"
            ▪ Else if userinput equals no or n
                • Print "You chose not to collect the item."
            ▪ Else
                • Print "Invalid input. Please enter 'yes' or 'no'."
                • Call addItem with params {name,quantity}
                • Increase score by 1
                • Print "You collected an item! Current score: {score}"
        ➢ Print "Congratulations! You collected the Following items: "
        ➢ For each item in itemsCollected
            ▪ Print str(item.get("quantity")) + "x " + item.get("name")
        ➢ Print "Your final score is: {score}"
    ❖ Define main
        ➢ Initialize globals itemsCollectCount
        ➢ Call printWelcomeMessage
        ➢ While True
            ▪ Ask user if they want to play
                • If input equals y or yes
                    ♦ Call collectItems
                • Else
                    ♦ Print "Thank you for playing! Goodbye.”
                    ♦ Break
                • Initialize playAgainInput as the input opf asking if the usr wants
                  to play again
                    ♦ If not yes or not y
                        ➢ Print "Thank you for playing! Goodbye."
                        ➢ Break
                    ♦ Increase ItemsCollectCount by an increment of 5
```

- Call main

CODE

```python
from random import randrange
import time
import builtins



print ("Type quit at any time to quit the program!")
#3 initialize the items that can be colleted
items = [
    {"name": "rusty sword"},  # add other options
    {"name": "stone"},
    {"name": "wood"},
    {"name": "leaf"},
    {"name": "iron"},
]
#initialize the score and the current items collected
score = 0
itemsCollected = []
# teh base amount of items to collect
itemsCollectCount=3

#min search time in sec
minSearchTime = 3
#max search time in sec
maxSearchTime = 15
def add_item(item):
    global itemsCollected
    itemName = item.get("name")
    itemQuantity = int(item.get("quantity", 0))

    # Flag to check if the item was found and updated
    item_found = False

    # Iterate over the collected items
    for i in range(len(itemsCollected)):
        collected = itemsCollected[i]
        if collected.get("name") == itemName:
            # Update the quantity for the existing item
            currentQuantity = int(collected.get("quantity", 0))
            newQuantity = itemQuantity + currentQuantity
            itemsCollected[i] = {"name": itemName, "quantity": newQuantity}
            item_found = True
            break

    # If the item was not found in the list, add it as a new item
    if not item_found:
        itemsCollected.append({"name": itemName, "quantity": itemQuantity})
def input(prompt: object = "",default:str="") -> str:
    input = builtins.input(prompt)
    # if the input is equal to "quit" quit the progam and exit any processes that
be linger
    if input.lower().strip() == "quit":
        print("Thank You for playing! Goodbye :)")
        quit()
    input = input.strip().lower()
    # if the input is blank return the default value if there is one
    if (input==""): return default
    return input

# welcome the user
def printWelcomeMessage():
    time.sleep(1)
    print("Welcome to the Adventure Game!")
    time.sleep(1)
    print(f"Your goal is to collect {itemsCollectCount} items before completing
the level.")
    time.sleep(1)



def collectItems():
    # allow the inner function to use the outside variables
    global items, itemsCollected,
score,itemsCollectCount,minSearchTime,maxSearchTime




    while score < itemsCollectCount:
        print("Searching...")
        # tell teh program to sleep whle it searches for an item
        time.sleep(randrange(minSearchTime,maxSearchTime+1))
        itemToFind = items[randrange(5)]
        # ask the user if they want the item
        userInput = input(
            "You have found 1x "
            + itemToFind.get("name")
            + ". Do you want to collect it? (default:y) (y/n): ","yes"
        )
        # if the input equals y or yes make the user pickup the item
        if userInput.lower() == "yes" or "y":
            add_item({"name": itemToFind.get("name"), "quantity": 1})
            score += 1
            print(f"You collected an item! Current score: {score}")
            # if no tell the user they didnt want the item
        elif userInput == "no" or "n":
            print("You chose not to collect the item.")
            # if its an input other than yes y no n then tell tghe user its an
invalid input and that they collected the item
        else:
            print("Invalid input. Please enter 'yes' or 'no'.")
            add_item({"name": itemToFind.get("name"), "quantity": 1})
            score += 1
            print(f"You collected an item! Current score: {score}")

    # tell the user the items they have collected
    print("Congratulations! You collected the Following items: ")
    for item in itemsCollected:
        print(str(item.get("quantity")) + "x " + item.get("name"))
    print(f"Your final score is: {score}")



def main():
    global itemsCollectCount
    # print the welcome messages
    printWelcomeMessage()

    while True:
        userInput = input("Do you want to start collecting items? (default:y)
(y/n): ","yes")
        if userInput == "yes" or userInput == "y" or userInput == "ye":
            collectItems()
        else:
            print("Thank you for playing! Goodbye.")
            break

        playAgainInput = input("Do you want to play again? (default:y) (y/n):
","yes")

        if playAgainInput != "yes" or not "y":
            print("Thank you for playing! Goodbye.")
            break
        itemsCollectCount=+5



main()
```