```python
# prompt: 1.Text-Based Adventure Game
# Assignment Description:
# Develop a text-based adventure game where the player can navigate
through different rooms, pick up items, and interact with objects and
characters. Implement at least three different rooms and a simple
inventory system.
# Requirements:
# -Use functions to handle different rooms and actions.
# -Store items in a dictionary.
# -Include user input for navi

# Define the rooms dictionary
rooms = {
    'Hallway': {
        'description': 'You are standing in a long, dimly lit
hallway.',
        'items': ['key'],
        'exits': {'north': 'Kitchen', 'south': 'Living Room'}
    },
    'Kitchen': {
        'description': 'You are in a small, cozy kitchen.',
        'items': ['apple'],
        'exits': {'south': 'Hallway'}
    },
    'Living Room': {
        'description': 'You are in a spacious living room.',
        'items': ['book'],
        'exits': {'north': 'Hallway'}
    }
}

# Define the inventory dictionary
inventory = {}

# Define the current room
current_room = 'Hallway'

# Game loop
while True:
    # Print the current room description
    print(rooms[current_room]['description'])

    # Print the available items in the current room
    if len(rooms[current_room]['items']) > 0:
        print('Items in this room:')
        for item in rooms[current_room]['items']:
            print(f'- {item}')
```

```
    # Get user input
    command = input('What do you want to do? (move, pickup, or
quit)').lower()

    # Handle user input
    if command == 'move':
        # Get the direction the user wants to move
        direction = input('Which direction do you want to move?
(north, south, east, or west)').lower()

        # Check if the direction is valid
        if direction in rooms[current_room]['exits']:
            # Move to the new room
            current_room = rooms[current_room]['exits'][direction]
        else:
            print('You cannot move in that direction.')
    elif command == 'pickup':
        # Get the item the user wants to pick up
        item = input('Which item do you want to pick up?').lower()

        # Check if the item is in the current room
        if item in rooms[current_room]['items']:
            # Add the item to the inventory
            inventory[item] = True
            print(f'You picked up the {item}.')

            # Remove the item from the room
            rooms[current_room]['items'].remove(item)
        else:
            print(f'There is no {item} in this room.')
    elif command == 'quit':
        # Exit the game
        break
    else:
        print('Invalid command.')
```

```
You are standing in a long, dimly lit hallway.
Items in this room:
- key
What do you want to do? (move, pickup, or quit)move
Which direction do you want to move? (north, south, east, or
west)north
You are in a small, cozy kitchen.
Items in this room:
- apple
What do you want to do? (move, pickup, or quit)exit
Invalid command.
You are in a small, cozy kitchen.
Items in this room:
```

```
- apple
What do you want to do? (move, pickup, or quit)quit
```

```
rooms = {
    'Hallway': {
        'description': 'You are standing in a long, dimly lit
hallway.',
        'items': ['key'],
        'exits': {'north': 'Kitchen', 'south': 'Living Room'}
    },
    'Kitchen': {
        'description': 'You are in a small, cozy kitchen.',
        'items': ['apple'],
        'exits': {'south': 'Hallway'}
    },
    'Living Room': {
        'description': 'You are in a spacious living room.',
        'items': ['book'],
        'exits': {'north': 'Hallway'}
    }
}

inventory = {}
current_room = "Hallway"

while True:
    print(rooms[current_room]["description"])
    if len(rooms[current_room]["items"]) > 0:
        print('Items in this room:')
        for item in rooms[current_room]["items"]:
            print(f"- {item}")

    command = input("What do you want to do? (move, pickup, or quit)
").lower()

    if command == "move":
        direction = input("Enter the direction you want to move
(north, south, east, west): ").lower()
        if direction in rooms[current_room]["exits"]:
            current_room = rooms[current_room]["exits"][direction]
        else:
            print("You cannot move in that direction.")

    elif command == "pickup":
        item = input("Enter the item you want to pick up: ").lower()
        if item in rooms[current_room]["items"]:
            inventory[item] = True
            print(f"You picked up the {item}.")
            rooms[current_room]["items"].remove(item)
        else:
```

```
            print(f"There is no {item} in this room.")

    elif command == "quit":
        break

    else:
        print("Invalid command.")
```

```
You are standing in a long, dimly lit hallway.
Items in this room:
- key
What do you want to do? (move, pickup, or quit) quit
```

```
import random

# 1. Define the list of words
word_list = ["aardvark", "baboon", "camel"]

# 2. Choose a random word from the list
chosen_word = random.choice(word_list)

# 3. Initialize an empty list to store the guessed letters
guessed_letters = []

# 4. Initialize a variable to track the number of incorrect guesses
incorrect_guesses = 0

# 5. Set the maximum number of incorrect guesses allowed
max_incorrect_guesses = 6

# 6. Start the game loop
while incorrect_guesses < max_incorrect_guesses:
    # 6.1 Print the current state of the word
    all_letters_guessed = True
    for letter in chosen_word:
        if letter in guessed_letters:
            print(letter, end=" ")
        else:
            print("_", end=" ")
            all_letters_guessed = False
    print()  # Newline for better formatting

    # Check if the word has been completely guessed
    if all_letters_guessed:
        print("Congratulations! You guessed the word!")
        break

    # 6.2 Print the number of incorrect guesses left
    print(f"You have {max_incorrect_guesses - incorrect_guesses}
guesses remaining.")
```

```
    # 6.3 Get the player's guess
    guess = input("Enter a letter: ").lower()

    # 6.4 Check if the guess is valid
    if not guess.isalpha() or len(guess) != 1:
        print("Invalid input. Please enter a single letter.")
        continue

    # 6.5 Check if the guess has already been made
    if guess in guessed_letters:
        print("You already guessed that letter.")
        continue

    # 6.6 Add the guess to the list of guessed letters
    guessed_letters.append(guess)

    # 6.7 Check if the guess is correct
    if guess in chosen_word:
        print("Correct!")
    else:
        print("Incorrect.")
        incorrect_guesses += 1

# 7. End of the game loop
if incorrect_guesses == max_incorrect_guesses:
    print(f"You lost! The word was {chosen_word}.")
```

```
_ _ _ _ _ _
You have 6 guesses remaining.
Enter a letter: a
Correct!
_ a _ _ _ _
You have 6 guesses remaining.
Enter a letter: c
Incorrect.
_ a _ _ _ _
You have 5 guesses remaining.
Enter a letter: b
Correct!
b a b _ _ _
You have 5 guesses remaining.
Enter a letter: o
Correct!
b a b o o _
You have 5 guesses remaining.
Enter a letter: n
Correct!
b a b o o n
Congratulations! You guessed the word!
```

```python
# prompt: 4. Expense Tracker
# Assignment Description:
# Build an expense tracker that allows users to log their daily
expenses and generate a report of total expenditures.
# Requirements:
# Create functions to add and list expenses.
# Generate a summary report of total expenditures.

# Initialize variables
expenses = []

# Function to add an expense
def add_expense(item, amount):
    expenses.append({'item': item, 'amount': amount})

# Function to list all expenses
def list_expenses():
    for expense in expenses:
        print(f"{expense['item']} - ${expense['amount']}")

# Function to generate a summary report
def generate_report():
    total_expenses = sum([expense['amount'] for expense in expenses])
    print(f"Total expenses: ${total_expenses}")

# Main program loop
while True:
    # Display menu
    print("Expense Tracker")
    print("1. Add Expense")
    print("2. List Expenses")
    print("3. Generate Report")
    print("4. Exit")

    # Get user input
    choice = int(input("Enter your choice: "))

    # Perform corresponding action
    if choice == 1:
        item = input("Enter the item: ")
        amount = float(input("Enter the amount: "))
        add_expense(item, amount)
    elif choice == 2:
        list_expenses()
    elif choice == 3:
        generate_report()
    elif choice == 4:
        break
    else:
```

```
        print("Invalid choice.")


# Assignment Description:
# Create a contact book application that allows users to add, remove,
search, and list contacts. Ensure that contacts persist by saving them
to a file.

# Requirements:

# Implement functions for adding, removing, searching, and listing
contacts.
# Use file I/O to save and load contacts.
# Handle exceptions for file operations and invalid input.
# Contacts dictionary to store contacts
contacts = {}

# Add a new contact
def add_contact(name, phone):
    contacts[name] = phone
    print("Contact added successfully.")

# Remove a contact
def remove_contact(name):
    if name in contacts:
        del contacts[name]
        print("Contact removed successfully.")
    else:
        print("Contact not found.")

# Search for a contact by name
def search_contact(name):
    if name in contacts:
        print(f"Name: {name}, Phone: {contacts[name]}")
    else:
        print("Contact not found.")

# List all contacts
def list_contacts():
    if contacts:
        for name, phone in contacts.items():
            print(f"Name: {name}, Phone: {phone}")
    else:
        print("No contacts found.")

# Main function to run the application

while True:
        print("\nContact Book Menu:")
        print("1. Add Contact")
```

```
        print("2. Remove Contact")
        print("3. Search Contact")
        print("4. List All Contacts")
        print("5. Exit")

        choice = input("Enter your choice: ")

        if choice == "1":
            name = input("Enter name: ")
            phone = input("Enter phone number: ")
            add_contact(name, phone)
        elif choice == "2":
            name = input("Enter name to remove: ")
            remove_contact(name)
        elif choice == "3":
            name = input("Enter name to search: ")
            search_contact(name)
        elif choice == "4":
            list_contacts()
        elif choice == "5":
            break
```

```
Contact Book Menu:
1. Add Contact
2. Remove Contact
3. Search Contact
4. List All Contacts
5. Exit
Enter your choice: 5
```