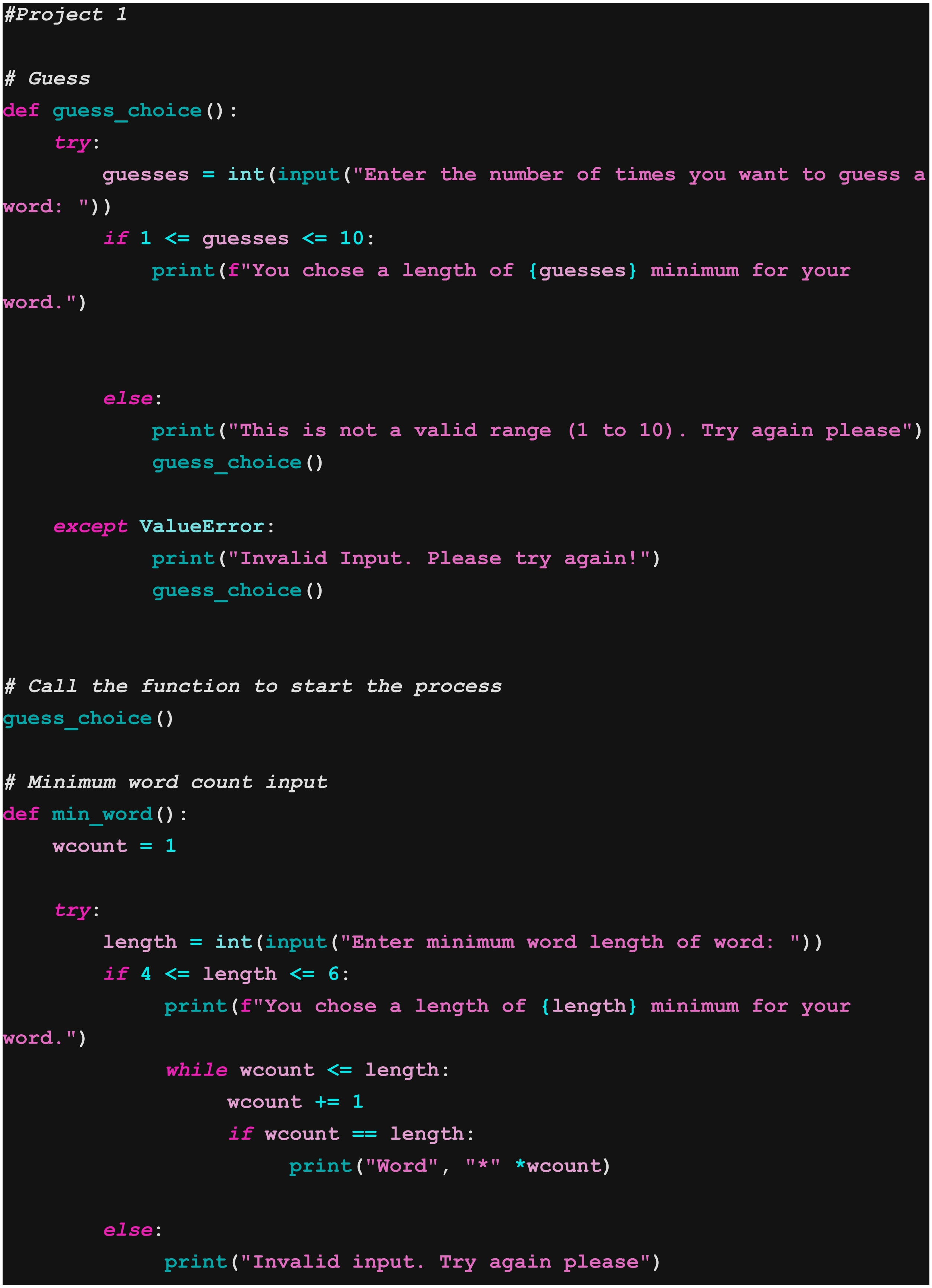
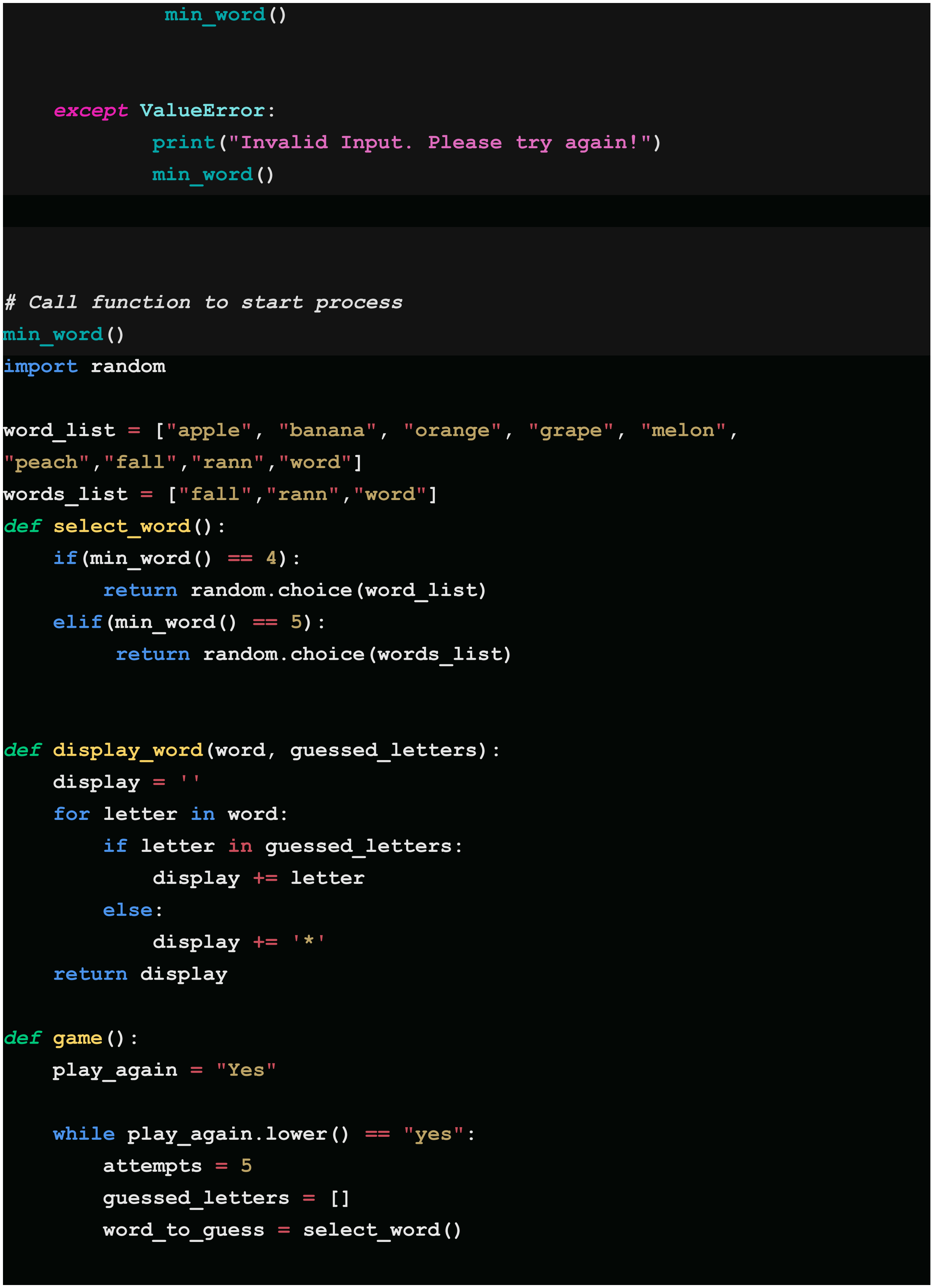
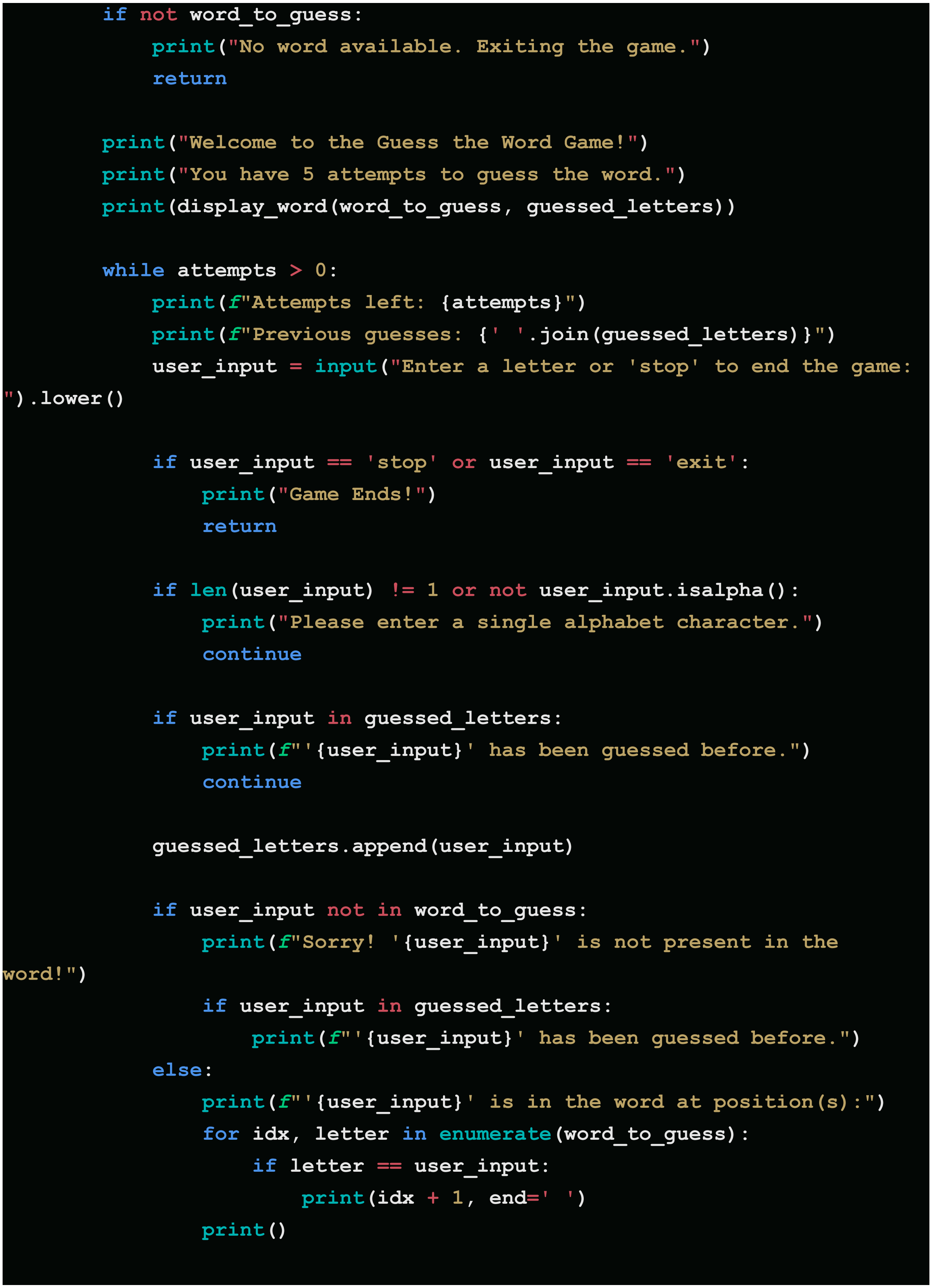


```
#Project 1

# Guess
def guess_choice():
    try:
        guesses = int(input("Enter the number of times you want to guess a
word: "))
        if 1 <= guesses <= 10:
            print(f"You chose a length of {guesses} minimum for your
word.")


        else:
            print("This is not a valid range (1 to 10). Try again please")
            guess_choice()

    except ValueError:
            print("Invalid Input. Please try again!")
            guess_choice()


# Call the function to start the process
guess_choice()

# Minimum word count input
def min_word():
    wcount = 1

    try:
        length = int(input("Enter minimum word length of word: "))
        if 4 <= length <= 6:
             print(f"You chose a length of {length} minimum for your
word.")
             while wcount <= length:
                  wcount += 1
                  if wcount == length:
                       print("Word", "*" *wcount)

        else:
             print("Invalid input. Try again please")
```

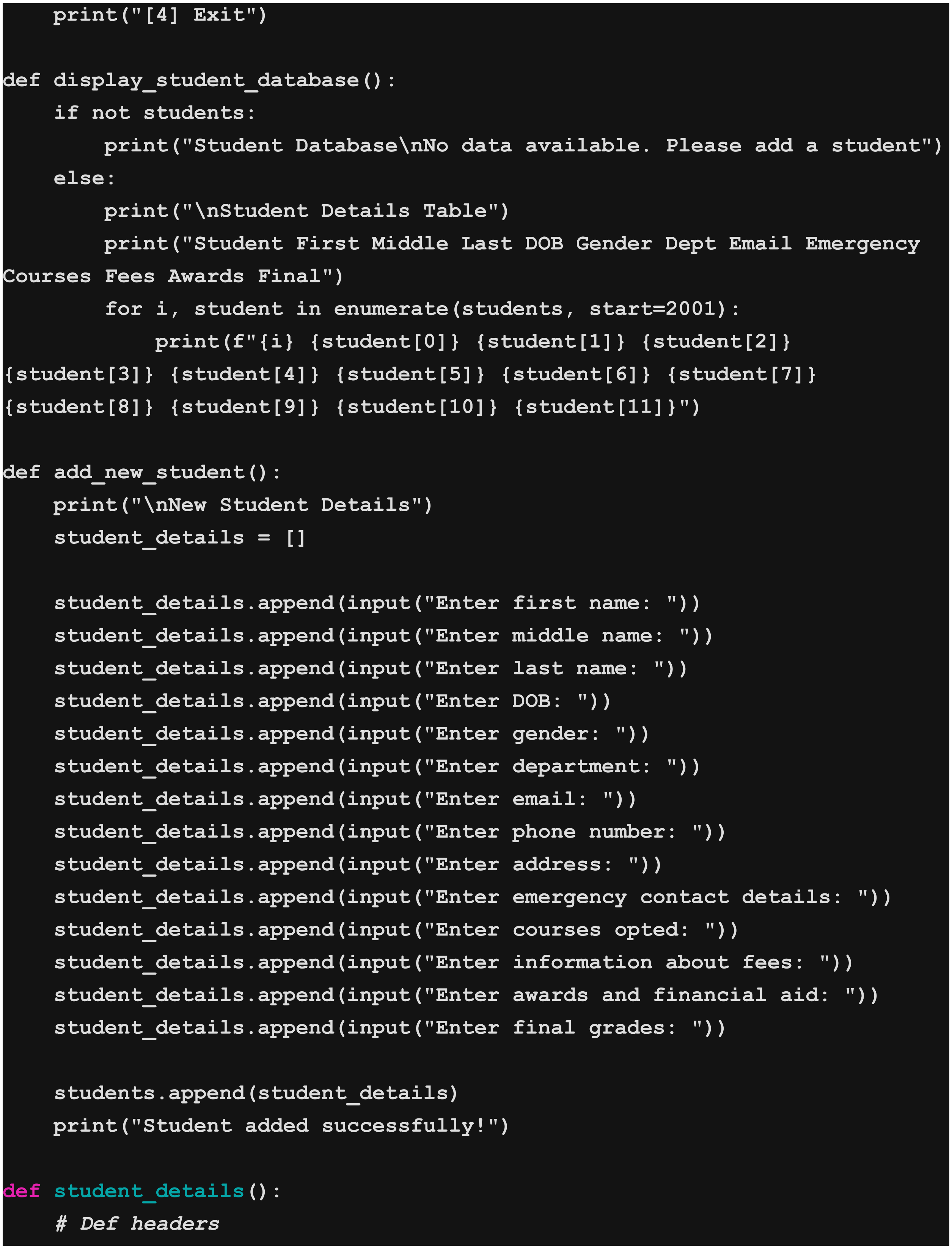


```
             min_word()


    except ValueError:
            print("Invalid Input. Please try again!")
            min_word()



# Call function to start process
min_word()
import random

word_list = ["apple", "banana", "orange", "grape", "melon",
"peach","fall","rann","word"]
words_list = ["fall","rann","word"]
def select_word():
    if(min_word() == 4):
        return random.choice(word_list)
    elif(min_word() == 5):
         return random.choice(words_list)


def display_word(word, guessed_letters):
    display = ''
    for letter in word:
        if letter in guessed_letters:
            display += letter
        else:
            display += '*'
    return display

def game():
    play_again = "Yes"

    while play_again.lower() == "yes":
        attempts = 5
        guessed_letters = []
        word_to_guess = select_word()
```



```
        if not word_to_guess:
            print("No word available. Exiting the game.")
            return

        print("Welcome to the Guess the Word Game!")
        print("You have 5 attempts to guess the word.")
        print(display_word(word_to_guess, guessed_letters))

        while attempts > 0:
            print(f"Attempts left: {attempts}")
            print(f"Previous guesses: {' '.join(guessed_letters)}")
            user_input = input("Enter a letter or 'stop' to end the game:
").lower()

            if user_input == 'stop' or user_input == 'exit':
                print("Game Ends!")
                return

            if len(user_input) != 1 or not user_input.isalpha():
                print("Please enter a single alphabet character.")
                continue

            if user_input in guessed_letters:
                print(f"'{user_input}' has been guessed before.")
                continue

            guessed_letters.append(user_input)

            if user_input not in word_to_guess:
                print(f"Sorry! '{user_input}' is not present in the
word!")
                if user_input in guessed_letters:
                    print(f"'{user_input}' has been guessed before.")
            else:
                print(f"'{user_input}' is in the word at position(s):")
                for idx, letter in enumerate(word_to_guess):
                    if letter == user_input:
                        print(idx + 1, end=' ')
                print()
                current_display = display_word(word_to_guess, guessed_letters)
                print(current_display)
                if '*' not in current_display:
                    print("Congratulations! You guessed the word!")
                    break

            attempts -= 1

        def play_again():
            replay = input("Would you like to play again? Types 'yes/no")
            re = replay.lower()


            if re == "yes":
                guess_choice()

            else:
                print("GoodBye:D")
```

```
def play_again():
    replay = input("Would you like to play again? Types 'yes/no")
    re = replay.lower()

    if re == "yes":
         guess_choice()

    else:
        print("GoodBye:D")

#End

#Project 2

import random

def choose_rand():
    # Function to choose a random word from the list
    words = ["programming", "example", "player", "shuffle", "python",
"jumbled", "winner", "terminate", "restart", "format"]
    return random.choice(words)

def shuffle_letters(word):
    # Function to shuffle the letters of a word
    return ''.join(random.sample(word, len(word)))

def final_score(player1, player2):
    # Function to print the final scores of both players
    print(f"{player1['name']}, your score: {player1['score']}")
    print(f"{player2['name']}, your score: {player2['score']}")

def declare_winner(player1, player2):
    # Function to declare the winner or a draw
    if player1['score'] > player2['score']:
        print(f"{player1['name']}, congratulations you are the winner!
Thanks for playing!")
    elif player2['score'] > player1['score']:
        print(f"{player2['name']}, congratulations you are the winner!
Thanks for playing!")
    else:
        print("It's a draw!")

def play_game():
    # Function to manage the game flow
    print("Welcome to Two-Player-Jumbled-Words Game")

    # Initialize player dictionaries with names and scores
    player1 = {"name": input("Player 1, enter your name: "), "score": 0}
    player2 = {"name": input("Player 2, enter your name: "), "score": 0}

    turn = 1
    while True:
        # Select a random word and shuffle its letters
        word = choose_rand()
        jumbled_word = shuffle_letters(word)
        print(f"Jumbled word is: {jumbled_word}")
        # Determine the current player based on the turn
        current_player = player1 if turn % 2 == 1 else player2
        print(f"{current_player['name']}, your turn")

        # Get the player's answer
        answer = input(f"{current_player['name']}, your answer is:
").lower()

        # Check if the player wants to stop or exit the game
        if answer == 'stop' or answer == 'exit':
            print("Game is paused!")
            final_score(player1, player2)
            break

        # Check if the answer is in the correct format (only alphabets)
        if not answer.isalpha():
            print("Incorrect format! Please enter the correct format (only
alphabets)")
            continue

        # Check if the answer is correct and update the score
        if answer == word:
            current_player['score'] += 1
            print("Right Answer!")
            print(f"Your score: {current_player['score']}")
        else:
            print("You are wrong!")
            print(f"Your score: {current_player['score']}")

        # Switch to the next player's turn
        turn += 1

    # Declare the winner or a draw
    declare_winner(player1, player2)

    # Ask if the players want to restart the game
    restart = input("Do you want to restart the game? (Type 'Y' for Yes
and 'N' for No): ").lower()
    if restart == 'y':
        play_game()
    else:
        print("Game Ends!")

# Start the game
play_game()

#End








#Project 3

# Default login credentials
ADMIN_USERNAME = "Admin"
ADMIN_PASSWORD = "password"

# Student database
students = []

def admin_login():
    print("Admin Login")
    username = input("Enter username: ")
    password = input("Enter password: ")

    if username == ADMIN_USERNAME and password == ADMIN_PASSWORD:
        return True
    else:
        print("Invalid credentials. Exiting...")
        return False

def display_main_menu():
    print("\nStudent Ledger")
    print("Select an option:")
    print("[1] Add a student")
    print("[2] Display student database")
    print("[3] Search student details")
```

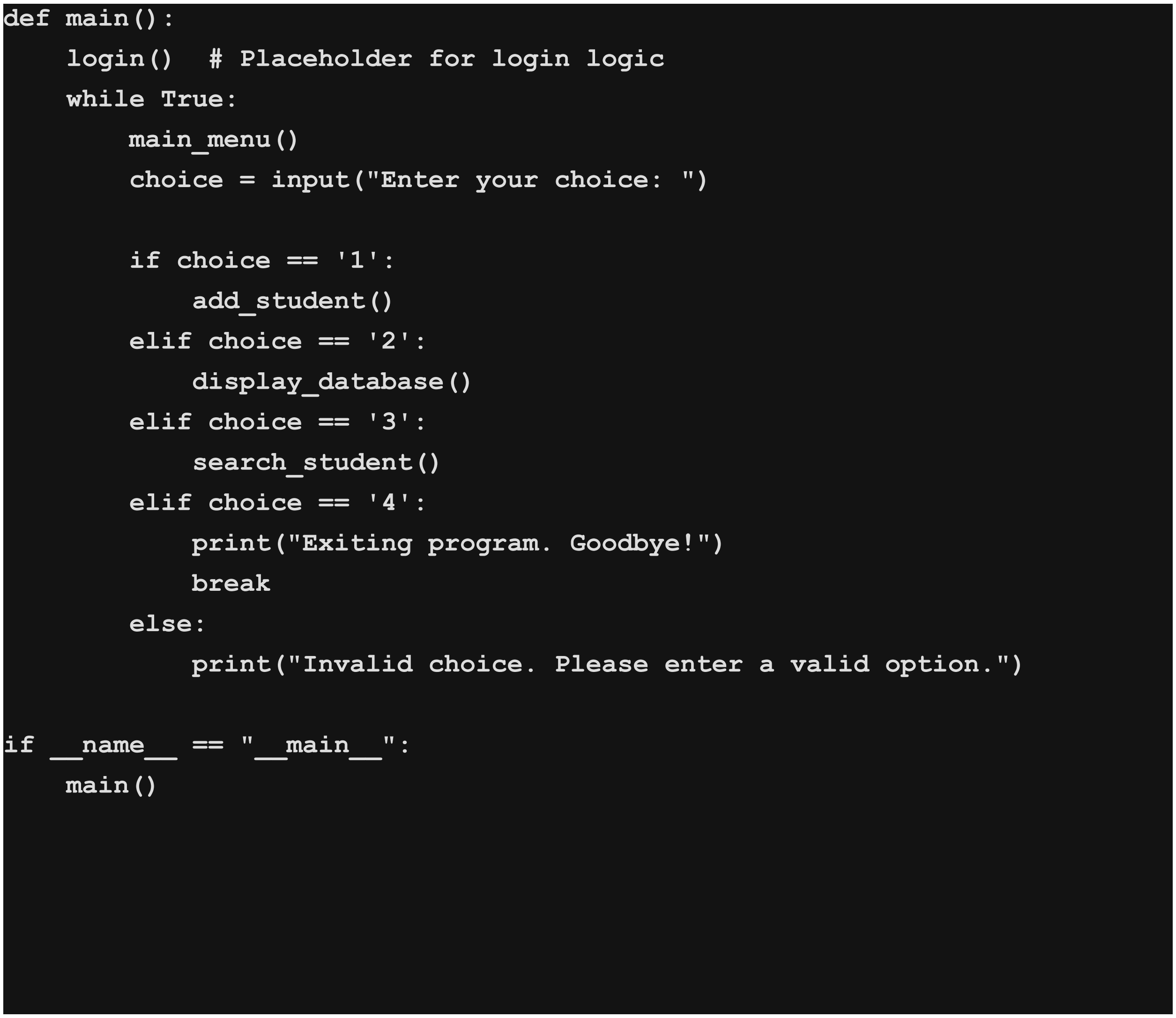


```
    print("[4] Exit")

def display_student_database():
    if not students:
        print("Student Database\nNo data available. Please add a student")
    else:
        print("\nStudent Details Table")
        print("Student First Middle Last DOB Gender Dept Email Emergency
Courses Fees Awards Final")
        for i, student in enumerate(students, start=2001):
            print(f"{i} {student[0]} {student[1]} {student[2]}
{student[3]} {student[4]} {student[5]} {student[6]} {student[7]}
{student[8]} {student[9]} {student[10]} {student[11]}")

def add_new_student():
    print("\nNew Student Details")
    student_details = []

    student_details.append(input("Enter first name: "))
    student_details.append(input("Enter middle name: "))
    student_details.append(input("Enter last name: "))
    student_details.append(input("Enter DOB: "))
    student_details.append(input("Enter gender: "))
    student_details.append(input("Enter department: "))
    student_details.append(input("Enter email: "))
    student_details.append(input("Enter phone number: "))
    student_details.append(input("Enter address: "))
    student_details.append(input("Enter emergency contact details: "))
    student_details.append(input("Enter courses opted: "))
    student_details.append(input("Enter information about fees: "))
    student_details.append(input("Enter awards and financial aid: "))
    student_details.append(input("Enter final grades: "))

    students.append(student_details)
    print("Student added successfully!")

def student_details():
    # Def headers
```

```python
    c_header = ["Student #", "First Name  ", "Middle Name  ", "Last Name ", "DOB", "Gender", "Dept", "Email", "Emerge Contacts ", "Courses", "Fees", "Awards & Financial Aid", "Final Grades"]

    # Calc width per column
    c_width = [max(len(c), 10) for c in c_header]

    # Print headers + adjusts
    print("|".join(f"{c:{width}}" for c, width in zip(c_header, c_width)))

    # Sep line
    print("-" * (sum(c_width) + 16 ))

    # Final Print
    for i in range(1, 4): # 4 for the wanted rows
        rows_val =[f"Value {i}", f"First Name {i}", f"Middle Name {i}", f"Last Name {i}", f"DOB {i}", f"Gender {i}", f"Dept {i}", f"Email {i}", f"Emerg Contacts {i}", f"Courses {i}", f"Fees {i}", f"Awards {i}", f"Final Grades {i}"]
        print("|".join(f"{val:{width}}" for val, width in zip(rows_val, c_width)))

# Call func
student_details()


def main():
    if not admin_login():
        return

    while True:
        display_main_menu()

        choice = input("Enter your choice: ")

        if choice == '1':
            add_new_student()
        elif choice == '2':
            display_student_database()
        elif choice == '3':
            # Add function for 'search student details'
            pass
        elif choice == '4':
            print("Exiting the system. Goodbye!")
            break
        else:
            print("Invalid choice. Please try again.")

if __name__ == "__main__":
    main()

#End









def student_details():
    # Def headers
    c_header = ["Student #", "First Name  ", "Middle Name  ", "Last Name
", "DOB", "Gender", "Dept", "Email", "Emerge Contacts ", "Courses",
"Fees", "Awards & Financial Aid", "Final Grades"]

    # Calc width per column
    c_width = [max(len(c), 10) for c in c_header]

    # Print headers + adjusts
    print("|".join(f"{c:{width}}" for c, width in zip(c_header, c_width)))

    # Sep line
    print("-" * (sum(c_width) + 16 ))

    # Final Print
    for i in range(1, 4): # 4 for the wanted rows
        rows_val =[f"Value {i}", f"First Name {i}", f"Middle Name {i}",
f"Last Name {i}", f"DOB {i}", f"Gender {i}", f"Dept {i}", f"Email {i}",
f"Emerg Contacts {i}", f"Courses {i}", f"Fees {i}", f"Awards {i}", f"Final
Grades {i}"]
        print("|".join(f"{val:{width}}" for val, width in zip(rows_val,
c_width)))

# Call func
student_details()
```

```
def add_new_student():
    print("\nNew Student Details")
    students = []

    first_name = (input("Enter first name: "))
    middle_name = (input("Enter middle name: "))
    last_name = (input("Enter last name: "))
    dob = (input("Enter DOB: "))
    gender = (input("Enter gender: "))
    department = (input("Enter department: "))
    email = (input("Enter email: "))
    phone = (input("Enter phone number: "))
    address = (input("Enter address: "))
    e_contact = (input("Enter emergency contact details: "))
    courses = (input("Enter courses opted: "))
    fees = (input("Enter information about fees: "))
    a_a = (input("Enter awards and financial aid: "))
    f_grades = (input("Enter final grades: "))
```

```
#Project 3 Final

students = []

def login():
    print("Admin Login")
    username = input("Enter username: ")
    password = input("Enter password: ")
    # Add authentication logic if needed

def main_menu():
    print("Student Ledger")
    print("Select an option:")
    print("[1] Add a student")
    print("[2] Display student database")
    print("[3] Search student details")
    print("[4] Exit")

def display_database():
    print("Student Database")
    if not students:
        print("No data available. Please add a student")
    else:
        for index, student in enumerate(students, start=1):
            print(f"Student {index}:")
            for key, value in student.items():
                print(f"{key}: {value}")
            print()

def search_student():
    if not students:
        print("No data available. Please add a student")
    else:
        search_name = input("Enter the first name to search: ")
        found = False
        for student in students:
            if student["First Name"].lower() == search_name.lower():
                found = True
                print("Student found:")
                for key, value in student.items():
                    print(f"{key}: {value}")
                break

        if not found:
            print(f"No student found with the first name '{search_name}'")

def add_student():
    print("New Student Details")
    first_name = input("Enter first name: ")
    middle_name = input("Enter middle name: ")
    last_name = input("Enter last name: ")
    dob = input("Enter dob: ")
    gender = input("Enter gender: ")
    department = input("Enter department: ")
    email = input("Enter email: ")
    emergency_contact = input("Enter emergency contact details: ")
    courses = input("Courses opted: ")
    fees = input("Information about fees: ")
    awards = input("Awards and financial aid: ")
    final_grades = input("Final grades: ")

    student_details = {
        "First Name": first_name,
        "Middle Name": middle_name,
        "Last Name": last_name,
        "DOB": dob,
        "Gender": gender,
        "Department": department,
        "Email": email,
        "Emergency Contact": emergency_contact,
        "Courses": courses,
        "Fees": fees,
        "Awards": awards,
        "Final Grades": final_grades
    }

    students.append(student_details)  # Append to the list
    print("Student added successfully!")
```



```
def main():
    login()  # Placeholder for login logic
    while True:
        main_menu()
        choice = input("Enter your choice: ")

        if choice == '1':
            add_student()
        elif choice == '2':
            display_database()
        elif choice == '3':
            search_student()
        elif choice == '4':
            print("Exiting program. Goodbye!")
            break
        else:
            print("Invalid choice. Please enter a valid option.")

if __name__ == "__main__":
    main()
```