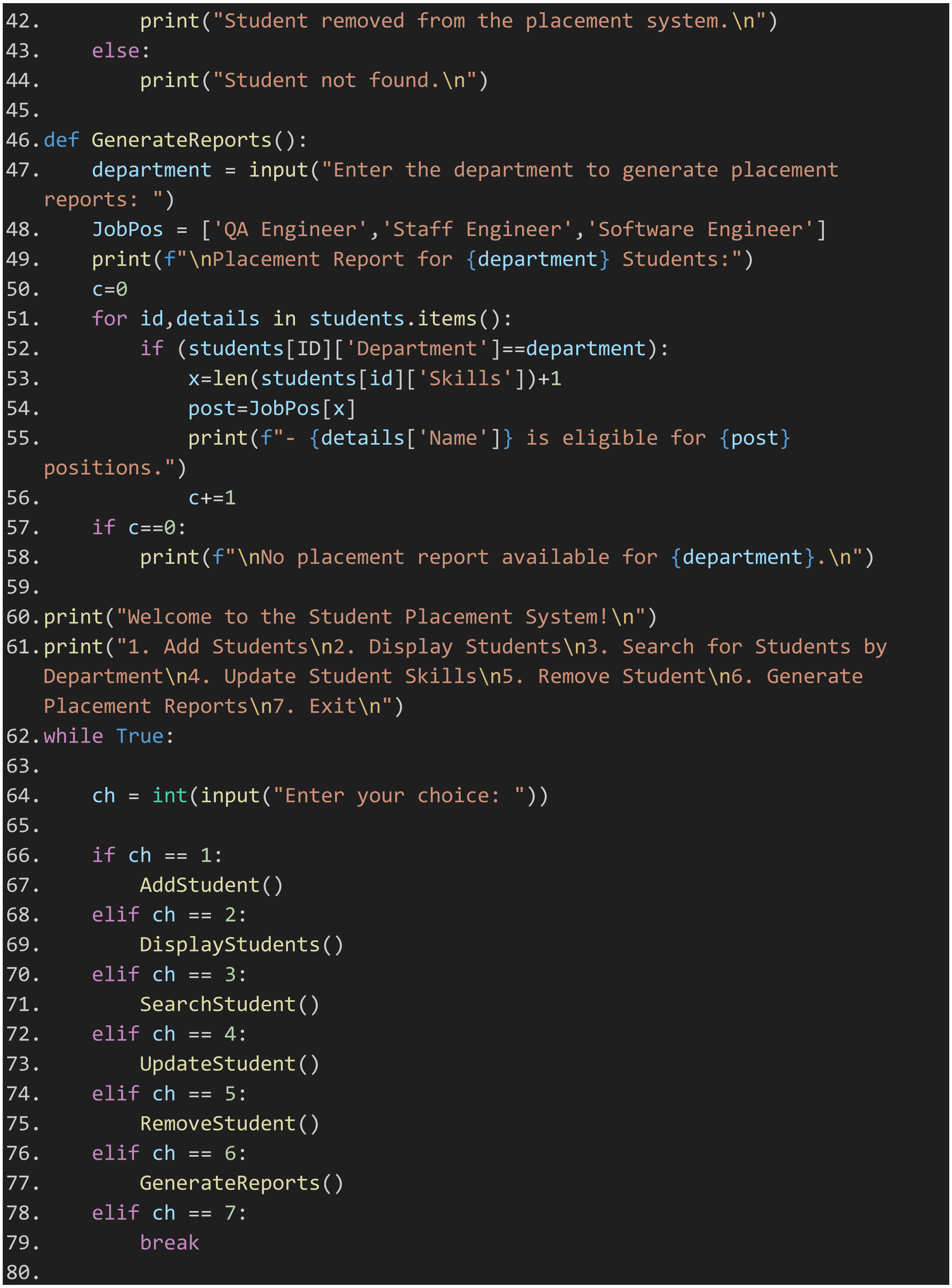
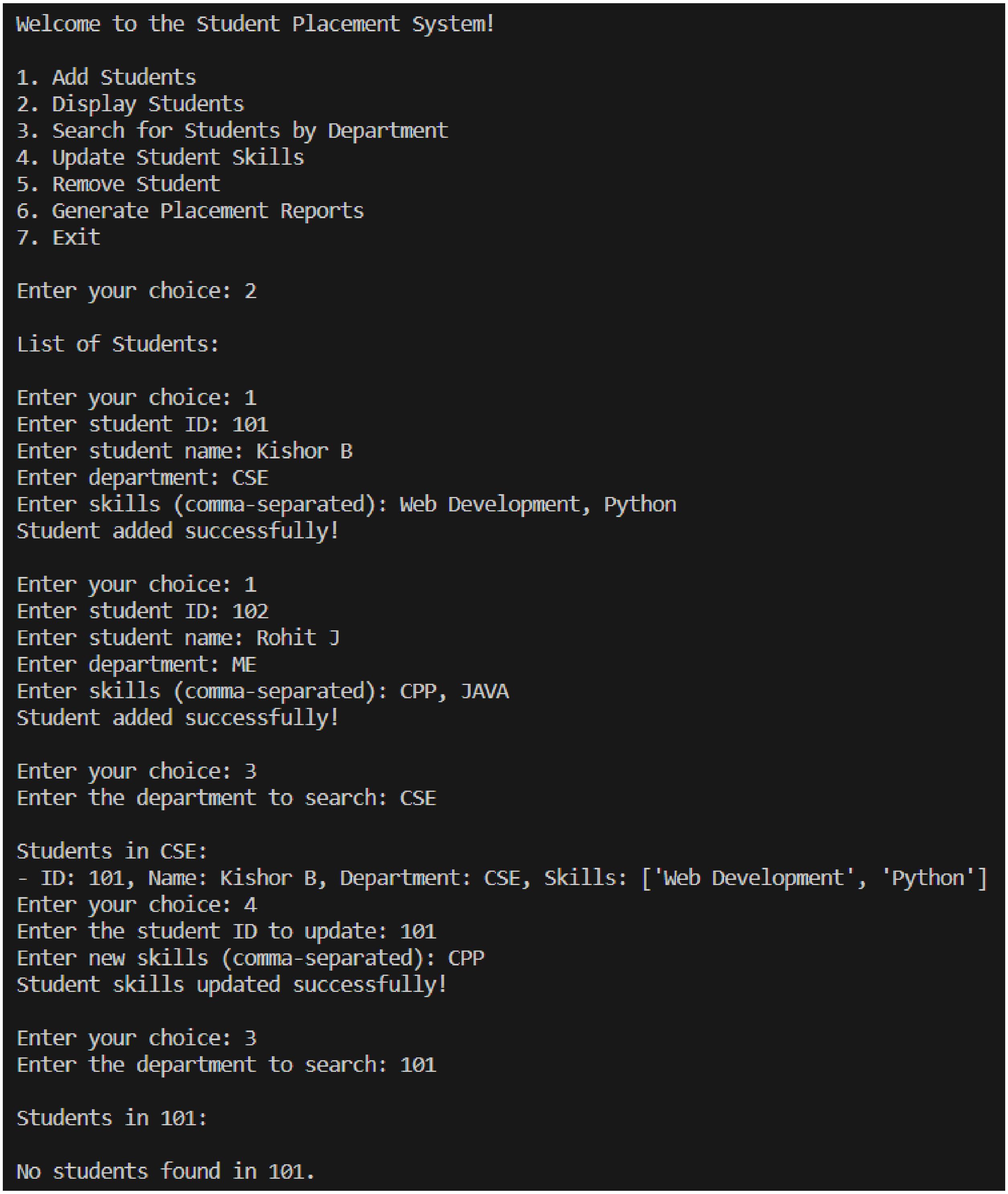
1. Student Placement System

```
students = {}

def AddStudent():
    ID = input("Enter student ID: ")
    name = input("Enter student name: ")
    department = input("Enter department: ")
    skills = input("Enter skills (comma-separated): ").split(', ')
    students[ID] = {'Name': name, 'Department': department, 'Skills': skills}
    print("Student added successfully!\n")

def DisplayStudents():
    print("\nList of Students:")
    for ID, details in students.items():
        print(f"ID: {ID}, Name: {details['Name']}, Department: {details['Department']}, Skills: {details['Skills']}")
    print()

def SearchStudent():
    department = input("Enter the department to search: ")
    print(f"\nStudents in {department}:")
    c=0
    for id,details in students.items():
            if(students[id]['Department']==department):
                print(f"- ID: {id}, Name: {details['Name']}, Department: {details['Department']}, Skills: {details['Skills']}")
                c+=1
    if c==0:
        print(f"\nNo students found in {department}.\n")

def UpdateStudent():
    ID = input("Enter the student ID to update: ")
    if ID in students:
        nskills = input("Enter new skills (comma-separated): ").split(', ')
        students[ID]['Skills'] = nskills
        print("Student skills updated successfully!\n")
    else:
        print("Student not found.\n")

def RemoveStudent():
    ID = input("Enter the student ID to remove: ")
    if ID in students:
        del students[ID]
```

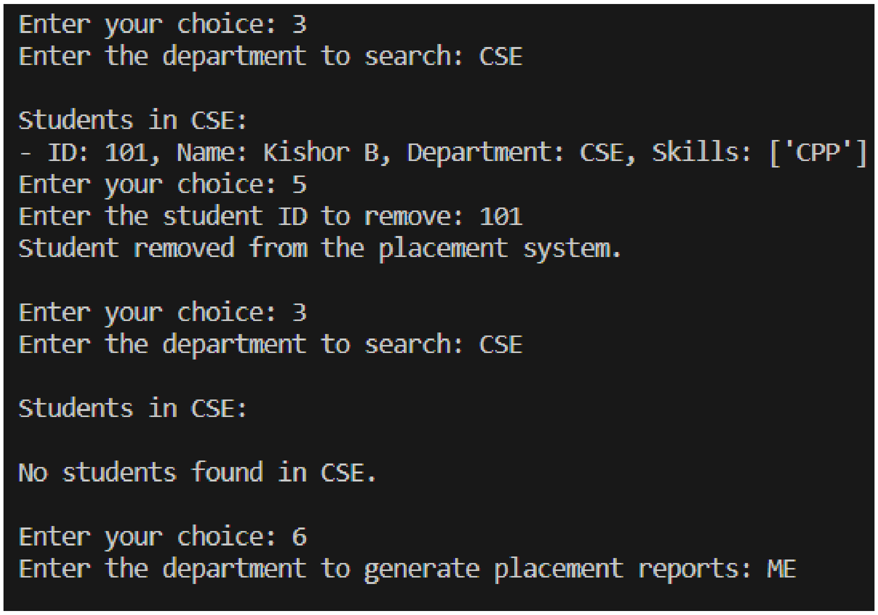


```
        print("Student removed from the placement system.\n")
    else:
        print("Student not found.\n")

def GenerateReports():
    department = input("Enter the department to generate placement reports: ")
    JobPos = ['QA Engineer','Staff Engineer','Software Engineer']
    print(f"\nPlacement Report for {department} Students:")
    c=0
    for id,details in students.items():
        if (students[ID]['Department']==department):
            x=len(students[id]['Skills'])+1
            post=JobPos[x]
            print(f"- {details['Name']} is eligible for {post} positions.")
            c+=1
    if c==0:
        print(f"\nNo placement report available for {department}.\n")

print("Welcome to the Student Placement System!\n")
print("1. Add Students\n2. Display Students\n3. Search for Students by Department\n4. Update Student Skills\n5. Remove Student\n6. Generate Placement Reports\n7. Exit\n")
while True:

    ch = int(input("Enter your choice: "))

    if ch == 1:
        AddStudent()
    elif ch == 2:
        DisplayStudents()
    elif ch == 3:
        SearchStudent()
    elif ch == 4:
        UpdateStudent()
    elif ch == 5:
        RemoveStudent()
    elif ch == 6:
        GenerateReports()
    elif ch == 7:
        break

```

Outputs:



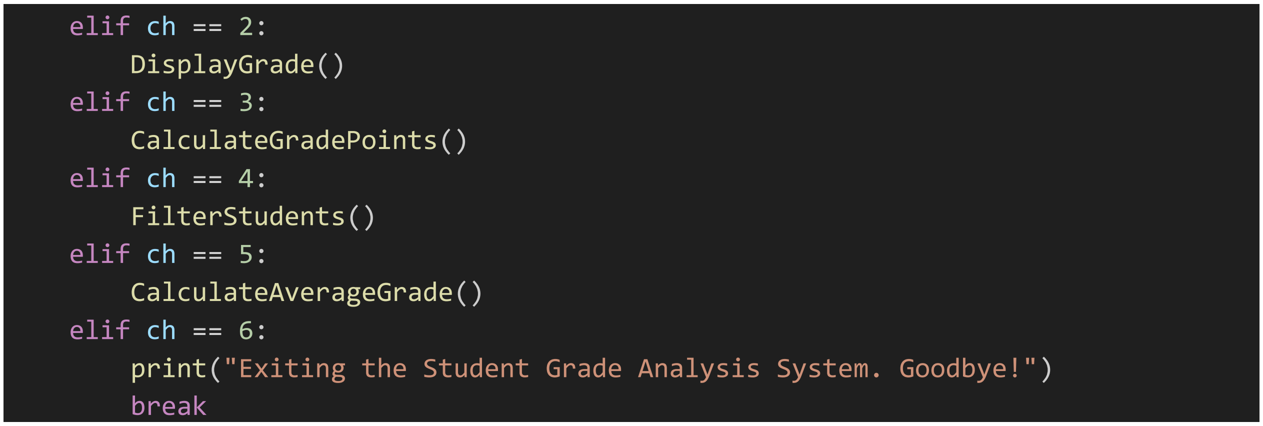
```
Welcome to the Student Placement System!

1. Add Students
2. Display Students
3. Search for Students by Department
4. Update Student Skills
5. Remove Student
6. Generate Placement Reports
7. Exit

Enter your choice: 2

List of Students:

Enter your choice: 1
Enter student ID: 101
Enter student name: Kishor B
Enter department: CSE
Enter skills (comma-separated): Web Development, Python
Student added successfully!

Enter your choice: 1
Enter student ID: 102
Enter student name: Rohit J
Enter department: ME
Enter skills (comma-separated): CPP, JAVA
Student added successfully!

Enter your choice: 3
Enter the department to search: CSE

Students in CSE:
- ID: 101, Name: Kishor B, Department: CSE, Skills: ['Web Development', 'Python']
Enter your choice: 4
Enter the student ID to update: 101
Enter new skills (comma-separated): CPP
Student skills updated successfully!

Enter your choice: 3
Enter the department to search: 101

Students in 101:

No students found in 101.
```



```
Enter your choice: 3
Enter the department to search: CSE

Students in CSE:
- ID: 101, Name: Kishor B, Department: CSE, Skills: ['CPP']
Enter your choice: 5
Enter the student ID to remove: 101
Student removed from the placement system.

Enter your choice: 3
Enter the department to search: CSE

Students in CSE:

No students found in CSE.

Enter your choice: 6
Enter the department to generate placement reports: ME
```

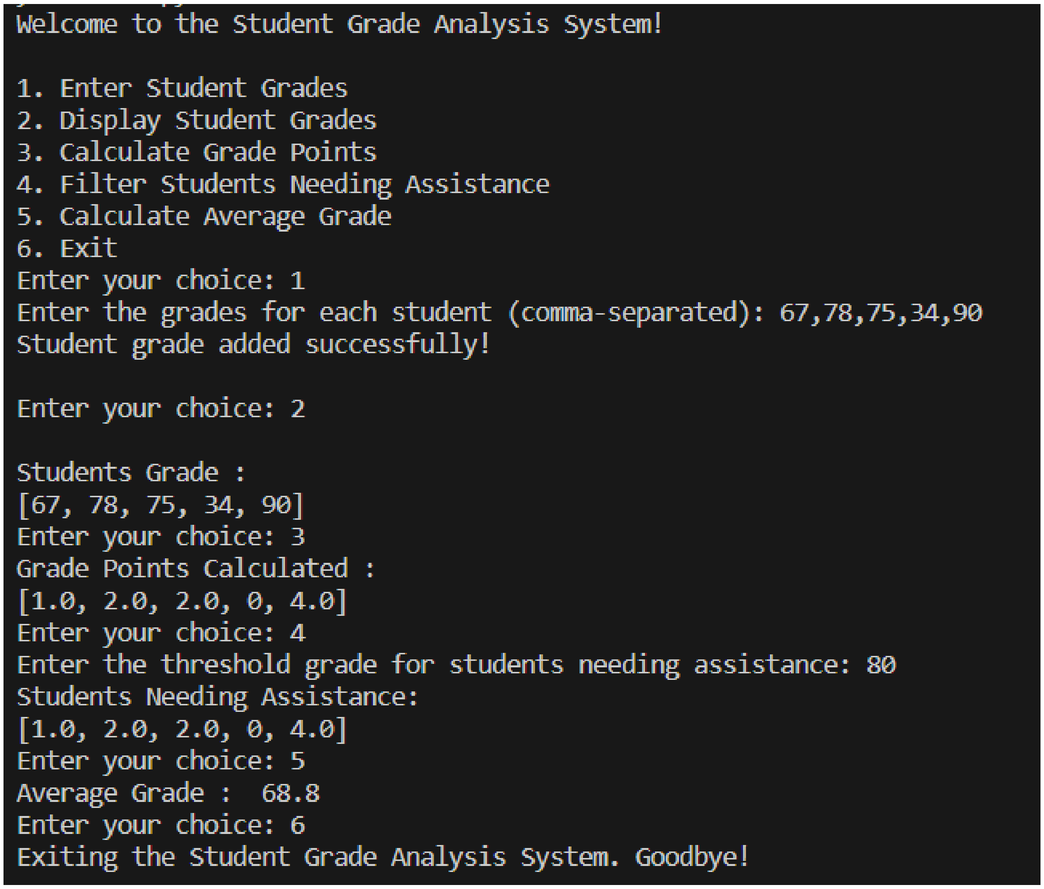
2. Grade System

```
from functools import reduce
grade = []
point=[]
def AddGrade():
    global grade
    grade=list(map(int,input("Enter the grades for each student (comma-separated): ").split(",")))

    print("Student grade added successfully!\n")

def DisplayGrade():
    print("\nStudents Grade :")
    print(grade)

def CalculateGradePoints():
    global point
    print("Grade Points Calculated : ")
    point=list(map(lambda x:5.0  if x==100 else (4.0 if x>=90 else (3.0 if x>=80 else (2.0 if x>=70 else (1.0 if x>=60 else 0)))) ,grade))
    print(point)

def FilterStudents():
    th=float(input("Enter the threshold grade for students needing assistance: "))
    print("Students Needing Assistance:")
    n=list(filter(lambda x: x<th ,point))
    print(n)

def CalculateAverageGrade():
    avg=reduce(lambda x,y:x+y,grade)/len(grade)
    print("Average Grade : ",avg)



print("Welcome to the Student Grade Analysis System!\n")
print("1. Enter Student Grades\n2. Display Student Grades\n3. Calculate Grade Points\n4. Filter Students Needing Assistance\n5. Calculate Average Grade\n6. Exit")
while True:

    ch = int(input("Enter your choice: "))

    if ch == 1:
        AddGrade()
```



```
    elif ch == 2:
        DisplayGrade()
    elif ch == 3:
        CalculateGradePoints()
    elif ch == 4:
        FilterStudents()
    elif ch == 5:
        CalculateAverageGrade()
    elif ch == 6:
        print("Exiting the Student Grade Analysis System. Goodbye!")
        break
```

Outputs:

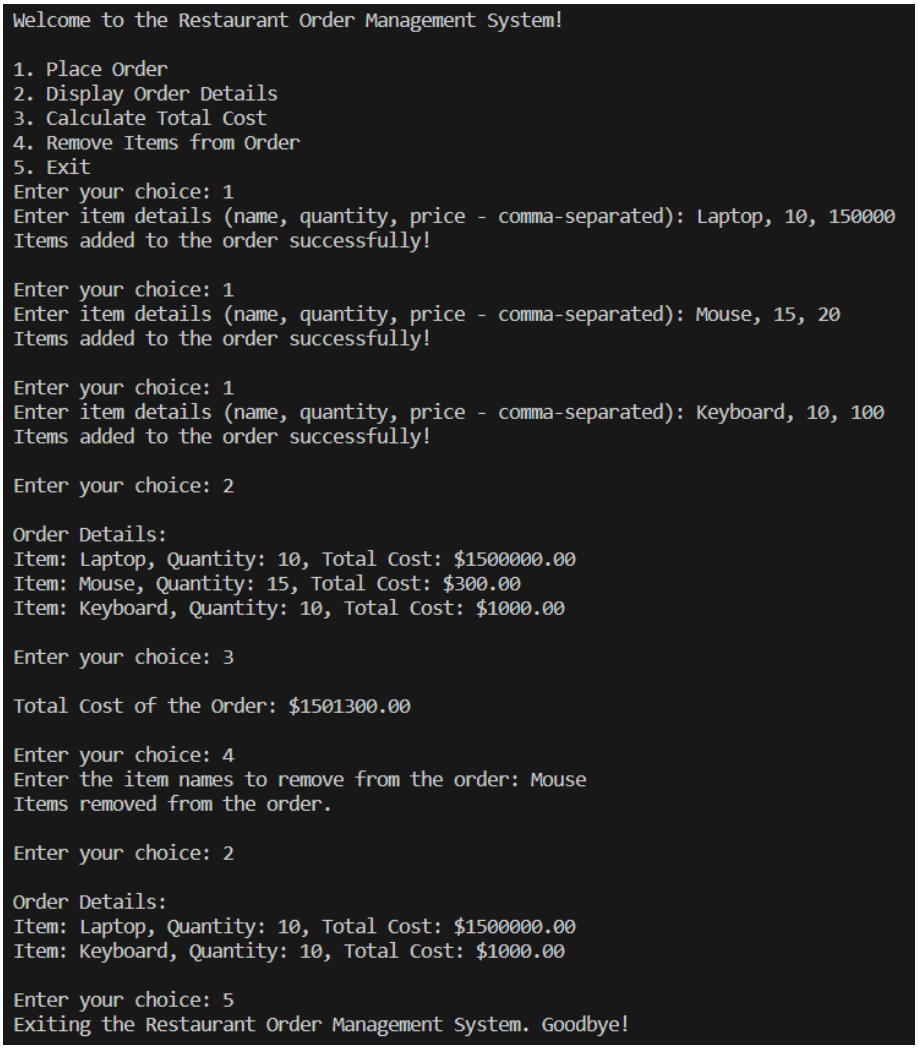


```
Welcome to the Student Grade Analysis System!

1. Enter Student Grades
2. Display Student Grades
3. Calculate Grade Points
4. Filter Students Needing Assistance
5. Calculate Average Grade
6. Exit
Enter your choice: 1
Enter the grades for each student (comma-separated): 67,78,75,34,90
Student grade added successfully!

Enter your choice: 2

Students Grade :
[67, 78, 75, 34, 90]
Enter your choice: 3
Grade Points Calculated :
[1.0, 2.0, 2.0, 0, 4.0]
Enter your choice: 4
Enter the threshold grade for students needing assistance: 80
Students Needing Assistance:
[1.0, 2.0, 2.0, 0, 4.0]
Enter your choice: 5
Average Grade :  68.8
Enter your choice: 6
Exiting the Student Grade Analysis System. Goodbye!
```

3. Order Management

```python
order=[]
def PlaceOrder(*items):
    for item in items:
        name, quantity, price = map(str, item.split(','))
        quantity = int(quantity)
        price = float(price)
        order.append({'name': name, 'quantity': quantity, 'price': price})
    print("Items added to the order successfully!\n")


def DisplayOrderDetails():
    print("\nOrder Details:")
    for item in order:
        total_cost = item['quantity'] * item['price']
        print(f"Item: {item['name']}, Quantity: {item['quantity']}, Total Cost: ${total_cost:.2f}")
    print()


def CalculateTotalCost():
    total=0
    for i in order:
        total+=(i['price']*i['quantity'])
    print(f"\nTotal Cost of the Order: ${total:.2f}\n")


def RemoveItems(*items):
    global order
    order = [item for item in order if item['name'] not in items]
    print("Items removed from the order.\n")




print("Welcome to the Restaurant Order Management System!\n")
print("1. Place Order\n2. Display Order Details\n3. Calculate Total Cost\n4. Remove Items from Order\n5. Exit")
while True:

    ch = int(input("Enter your choice: "))

    if ch == 1:
        items_input = input("Enter item details (name, quantity, price - comma-separated): ")
        PlaceOrder(items_input)
    elif ch == 2:
        DisplayOrderDetails()
    elif ch == 3:
        CalculateTotalCost()
    elif ch == 4:
        items_to_remove = input("Enter the item names to remove from the order:
")
        RemoveItems(items_to_remove)
    elif ch == 5:
        print("Exiting the Restaurant Order Management System. Goodbye!")
        break
```

Outputs:



```
Welcome to the Restaurant Order Management System!

1. Place Order
2. Display Order Details
3. Calculate Total Cost
4. Remove Items from Order
5. Exit
Enter your choice: 1
Enter item details (name, quantity, price - comma-separated): Laptop, 10, 150000
Items added to the order successfully!

Enter your choice: 1
Enter item details (name, quantity, price - comma-separated): Mouse, 15, 20
Items added to the order successfully!

Enter your choice: 1
Enter item details (name, quantity, price - comma-separated): Keyboard, 10, 100
Items added to the order successfully!

Enter your choice: 2

Order Details:
Item: Laptop, Quantity: 10, Total Cost: $1500000.00
Item: Mouse, Quantity: 15, Total Cost: $300.00
Item: Keyboard, Quantity: 10, Total Cost: $1000.00

Enter your choice: 3

Total Cost of the Order: $1501300.00

Enter your choice: 4
Enter the item names to remove from the order: Mouse
Items removed from the order.

Enter your choice: 2

Order Details:
Item: Laptop, Quantity: 10, Total Cost: $1500000.00
Item: Keyboard, Quantity: 10, Total Cost: $1000.00

Enter your choice: 5
Exiting the Restaurant Order Management System. Goodbye!
```