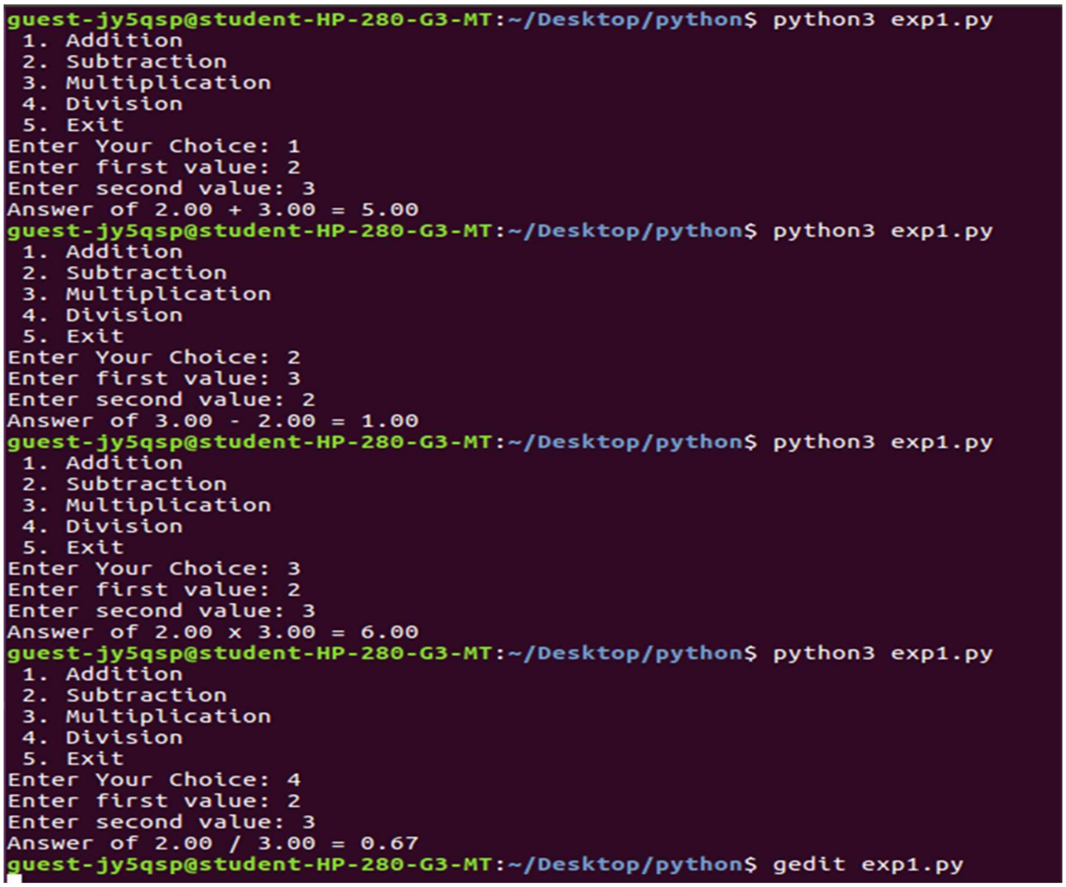
Code-

```
#exp-1
print(" 1. Addition\n 2. Subtraction\n 3. Multiplication\n 4. Division\n 5. Exit")
ch = int(input("Enter Your Choice: "))

a  = float(input("Enter first value: "))
b  = float(input("Enter second value: "))

if ch == 1:
        print("Answer of %.2f + %.2f = %.2f"%(a , b, a+b))

elif ch==2:
        print("Answer of %.2f - %.2f = %.2f"%(a, b, a-b))

elif ch==3:
        print("Answer of %.2f x %.2f = %.2f"%(a, b, a*b))

elif ch==4:
        print("Answer of %.2f / %.2f = %.2f"%(a, b, a/b))

else:
        print("BYE")
```

OUTPUT-

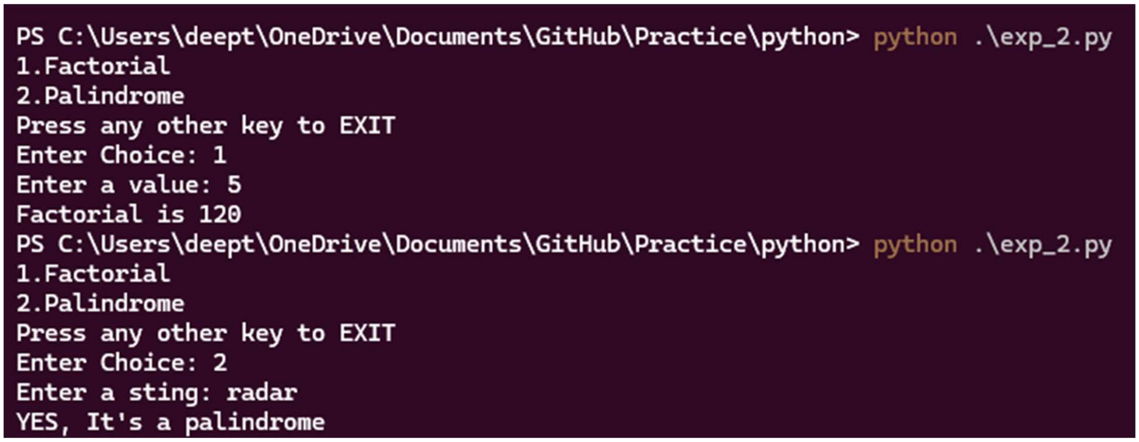


```
guest-jy5qsp@student-HP-280-G3-MT:~/Desktop/python$ python3 exp1.py
 1. Addition
 2. Subtraction
 3. Multiplication
 4. Division
 5. Exit
Enter Your Choice: 1
Enter first value: 2
Enter second value: 3
Answer of 2.00 + 3.00 = 5.00
guest-jy5qsp@student-HP-280-G3-MT:~/Desktop/python$ python3 exp1.py
 1. Addition
 2. Subtraction
 3. Multiplication
 4. Division
 5. Exit
Enter Your Choice: 2
Enter first value: 3
Enter second value: 2
Answer of 3.00 - 2.00 = 1.00
guest-jy5qsp@student-HP-280-G3-MT:~/Desktop/python$ python3 exp1.py
 1. Addition
 2. Subtraction
 3. Multiplication
 4. Division
 5. Exit
Enter Your Choice: 3
Enter first value: 2
Enter second value: 3
Answer of 2.00 x 3.00 = 6.00
guest-jy5qsp@student-HP-280-G3-MT:~/Desktop/python$ python3 exp1.py
 1. Addition
 2. Subtraction
 3. Multiplication
 4. Division
 5. Exit
Enter Your Choice: 4
Enter first value: 2
Enter second value: 3
Answer of 2.00 / 3.00 = 0.67
guest-jy5qsp@student-HP-280-G3-MT:~/Desktop/python$ gedit exp1.py
```

Code-

```
#exp-2
print("1.Factorial")
print("2.Palindrome")
print("Press any other key to EXIT")

ch = int(input("Enter Choice: "))

if ch == 1:
    fact = 1
    n = int(input("Enter a value: "))

    while n != 1:
        fact = fact * n
        n = n - 1

    print("Factorial is",fact)

elif ch == 2:
    str1 = input("Enter a sting: ")

    if str1 == str1[::-1]:
        print("YES, It's a palindrome")
    else:
        print("NO, It's not a palindrome")

else:
    print("BYE")
```

OUTPUT-



```
PS C:\Users\deept\OneDrive\Documents\GitHub\Practice\python> python .\exp_2.py
1.Factorial
2.Palindrome
Press any other key to EXIT
Enter Choice: 1
Enter a value: 5
Factorial is 120
PS C:\Users\deept\OneDrive\Documents\GitHub\Practice\python> python .\exp_2.py
1.Factorial
2.Palindrome
Press any other key to EXIT
Enter Choice: 2
Enter a sting: radar
YES, It's a palindrome
```

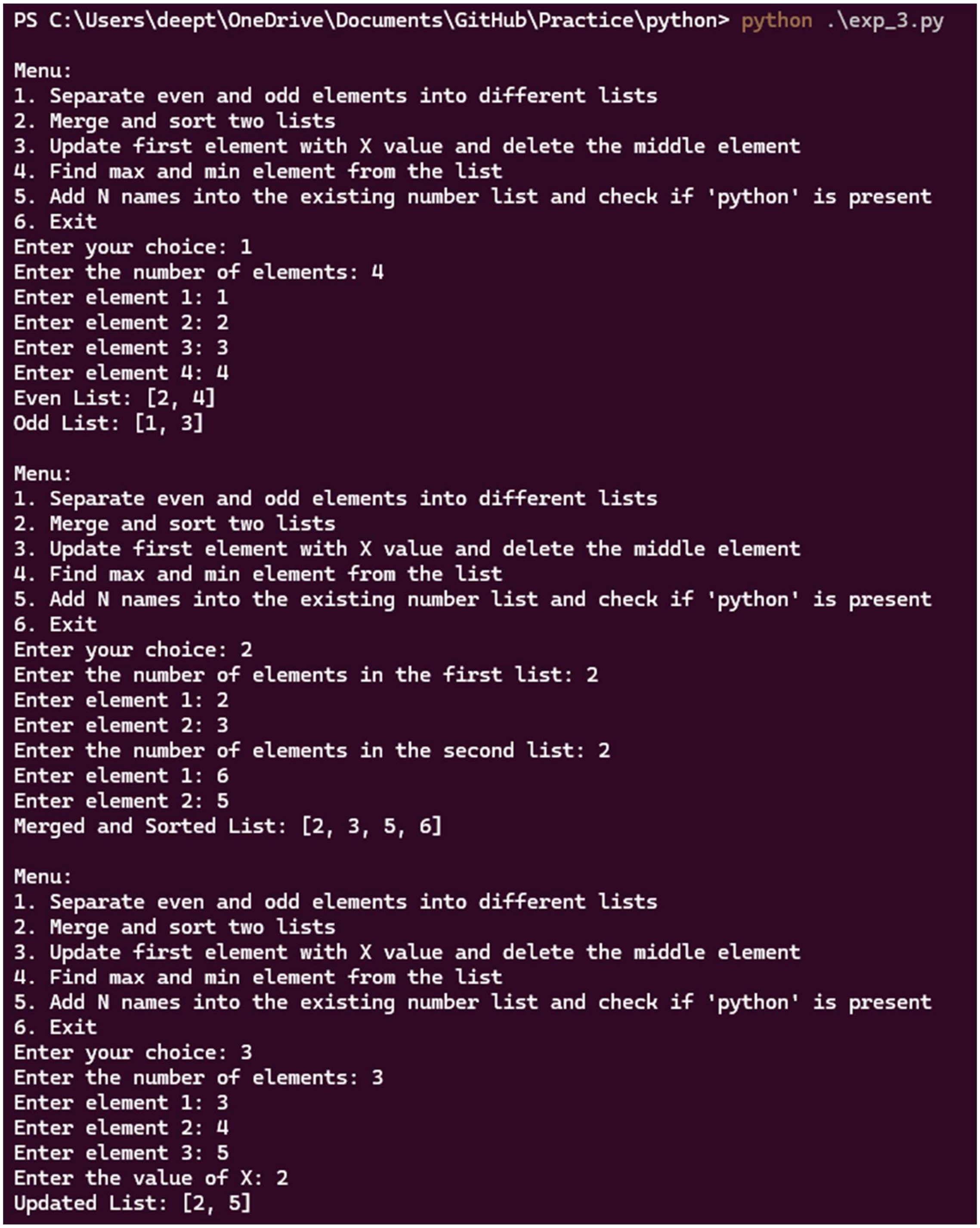
Code-

```
#exp-3
def separate_even_odd(lst):
    even_list = []
    odd_list = []
    for num in lst:
        if num % 2 == 0:
            even_list.append(num)
        else:
            odd_list.append(num)
    return even_list, odd_list

def merge_and_sort(even_list, odd_list):
    merged_list = even_list + odd_list
    merged_list.sort()
    return merged_list

def update_and_delete(lst, x):
    if len(lst) > 0:
        lst[0] = x
        if len(lst) % 2 == 0:
            middle_index = len(lst) // 2 - 1
        else:
            middle_index = len(lst) // 2
        lst.pop(middle_index)
    return lst

def find_max_min(lst):
    if len(lst) > 0:
        return max(lst), min(lst)
    else:
        return None, None

def add_names_and_check(lst, names):
    lst.extend(names)
    return 'python' in [name.lower() for name in lst]

def main():
    while True:
        print("\nMenu:")
        print("1. Separate even and odd elements into different lists")
        print("2. Merge and sort two lists")
        print("3. Update first element with X value and delete the middle element")
        print("4. Find max and min element from the list")
        print("5. Add N names into the existing number list and check if 'python' is
present")
        print("6. Exit")

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

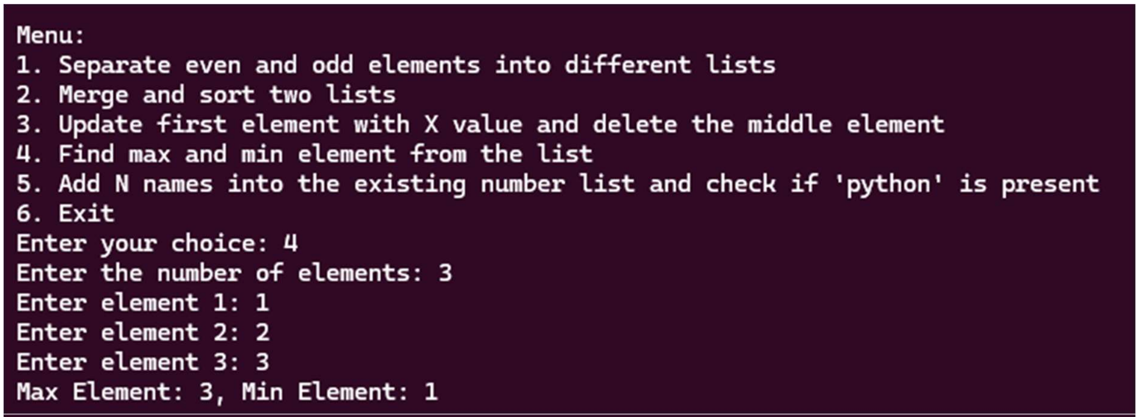
```
if choice == 1:
    n = int(input("Enter the number of elements: "))
    lst = []
    for i in range(n):
        element = int(input(f"Enter element {i+1}: "))
        lst.append(element)
    even_list, odd_list = separate_even_odd(lst)
    print("Even List:", even_list)
    print("Odd List:", odd_list)

elif choice == 2:
    n1 = int(input("Enter the number of elements in the first list: "))
    list1 = [int(input(f"Enter element {i+1}: ")) for i in range(n1)]
    n2 = int(input("Enter the number of elements in the second list: "))
    list2 = [int(input(f"Enter element {i+1}: ")) for i in range(n2)]
    merged_sorted_list = merge_and_sort(list1, list2)
    print("Merged and Sorted List:", merged_sorted_list)

elif choice == 3:
    n = int(input("Enter the number of elements: "))
    lst = [int(input(f"Enter element {i+1}: ")) for i in range(n)]
    x = int(input("Enter the value of X: "))
    updated_list = update_and_delete(lst, x)
    print("Updated List:", updated_list)

elif choice == 4:
    n = int(input("Enter the number of elements: "))
    lst = [int(input(f"Enter element {i+1}: ")) for i in range(n)]
    max_val, min_val = find_max_min(lst)
    if max_val is not None:
        print(f"Max Element: {max_val}, Min Element: {min_val}")
    else:
        print("List is empty")

elif choice == 5:
    n = int(input("Enter the number of elements: "))
    lst = [input(f"Enter element {i+1}: ") for i in range(n)]
    name_count = int(input("Enter the number of elements: "))
    names = [input(f"Enter name {i+1}: ") for i in range(name_count)]
    python_present = add_names_and_check(lst, names)
    if python_present:
        print("'python' is in the list.")
    else:
        print("'python' is NOT in the list.")

elif choice == 6:
    print("BYE")
    break

else:
```

```
            print("Invalid choice!")

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

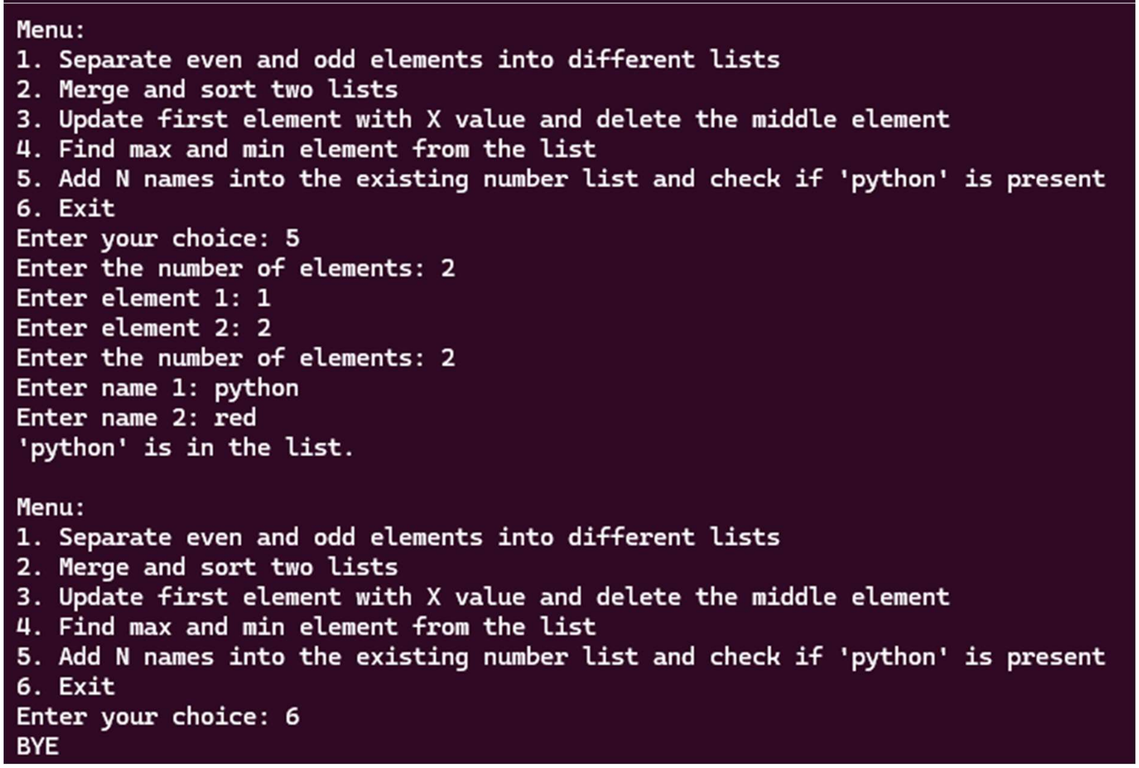
OUTPUT-



```
PS C:\Users\deept\OneDrive\Documents\GitHub\Practice\python> python .\exp_3.py

Menu:
1. Separate even and odd elements into different lists
2. Merge and sort two lists
3. Update first element with X value and delete the middle element
4. Find max and min element from the list
5. Add N names into the existing number list and check if 'python' is present
6. Exit
Enter your choice: 1
Enter the number of elements: 4
Enter element 1: 1
Enter element 2: 2
Enter element 3: 3
Enter element 4: 4
Even List: [2, 4]
Odd List: [1, 3]

Menu:
1. Separate even and odd elements into different lists
2. Merge and sort two lists
3. Update first element with X value and delete the middle element
4. Find max and min element from the list
5. Add N names into the existing number list and check if 'python' is present
6. Exit
Enter your choice: 2
Enter the number of elements in the first list: 2
Enter element 1: 2
Enter element 2: 3
Enter the number of elements in the second list: 2
Enter element 1: 6
Enter element 2: 5
Merged and Sorted List: [2, 3, 5, 6]

Menu:
1. Separate even and odd elements into different lists
2. Merge and sort two lists
3. Update first element with X value and delete the middle element
4. Find max and min element from the list
5. Add N names into the existing number list and check if 'python' is present
6. Exit
Enter your choice: 3
Enter the number of elements: 3
Enter element 1: 3
Enter element 2: 4
Enter element 3: 5
Enter the value of X: 2
Updated List: [2, 5]
```



```
Menu:
1. Separate even and odd elements into different lists
2. Merge and sort two lists
3. Update first element with X value and delete the middle element
4. Find max and min element from the list
5. Add N names into the existing number list and check if 'python' is present
6. Exit
Enter your choice: 4
Enter the number of elements: 3
Enter element 1: 1
Enter element 2: 2
Enter element 3: 3
Max Element: 3, Min Element: 1
```

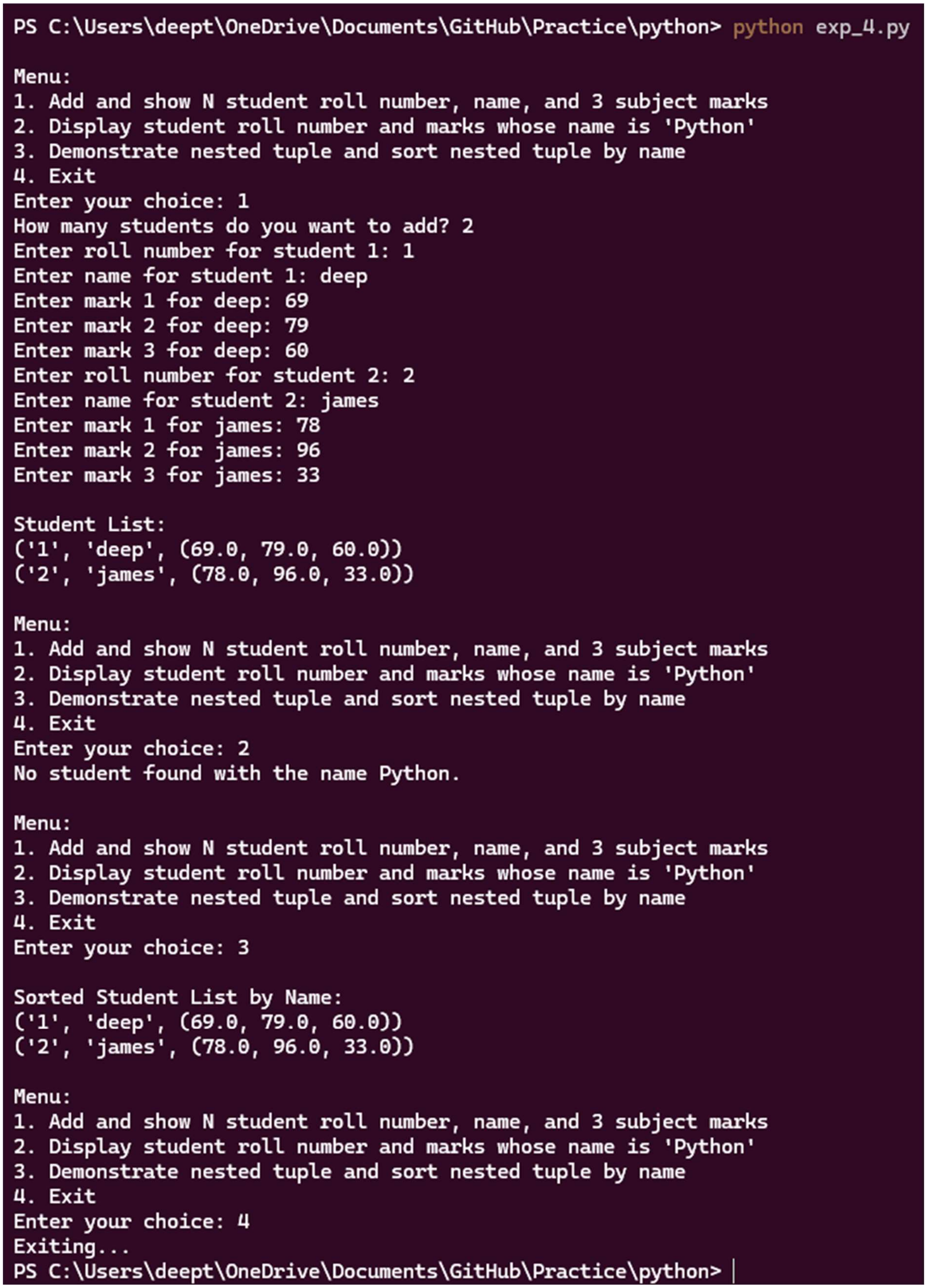


```
Menu:
1. Separate even and odd elements into different lists
2. Merge and sort two lists
3. Update first element with X value and delete the middle element
4. Find max and min element from the list
5. Add N names into the existing number list and check if 'python' is present
6. Exit
Enter your choice: 5
Enter the number of elements: 2
Enter element 1: 1
Enter element 2: 2
Enter the number of elements: 2
Enter name 1: python
Enter name 2: red
'python' is in the list.

Menu:
1. Separate even and odd elements into different lists
2. Merge and sort two lists
3. Update first element with X value and delete the middle element
4. Find max and min element from the list
5. Add N names into the existing number list and check if 'python' is present
6. Exit
Enter your choice: 6
BYE
```

Code-

```
#exp-4
def add_students(student_list, n):
    for i in range(n):
        roll_no = input(f"Enter roll number for student {i+1}: ")
        name = input(f"Enter name for student {i+1}: ")
        marks = []
        for j in range(1, 4):
            mark = float(input(f"Enter mark {j} for {name}: "))
            marks.append(mark)
        student_list.append((roll_no, name, tuple(marks)))
    return student_list

def display_student_by_name(student_list, search_name):
    found = False
    for student in student_list:
        roll_no, name, marks = student
        if name.lower() == search_name.lower():
            print(f"Roll Number: {roll_no}, Marks: {marks}")
            found = True
    if not found:
        print(f"No student found with the name {search_name}.")

def sort_students_by_name(nested_tuple):
    return sorted(nested_tuple, key=lambda x: x[1].lower())

def main():
    student_list = []

    while True:
        print("\nMenu:")
        print("1. Add and show N student roll number, name, and 3 subject marks")
        print("2. Display student roll number and marks whose name is 'Python'")
        print("3. Demonstrate nested tuple and sort nested tuple by name")
        print("4. Exit")

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

        if choice == 1:
            n = int(input("How many students do you want to add? "))
            student_list = add_students(student_list, n)
            print("\nStudent List:")
            for student in student_list:
                print(student)

        elif choice == 2:
            search_name = "Python"
            display_student_by_name(student_list, search_name)

        elif choice == 3:
            sorted_students = sort_students_by_name(student_list)
            print("\nSorted Student List by Name:")
            for student in sorted_students:
                print(student)

        elif choice == 4:
            print("Exiting...")
            break

        else:
            print("Invalid choice! Please choose again.")

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

OUTPUT-

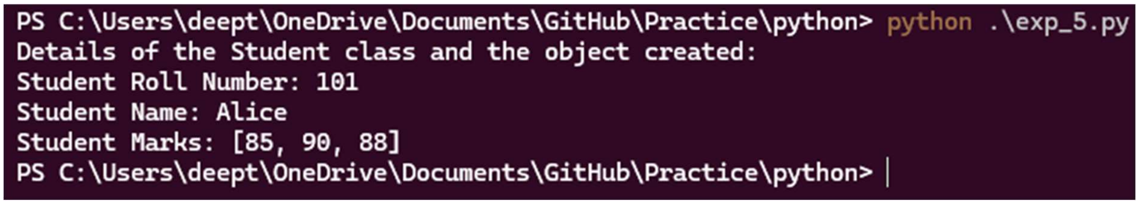


```
PS C:\Users\deept\OneDrive\Documents\GitHub\Practice\python> python exp_4.py

Menu:
1. Add and show N student roll number, name, and 3 subject marks
2. Display student roll number and marks whose name is 'Python'
3. Demonstrate nested tuple and sort nested tuple by name
4. Exit
Enter your choice: 1
How many students do you want to add? 2
Enter roll number for student 1: 1
Enter name for student 1: deep
Enter mark 1 for deep: 69
Enter mark 2 for deep: 79
Enter mark 3 for deep: 60
Enter roll number for student 2: 2
Enter name for student 2: james
Enter mark 1 for james: 78
Enter mark 2 for james: 96
Enter mark 3 for james: 33

Student List:
('1', 'deep', (69.0, 79.0, 60.0))
('2', 'james', (78.0, 96.0, 33.0))

Menu:
1. Add and show N student roll number, name, and 3 subject marks
2. Display student roll number and marks whose name is 'Python'
3. Demonstrate nested tuple and sort nested tuple by name
4. Exit
Enter your choice: 2
No student found with the name Python.

Menu:
1. Add and show N student roll number, name, and 3 subject marks
2. Display student roll number and marks whose name is 'Python'
3. Demonstrate nested tuple and sort nested tuple by name
4. Exit
Enter your choice: 3

Sorted Student List by Name:
('1', 'deep', (69.0, 79.0, 60.0))
('2', 'james', (78.0, 96.0, 33.0))

Menu:
1. Add and show N student roll number, name, and 3 subject marks
2. Display student roll number and marks whose name is 'Python'
3. Demonstrate nested tuple and sort nested tuple by name
4. Exit
Enter your choice: 4
Exiting...
PS C:\Users\deept\OneDrive\Documents\GitHub\Practice\python>
```

Code-

```
#exp-5
class Student:
    def __init__(self, roll_no, name, marks):

        self.roll_no = roll_no
        self.name = name
        self.marks = marks

    def display_details(self):
        print(f"Student Roll Number: {self.roll_no}")
        print(f"Student Name: {self.name}")
        print(f"Student Marks: {self.marks}")

student1 = Student(101, "Alice", [85, 90, 88])

print("Details of the Student class and the object created:")
student1.display_details()
```

OUTPUT-



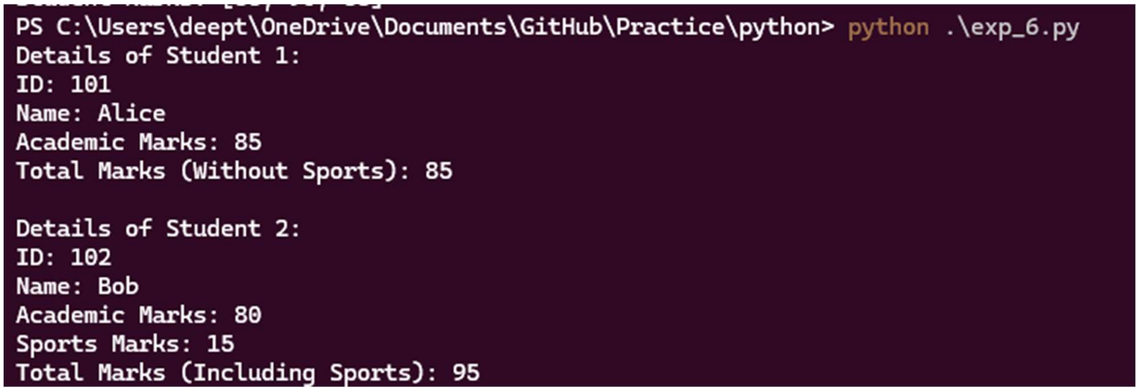
```
PS C:\Users\deept\OneDrive\Documents\GitHub\Practice\python> python .\exp_5.py
Details of the Student class and the object created:
Student Roll Number: 101
Student Name: Alice
Student Marks: [85, 90, 88]
PS C:\Users\deept\OneDrive\Documents\GitHub\Practice\python>
```

Code-

#exp-6

```
class Person:
    def __init__(self, id, name):
        self.id = id
        self.name = name

    def display_details(self):
        print(f"ID: {self.id}")
        print(f"Name: {self.name}")

class Student(Person):
    def __init__(self, id, name, academic_marks, sports_marks=None):
        super().__init__(id, name)
        self.academic_marks = academic_marks
        self.sports_marks = sports_marks

    def display_details(self):
        super().display_details()
        print(f"Academic Marks: {self.academic_marks}")
        if self.sports_marks is not None:
            print(f"Sports Marks: {self.sports_marks}")
            total_marks = self.academic_marks + self.sports_marks
            print(f"Total Marks (Including Sports): {total_marks}")
        else:
            print(f"Total Marks (Without Sports): {self.academic_marks}")

def main():
    student1 = Student(101, "Alice", 85)
    print("Details of Student 1:")
    student1.display_details()
    student2 = Student(102, "Bob", 80, 15)
    print("\nDetails of Student 2:")
    student2.display_details()

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

OUTPUT-



```
PS C:\Users\deept\OneDrive\Documents\GitHub\Practice\python> python .\exp_6.py
Details of Student 1:
ID: 101
Name: Alice
Academic Marks: 85
Total Marks (Without Sports): 85

Details of Student 2:
ID: 102
Name: Bob
Academic Marks: 80
Sports Marks: 15
Total Marks (Including Sports): 95
```