```
'''
In Japan ,there was a very huge Tsunami. Millions and millions worth buildings and properties were destroyed. Many people lost
Most of them were injured and few were safe. A news reporter arrives to the spot to take the current survey regarding the situ
dead and injured. He wanted to publish it in the newspaper and ask the other countries to help the affected people.
'''
dead = int(input("Enter the number of people dead "))
injured = int(input("Enter the number of people injured "))
safe = int(input("Enter the number of people safe "))
print(f'''TSUNAMI REPORT OF JAPAN
  The number of people
  1)Dead: {dead}
  2)Injured: {injured}
  3)Safe: {safe}
  Please help the people who are suffering!!!'''
)
```

```
Enter the number of people dead 2000
Enter the number of people injured 3000
Enter the number of people safe 10000
TSUNAMI REPORT OF JAPAN
  The number of people
  1)Dead: 2000
  2)Injured: 3000
  3)Safe: 10000
  Please help the people who are suffering!!!
```

```
'''
Sumanth having N number of elements and he decided to create a set for it.and after creating set he wants to add some elements
remove elements to set and find the difference between two sets, and he wants to know the intersection elements in two sets.an
will be shown in input and output format specifications.
'''

set1 = set(input("Please Enter the set-1 elements :: ").split())
set2 = set(input("Please Enter the set-2 elements :: ").split())

print("choose one option: 1) add \n 2) remove \n 3) difference \n 4) intersection\n")
choice = int(input().strip())

print("choose set: 1) set1 2) set2")
set_choice = int(input().strip())

# perform operation
if choice == 1:
    element = input().strip()
    if set_choice == 1:
        set1.add(element)
    else:
        set2.add(element)

elif choice == 2:
    element = input().strip()
    if set_choice == 1:
        set1.discard(element)
    else:
        set2.discard(element)

elif choice == 3:
    if set_choice == 1:
        result = set1.difference(set2)
    else:
        result = set2.difference(set1)
    print("set(" + str(sorted(list(result))) + ")")

elif choice == 4:
    result = set1.intersection(set2)
    print("set(" + str(sorted(list(result))) + ")")

else:
    print("invalid choice")

if choice in [1, 2]:
    if set_choice == 1:
        print("set(" + str(sorted(list(set1))) + ")")
    else:
        print("set(" + str(sorted(list(set2))) + ")")
```

```
Please Enter the set-1 elements :: mango,apple,samsung
Please Enter the set-2 elements :: mango,apple,samsung
choose one option: 1) add
 2) remove
```

```
 3) difference
 4) intersection

1
choose set: 1) set1 2) set2
1
orange
set(['mango,apple,samsung', 'orange'])
```

```
'''
Newton’s method (also known as the Newton–Raphson method),  is a method for finding successively better approximations to the
of a real-valued function. The algorithm is first in the class of Householder’s methods, succeeded by Halley’s method. The met
extended to complex functions and to systems of equations.
Given a number, write a program to find the square root of a number using Newton’s Square Root.
'''

x = int(input("Enter the number whose square root is to be found"))
# Initialize the approximation to the square root and defining the accuracy
approx = x / 2
epsilon = 0.001
while abs(approx * approx - x) > epsilon:
    approx = (approx + x / approx) / 2

print("The square root of", x, "is approximately", approx)
```

```
Enter the number whose square root is to be found16
The square root of 16 is approximately 4.0000001858445895
```

```
'''
Ravi wants to familiar with some pattern programs. So help him to
Write a python code to read a number n from a user and print an inverted right
triangle star pattern of the desired size.
'''

n=int(input("Enter number of rows: "))
for i in range (n):
    print((n-i) * '*')
```

```
Enter number of rows: 4
****
***
**
*
```

```
'''
Write a Python program
To add new elements to the end of the list
To reverse elements in the list
To display same list elements multiple times
To concatenate two lists
To sort the elements in the list in ascending order
'''

list1 = [i for i in range(5)]
list1.append(int(input("Enter element to be appended to the list")))
list1.append(int(input("Enter element to be appended to the list")))

list1.reverse()

list2 = list1 * 3

list3 = list1 + list2

list3.sort()

print("Final list:", list3)
```

```
Enter element to be appended to the list12
Enter element to be appended to the list14
Final list: [0, 0, 0, 0, 1, 1, 1, 1, 2, 2, 2, 2, 3, 3, 3, 3, 4, 4, 4, 4, 12, 12, 12, 12, 14, 14, 14, 14]
```

✓ 18s completed at 20:19