```
# 1. Write a Python program that takes a list of elements as input and removes
# duplicate elements, while preserving the original order. Do not use the set()
# data structure.

lst = list(map(int, input().split()))
res = []
for i in lst:
    if i not in res:
        res.append(i)
print(res)
```

```
45
[45]
```

```
# 2. Write a Python program that takes a list of integers and finds the second largest
# element by traversing the list only once.
lst = list(map(int, input().split()))
first = second = float('-inf')
for i in lst:
    if i > first:
        second = first
        first = i
    elif first > i > second:
        second = i
print(second)
```

```
# 3. Write a Python program that takes a list and an integer k as input and rotates the
# list to the right by k positions.
lst = list(map(int, input().split()))
k = int(input())
k %= len(lst)
print(lst[-k:] + lst[:-k])
#
```

```
# 4. Write a Python program that takes a list of integers and a target sum, and finds all
# pairs of elements whose sum equals the target value.
lst = list(map(int, input().split()))
target = int(input())
pairs = []
for i in range(len(lst)):
    for j in range(i+1, len(lst)):
        if lst[i] + lst[j] == target:
            pairs.append((lst[i], lst[j]))
print(pairs)
```

```
# 5. Write a Python program that checks whether a given list of numbers is already
# sorted in ascending order.
lst = list(map(int, input().split()))
print(lst == sorted(lst))
```

```
# 6. Write a Python program that takes a nested list (which may contain lists inside lists
# at any depth) and flattens it into a single list.
def flatten(lst):
    res = []
    for i in lst:
        if isinstance(i, list):
            res.extend(flatten(i))
        else:
            res.append(i)
    return res

print(flatten(eval(input())))
```

```
# 7. Write a Python program that moves all zero elements to the end of the list while
# maintaining the relative order of the non-zero elements.
lst = list(map(int, input().split()))
res = [i for i in lst if i != 0] + [0] * lst.count(0)
print(res)
```

```
# 8. Write a Python program that counts the frequency of each element in a list and
# displays the result.
lst = input().split()
freq = {}
for i in lst:
    freq[i] = freq.get(i, 0) + 1
print(freq)
```

```
# 9. Write a Python program that finds the common elements between two lists,
# without using the set() data structure.
a = input().split()
b = input().split()
res = []
for i in a:
    if i in b and i not in res:
        res.append(i)
print(res)
```

```
# 10. Write a Python program that removes all elements located at even index positions
# from a list.
lst = input().split()
print([lst[i] for i in range(len(lst)) if i % 2 != 0])
```

```
# 11. Write a Python program that splits a list into smaller sublists (chunks) of size n.
lst = input().split()
n = int(input())
print([lst[i:i+n] for i in range(0, len(lst), n)])
```

```
# 12. Write a Python program that merges two already sorted lists into a single sorted
# list.
a = list(map(int, input().split()))
b = list(map(int, input().split()))
res = []
i = j = 0
while i < len(a) and j < len(b):
    if a[i] < b[j]:
        res.append(a[i]); i += 1
    else:
        res.append(b[j]); j += 1
res.extend(a[i:])
res.extend(b[j:])
print(res)
```

```
# 13. Write a Python program that finds the missing number from a list containing
# numbers from 1 to n, with exactly one number missing.
lst = list(map(int, input().split()))
n = len(lst) + 1
print(n*(n+1)//2 - sum(lst))
```

```
# 16. Write a Python program that finds the maximum and minimum elements in a tuple
# of numbers.
t = tuple(map(int, input().split()))
print(max(t), min(t))
```

```
# 17. Write a Python program that converts a tuple into a list, performs a modification,
# and then converts it back into a tuple.
t = tuple(input().split())
lst = list(t)
lst.append("X")
print(tuple(lst))
```

```
# 18. Write a Python program that counts the number of occurrences of a given
# element in a tuple.
t = tuple(input().split())
x = input()
print(t.count(x))
```

```
# 19. Write a Python program that removes duplicate elements from a tuple and returns
# a new tuple.
t = tuple(input().split())
res = []
for i in t:
    if i not in res:
        res.append(i)
print(tuple(res))
```

```
# 20.Write a Python program that sorts a tuple of integers and returns a new sorted
# tuple.
t = tuple(map(int, input().split()))
print(tuple(sorted(t)))
```

```
# 21. Write a Python program that finds the index of a specific element in a tuple,
# without using the built-in index() method.
t = tuple(input().split())
x = input()
for i in range(len(t)):
    if t[i] == x:
        print(i)
        break
```

```
# 22.Write a Python program that unpacks a tuple into multiple variables and prints
# each variable.
t = tuple(input().split())
a, b, c = t
print(a, b, c)
```

```
# 23.Write a Python program that checks whether a tuple is a palindrome, meaning it
# reads the same forward and backward.
t = tuple(input().split())
print(t == t[::-1])
```

```
# 24.Write a Python program that merges two tuples element-wise (pairing elements at
# the same index).
a = tuple(input().split())
b = tuple(input().split())
print(tuple(zip(a, b)))
```

```
# 25. Write a Python program that finds the common elements between two tuples.
a = tuple(input().split())
b = tuple(input().split())
print(tuple(i for i in a if i in b))
```

```
# 26.Write a Python program that removes duplicate elements from a list using a set
# and converts the result back into a list.
lst = input().split()
print(list(set(lst)))
```

```
# 27. Write a Python program that finds the union, intersection, and difference of two
# sets.
a = set(input().split())
b = set(input().split())
print(a | b, a & b, a - b)
```

```
# 28.Write a Python program that checks whether one set is a subset of another set.
a = set(input().split())
b = set(input().split())
print(a.issubset(b))
```

```
# 29.Write a Python program that finds the elements present in the first set but not in
# the second set.
a = set(input().split())
b = set(input().split())
```

```
print(a - b)
```

```
# 30.Write a Python program that removes the common elements from two sets,
# leaving only unique elements in each.
a = set(input().split())
b = set(input().split())
print(a - b, b - a)
```

```
# 31. Write a Python program that finds the symmetric difference between two sets.
a = set(input().split())
b = set(input().split())
print(a ^ b)
```

```
# 32.Write a Python program that checks whether two sets are disjoint, meaning they
# have no elements in common.
a = set(input().split())
b = set(input().split())
print(a.isdisjoint(b))
```

```
# 33.Write a Python program that converts a list of lists into a set of tuples so that
# duplicates can be removed.
lst = eval(input())
print(set(map(tuple, lst)))
```

```
# 34.Write a Python program that counts the number of unique vowels present in a
# string using a set.
s = input().lower()
print(len(set(s) & set('aeiou')))
```

```
# 35. Write a Python program that removes duplicate characters from a string using a
# set, while preserving the original order.
s = input()
res = []
seen = set()
for i in s:
    if i not in seen:
        seen.add(i)
        res.append(i)
print("".join(res))
```

```
# 36.Write a Python program that reverses a string without using slicing.
s = input()
rev = ""
for i in s:
    rev = i + rev
print(rev)
```

```
# 37. Write a Python program that checks whether a given string is a palindrome.
s = input()
print(s == s[::-1])
```

```
# 38.Write a Python program that checks whether two strings are anagrams of each
# other.
a = input()
b = input()
print(sorted(a) == sorted(b))
```

```
# 39.Write a Python program that finds the first non-repeating character in a string.
s = input()
for i in s:
    if s.count(i) == 1:
        print(i)
        break
```

```
# 40.Write a Python program that counts the frequency of each character in a string.
s = input()
freq = {}
for i in s:
    freq[i] = freq.get(i, 0) + 1
print(freq)
```

```
# 41. Write a Python program that removes all special characters from a string, keeping
# only alphabets and digits.
s = input()
print("".join(i for i in s if i.isalnum()))
```

```
# 42.Write a Python program that finds the longest word in a given sentence.
s = input().split()
print(max(s, key=len))
```

```
# 43.Write a Python program that capitalizes the first letter of each word in a sentence.
s = input()
print(" ".join(word.capitalize() for word in s.split()))
```

```
# 44.Write a Python program that checks whether a string contains only digits.
s = input()
print(s.isdigit())
```

```
# 45.Write a Python program that replaces all vowels in a string with the * character.
s = input()
print("".join('*' if i.lower() in 'aeiou' else i for i in s))
```

```
# 46.Write a Python program that counts the number of words in a sentence.
s = input()
print(len(s.split()))
```

```
# 47. Write a Python program that finds duplicate characters in a string.
s = input()
print({i for i in s if s.count(i) > 1})
```

```
# 48.Write a Python program that removes consecutive duplicate characters from a
# string.
s = input()
res = s[0]
for i in s[1:]:
    if i != res[-1]:
        res += i
print(res)
```

```
# 49.Write a Python program that checks whether a sentence is a valid palindrome,
# ignoring spaces, punctuation, and case.
import string
s = input().lower()
s = "".join(i for i in s if i.isalnum())
print(s == s[::-1])
```

```
# 50.Write a Python program that finds the most frequently occurring character in a
# string.
s = input()
print(max(set(s), key=s.count))
```

```
# 51. Write a Python program that counts the frequency of each element in a list using a
# dictionary.
lst = input().split()
d = {}
for i in lst:
    d[i] = d.get(i, 0) + 1
```

```
print(d)
```

```
# 52.Write a Python program that counts the frequency of each word in a sentence
# using a dictionary.
s = input().split()
d = {}
for i in s:
    d[i] = d.get(i, 0) + 1
print(d)
```

```
# 53.Write a Python program that sorts a dictionary by its values in ascending or
# descending order.
d = eval(input())
print(dict(sorted(d.items(), key=lambda x: x[1])))
```

```
# 54.Write a Python program that finds the key associated with the maximum value in
# a dictionary.
d = eval(input())
print(max(d, key=d.get))
```

```
# 55. Write a Python program that merges two dictionaries into a single dictionary.
d1 = eval(input())
d2 = eval(input())
d1.update(d2)
print(d1)
```

```
# 56. Write a Python program that removes dictionary entries that have duplicate
# values, keeping only one unique value.
d = eval(input())
res = {}
for k, v in d.items():
    if v not in res.values():
        res[k] = v
print(res)
```

```
# 57. Write a Python program that inverts a dictionary, meaning keys become values
# and values become keys.
d = eval(input())
print({v: k for k, v in d.items()})
```

```
# 58.Write a Python program that checks whether two dictionaries are equal,
# regardless of key order.
d1 = eval(input())
d2 = eval(input())
print(d1 == d2)
```

```
# 59.Write a Python program that groups dictionary keys that have the same value
# together.
d = eval(input())
res = {}
for k, v in d.items():
    res.setdefault(v, []).append(k)
print(res)
```

```
# 60.Write a Python program that finds the first non-repeating character in a string
# using a dictionary.
s = input()
d = {}
for i in s:
    d[i] = d.get(i, 0) + 1
for k, v in d.items():
    if v == 1:
        print(k)
        break
```