```
# -*- coding: utf-8 -*-
"""PPA3.ipynb

Automatically generated by Colaboratory.

Original file is located at
    https://colab.research.google.com/drive/1IBHQ7YvMWHHwXUEP-1PhugHgLA_ReAgu
"""

import numpy as np
li=[1,2,3,4]
tup=(5,6,7,8)
lia=np.array(li)
tupa=np.array(tup)
print(lia)
print(tupa)





size=int(input())
li=[]
for i in range(size):
    li.append(int(input()))
li=[int(s) for s in input().split()]
li=np.array(li)
li=[[1,2,3],[4,5,6],[7,8,9]]
matrix=np.array(li)
print(matrix)

def find_common_values(array1, array2):
    common_values = []
    for value in array1:
        if value in array2 and value not in common_values:
            common_values.append(value)
    return common_values

array1 = [1, 2, 3, 4, 5,0,0]
array2 = [3, 4, 5, 6, 7,0,0,0]
common_values = find_common_values(array1, array2)
print("Common values between array1 and array2:", common_values)

def add_commas(number):

    num_str = str(number)[::-1]
    result = ""

    for i in range(len(num_str)):
        if i % 3 == 0 and i != 0:
            result += ","
        result += num_str[i]
    result = result[::-1]

    return result

input_number = 100000000
output_number = add_commas(input_number)
print("Input:", input_number)
print("Output:", output_number)

def convert_formula(formula):
    result = ""
    i = 0
    while i < len(formula):
        if formula[i].isdigit() and i+1 < len(formula) and formula[i+1].isalpha():
            result += formula[i] + "*" + formula[i+1]
            i += 2
        elif formula[i].isalpha() and i+1 < len(formula) and formula[i+1].isdigit():
            result += formula[i] + "*" + formula[i+1]
            i += 2
        else:
            result += formula[i]
            i += 1
    return result
```

```
formula1 = "3X+4Y"
formula2 = "3(X+Y)"
print("Original Formula 1:", formula1)
print("Converted Formula 1:", convert_formula(formula1))
print("Original Formula 2:", formula2)
print("Converted Formula 2:", convert_formula(formula2))

import numpy as np


x = np.array([1, 2])
print("Integer Datatype: ")
print(x.dtype)

x = np.array([1.0, 2.0])
print("\nFloat Datatype: ")
print(x.dtype)

x = np.array([1, 2], dtype = np.int64)
print("\nForcing a Datatype: ")
print(x.dtype)

lst= [1,2,3,4,56,45,7,8,34,4,56,775,34,6,54,2]
lis=lst
list.sort

def is_sorted(arr)
    for i in range(1,len(arr)):
        if arr[i] <= arr[i-1]:
            return False

def is_sorted(stuff):
    for i in stuff:
        if stuff[i+1] >= stuff[i]:
            return True
    else:
        return False

numbers = [1, 0, 5, 2, 8]

print is_sorted(numbers)

import numpy as np
def issorted(li):
    return all(np.driff(li)>=0)
list1 = [1,2,3,4,5]
if issorted(li)

'''Write a function called remove duplicates that takes a list and returns a new list with
only the unique elements from the original. Hint: they don’t have to be in the same order.'''
def Remove(duplicate):
 final_list = []
 for num in duplicate:
  if num not in final_list:
   final_list.append(num)
 return final_list
duplicate = [4,4,4,4,44, 4]
print(Remove(duplicate))
```