NUMBER OF DISTINCT ELEMENTS

We want to count how many distinct/unique elements appear in the list of input.

Turn Input into a List (split by spaces)	InputList = **input().split()**
Make an empty list	ListName = **[ ]** or ListName = **list()**
Add "x" to the end of a list	ListName.**append(x)**
Increment a variable	VariableName **+=** 1

```
# Read a list of numbers
nums = ________________

# Make a list to track of numbers we've seen
seen = ______

# Make a variable to track of how many elements are distinct
_____ = ___


# Iterate through the numbers
for num in nums:

  # Check if number has NOT been seen before
  if num not in seen:

    # Increase the count of Distinct numbers
    ________  __  __

    # Add this number to the list we've seen
    seen._________(num)


# Display Output (Total Count)
print( ______ )
```

SWAP NEIGHBORS

Given a list of numbers, swap adjacent elements in each pair.
If there's an odd length, leave the last element intact.

Input: 1 2 3 4 5
v v v v v
Index **0 1 2 3 4** nums[0] = 1 nums[1] = 2 nums[2] = 3

Output: 2 1 4 3 5

Turn Input into a List (split by spaces)	InputList = **input().split()**
Make an empty list	ListName = **[ ]** or ListName = **list()**
Add "x" to the end of a list	ListName.**append(x)**
Access an element from a list by the Index	ListName[index]

```
# Read a list of numbers
nums = _________________

# Make a list to track of answer list
answer = _____

# Use a for loop to generate the First Index          range(start, end, step)
for index1 in range(0, len(nums)-1, 2):

  # Calculate the Second Index
  index2 = ____________

  # Calculate the First and Second Values
  value1 = _____________
  value2 = _____________

  # Add the values in reverse order to the list
  answer._______( _______ )
  answer._______( _______ )

# If the length is odd, add the last element in the list
if (len(nums) % 2 == 1):
  answer._______(nums[-1])

# Display the answer
print(" ".join(answer))
```

UNIQUE ELEMENTS

Instead of merely counting the distinct elements, we want to know **which** elements are **distinct**.

Turn Input into a List (split by spaces)	InputList = **input().split()**
Make an empty list	ListName = **[ ]** or ListName = **list()**
Add "x" to the end of a list	ListName.**append(x)**

```
# Read a list of integers:
nums = ____________

# Make a list to track numbers we've seen & duplicate numbers
seen = ______
duplicates = ______

# Iterate through the list
for num in nums:
  # Add unseen numbers to the seen list
  if num not in ______:
    ______.______(______)
  # Add other numbers to the duplicates list
  else:
    ______.______(______)

# Make a list to track the unique numbers
unique = ______
# Iterate through the list again
for num in nums:
  # if that number is not in the duplicates list, its unique
  if num not in ______:
    ______.______(______)

# Display the unique numbers
print(" ".join(unique))
```

BOWLING

Here's how the problem works:

Input	Meaning	Pins
10 3	Make 10 pins and 3 bowling balls	**["I", "I", "I", "I", "I", "I", "I", "I", "I", "I"]**
8 10	1st bowl: pins 8-10	**["I", "I", "I", "I", "I", "I", "I", ".", ".", "."]** 0 1 2 3 4 5 6 7 8 9 (index)
2 5	2nd bowl: pins 2-5	**["I", ".", ".", ".", ".", "I", "I", ".", ".", "."]** 0 1 2 3 4 5 6 7 8 9 (index)
3 6	3rd bowl: pins 3-6	**["I", ".", ".", ".", ".", ".", "I", ".", ".", "."]** 0 1 2 3 4 5 6 7 8 9 (index)

Final Pins	**["I", ".", ".", ".", ".", ".", "I", ".", ".", "."]**
Output	**I.....I...**

Helpful hints:

Convert list of **2** numbers into **Integers**	`N, K = map(int, input().split())`
Set up a list with **X elements** of **"I"**	`pins = ['I'] * N`

```
# Read number of pins and balls
N, K = map(int, input().split())

# Start with all pins standing, represented by 'I'
pins = ____________

# Knock down pins based on each ball's range
for _ in range(K):
    l, r = map(int, input().split())
    for i in range(l - 1, r):  # convert to 0-based index
        pins[i] = '.'

# Output final pin states
print(''.join(pins))
```

EIGHT QUEENS

```
# Initialize an empty list to store the coordinates of the queens
queens = []

# Input the coordinates of the queens
for Q in range(8):
    row, col = map(int, input().split())  # Input the coordinates as integers
    queens.append((row, col))  # Add the coordinates to the list of queens

# Initialize a variable to store if placement is valid
valid = True

# Iterate through each queen's coordinates
for Q1 in range(len(queens)):
    row1, col1 = queens[Q1]  # Coordinates of the first queen

    # Iterate through each next queen's coordinates
    for Q2 in range(Q1 + 1, len(queens)):
        row2, col2 = queens[Q2]  # Coordinates of the second queen

        # Check if queens are in the same row
        if ____ == ____ :
            valid = False
            break  # Exit the inner loop

        # Check if queens are in the same column
        elif ____ == ____ :
            valid = False
            break  # Exit the inner loop

        # Check if queens are in the same diagonal
        elif abs(____  - ____ ) == abs(____  - ____ ):
            valid = False
            break  # Exit the inner loop

    if not valid:  # If valid is already False, exit the outer loop as well
        break

# Print the final result based on the value of valid
if valid:
    print("NO")
else:
    print("YES")
```