Q1. Create a list `L = [123, "python", 3.7]` and print it.
Demo Output: `[123, 'python', 3.7]`

Q2. Create an empty list and take 4 integers from user, then print the list.
Demo Input: `10 20 30 40`
Demo Output: `[10, 20, 30, 40]`

Q3. Create a list of 5 subjects and print each on a new line.
Demo Output:
Math
Physics
Chemistry
English
Python

Q4. Given `['banana','apple','mango','tomato','berry']`, print elements at index 0 and 2.
Demo Output:
banana
mango

Q5. Print the last element using negative indexing.
Demo Output: `berry`

Q6. Take a list from user and print first and last element.
Demo Input: `5 9 1 7 3`
Demo Output:
First: 5
Last: 3

Q7. Slice list from index 1 to 3.
Demo Output: `['apple', 'mango']`

Q8. Slice last three elements using negative slicing.
Demo Output: `['mango', 'tomato', 'berry']`

Q9. Slice elements from index 2 to end.
Demo Input: `11 22 33 44 55 66`
Demo Output: `[33, 44, 55, 66]`

Q10. Concatenate two lists `[1,2,3]` and `['python','c']`.
Demo Output: `[1, 2, 3, 'python', 'c']`

Q11. Repeat the list `[1, 2, 3]` two times using the `*` operator.
Demo Output:
`[1, 2, 3, 1, 2, 3]`

Q12. Take two lists from the user and display the concatenated list.
Demo Input:

List1: `1 2 3`
List2: `7 8`
Demo Output:
`[1, 2, 3, 7, 8]`

Q13. Given the list `['python', 'c', 'java', 'php']`, check whether `'cpp'` is present using `in`.
Demo Output:
`False`

Q14. Given the list `[1, 2, 3, 4, 5]`, check whether `6` is not present using `not in`.
Demo Output:
`True`

Q15. Take a list and a value from the user and check whether the value exists in the list.
Demo Input:
List: `4 8 2 9`
Value: `2`
Demo Output:
`True`

Q16. Given `num = [1, 2, 3, 4, 5]`, change the element at index 2 to `30` and print the list.
Demo Output:
`[1, 2, 30, 4, 5]`

Q17. Replace the elements from index 1 to 2 in the list `[1, 2, 3, 4, 5]` with `[25, 36]` and print the list.
Demo Output:
`[1, 25, 36, 4, 5]`

Q18. Take a list and index–value from user and update that position.
Demo Input:
List: `10 20 30 40`
Index: `1`
New Value: `999`
Demo Output:
`[10, 999, 30, 40]`

Q19. Given `num = [1, 2, 3, 4, 5]`, delete the element at index 1 using `del` and print the list.
Demo Output:
`[1, 3, 4, 5]`

Q20. Given `num = [1, 2, 3, 4, 5]`, delete the elements from index 1 to 2 using slicing with `del` and print the list.
Demo Output:
`[1, 4, 5]`

Q21. Given the list `num = [9, 8, 7, 6]`, delete the last element using `del` and print the list.
Demo Output:
```
[9, 8, 7]
```

Q22. Iterate the list `['python','c','java','php']` using a `for` loop and print each element.
Demo Output:

```
python
c
java
php
```

Q23. Take a list of numbers from user and print them in one line.
Demo Input:
```
1 4 9 16
```
Demo Output:
```
1 4 9 16
```

Q24. Count how many elements are greater than 10 in a list.
Demo Input:
```
5 12 3 25 10 19
```
Demo Output:
```
Count: 3
```

Q25. Find the length of the list `[1,2,3,4,5,6]` using `len()`.
Demo Output:
```
6
```

Q26. For the list `[1,3,2,4,6,5]`, find maximum, minimum, and sum.
Demo Output:

```
Max: 6
Min: 1
Sum: 21
```

Q27. Convert the string `"python"` into a list using `list()`.
Demo Output:
```
['p', 'y', 't', 'h', 'o', 'n']
```

Q28. Sort the list `[1,3,2,4,6,5]` using `sorted()` and print the result.
Demo Output:
```
[1, 2, 3, 4, 5, 6]
```

Q29. Sort the list `['java','c','python','cpp']` using `sorted()`.
Demo Output:
```
['c', 'cpp', 'java', 'python']
```

Q30. Take a list from user and display the sorted list.
Demo Input:
`9 1 5 2`
Demo Output:
`[1, 2, 5, 9]`

Q31. Given `num = [1,2,3,4,5]`, append `6` using `append()` and print the list.
Demo Output:
`[1, 2, 3, 4, 5, 6]`

Q32. Given `list1 = [1,2,3,2,5]`, remove `2` using `remove()` and print the list.
Demo Output:
`[1, 3, 2, 5]`

Q33. Take a list and a value from user. If the value exists, remove it; otherwise print `"Not Found"`.
Demo Input:
List: `10 20 30 40`
Value: `25`
Demo Output:
`Not Found`

Q34. Sort the list `[6,8,2,4,10]` using `sort()` and print it.
Demo Output:
`[2, 4, 6, 8, 10]`

Q35. Sort the same list in descending order using `sort(reverse=True)` and print it.
Demo Output:
`[10, 8, 6, 4, 2]`

Q36. Reverse the list `[6,8,2,4,10]` using `reverse()` and print it.
Demo Output:
`[10, 4, 2, 8, 6]`

Q37. For `num = [1,2,3,4,3,2,2,1,4,5,8]`, count the occurrences of `2` using `count()`.
Demo Output:
`3`

Q38. For `list1 = ['p','y','t','o','n','p']`, find the index of `'t'` using `index()`.
Demo Output:
`2`

Q39. Take a list and a value from user. If the value exists, print its index; otherwise print `-1`.
Demo Input:
List: `5 7 9 2`
Value: `9`

Demo Output:
```
2
```

Q40. Insert `60` at index `4` in the list `[10,20,30,40,50]` using `insert()` and print the list.
Demo Output:
```
[10, 20, 30, 40, 60, 50]
```

Q41. Insert `70` at index `7` in the list `[10,20,30,40,60,50]` and print the list.
Demo Output:
```
[10, 20, 30, 40, 60, 50, 70]
```

Q42. Extend the list `[60,70]` with `[10,20,30,40,50]` using `extend()` and print the final list.
Demo Output:
```
[60, 70, 10, 20, 30, 40, 50]
```

Q43. Given `num = [10,20,30,40,50]`, remove the last element using `pop()` and print the list.
Demo Output:
```
[10, 20, 30, 40]
```

Q44. Remove the element at index `2` using `pop()` and print the list.
Demo Output:
```
[10, 20, 40]
```

Q45. Clear all elements of the list `[10,20,30,40,50]` using `clear()` and print it.
Demo Output:
```
[]
```

Q46. Given the 2D list `L1 = [[1,2,3],[4,5,7],[6,7,8]]`, print the element at row index `1` and column index `2`.
Demo Output:
```
7
```

Q47. Take a 3×3 matrix (2D list) from user and print the element at row index `2` and column index `1`.
Demo Output (example):
```
8
```

Q48. Print all elements of the 2D list `[[1,2],[3,4]]` in matrix form.
Demo Output:

```
1 2
3 4
```