

Lists Class 11

Computer Science

NCERT Solutions

*Step-by-step answers for the 2026-27 NCERT
exercise*

Chapter 9: Lists

This PDF solves every exercise and programming problem from Class 11 Computer Science Chapter 9 Lists. It covers list output tracing, append and extend, deletion, nested lists, functions, median, duplicate removal and in-place reversal according to the 2026-27 NCERT book.

Also see for this chapter:

Exercise Solutions

Q 9.1 What will be the output of the following statements? (i) `list1 = [12,32,65,26,80,10]; list1.sort(); print(list1)`. (ii) `list1 = [12,32,65,26,80,10]; sorted(list1); print(list1)`. (iii) `list1 = [1,2,3,4,5,6,7,8,9,10]; list1[::-2]; list1[:3] + list1[3:]`. (iv) `list1 = [1,2,3,4,5]; list1[len(list1)-1]`.

SOLUTION

Quick solution

Source: NCERT Class 11 Computer Science Chapter 9 Exercise and Programming Problems, pages 204-206, Reprint 2026-27. Intent: Python list operations and programs.

Concept used. The method `sort()` changes the same list in-place. The function `sorted()` returns a new sorted list. Slicing expressions return list values.

Step 1. In part (i), `list1.sort()` arranges the list

itself in ascending order. `print(list1)` displays `[10, 12, 26, 32, 65, 80]`.

Step 2. In part (ii), `sorted(list1)` creates a sorted list but the result is not stored. So `print(list1)` displays the original list `[12, 32, 65, 26, 80, 10]`.

Step 3. In part (iii), `list1[::-2]` starts from 10 and moves left by two places. Its value is `[10, 8, 6, 4, 2]`.

Step 4. The expression `list1[:3] + list1[3:]` joins the first three elements with the remaining elements. Its value is `[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]`.

Step 5. In part (iv), `len(list1)-1` is the last valid index. The expression value is 5.

 Exam tip

`sorted(list1)` does not change `list1` unless its returned list is assigned to a variable.

Final Answer: `[10, 12, 26, 32, 65, 80];`
`[12, 32, 65, 26, 80, 10]; [10, 8, 6, 4, 2]`
and `[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]; 5`.

Scoring angle. List questions reward exact tracing. Write the index, method effect and final list state before the answer.

Step 1. Identify whether the question asks for output tracing, a concept difference or a program.

Step 2. In part (i), `list1.sort()` arranges the list itself in ascending order. `print(list1)` displays `[10, 12, 26, 32, 65, 80]`.

Step 3. In part (ii), `sorted(list1)` creates a sorted list but the result is not stored. So `print(list1)` displays the original list `[12, 32, 65, 26, 80, 10]`.

Step 4. In part (iii), `list1[::-2]` starts from 10 and moves left by two places. Its value is `[10, 8, 6, 4, 2]`.

Step 5. The expression `list1[:3] + list1[3:]` joins the first three elements with the remaining elements. Its value is `[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]`.

Step 6. In part (iv), `len(list1)-1` is the last valid index. The expression value is 5.

Step 7. End with the exact list value, code statement or program result required

by the question.

For programs, keep the function small and let the main part handle input and printing. For output questions, never skip the difference between returned value and in-place change.

Final Answer: [10, 12, 26, 32, 65, 80];
[12, 32, 65, 26, 80, 10]; [10, 8, 6, 4, 2]
and [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]; 5.

Q 9.2 Consider the list `myList = [10,20,30,40]`. What will be the elements of `myList` after these two operations: (i) `myList.append([50,60])` (ii) `myList.extend([80,90])`?

SOLUTION

Quick solution

Source: NCERT Class 11 Computer Science Chapter 9 Exercise and Programming Problems, pages 204-206, Reprint 2026-27. Intent: Python list operations and programs.

Concept used. `append()` adds its argument as

one element. `extend()` adds each element of the iterable to the list.

Step 1. The starting list is `[10, 20, 30, 40]`.

Step 2. After `myList.append([50,60])`, the list `[50,60]` is added as one nested element.

Step 3. So `myList` becomes `[10, 20, 30, 40, [50, 60]]`.

Step 4. After `myList.extend([80,90])`, the two values 80 and 90 are added separately.

Step 5. So the final list becomes `[10, 20, 30, 40, [50, 60], 80, 90]`.

 Exam tip

`append` keeps the list argument packed.
`extend` unpacks its elements.

Final Answer: After `append`: `[10, 20, 30, 40, [50, 60]]`. After `extend`: `[10, 20, 30, 40, [50, 60], 80, 90]`.

Scoring angle. List questions reward exact tracing. Write the index, method effect and final list state before the answer.

Step 1. Identify whether the question asks for output tracing, a concept difference or a program.

Step 2. The starting list is [10, 20, 30, 40].

Step 3. After `myList.append([50,60])`, the list [50,60] is added as one nested element.

Step 4. So `myList` becomes [10, 20, 30, 40, [50, 60]].

Step 5. After `myList.extend([80,90])`, the two values 80 and 90 are added separately.

Step 6. So the final list becomes [10, 20, 30, 40, [50, 60], 80, 90].

Step 7. End with the exact list value, code statement or program result required by the question.

For programs, keep the function small and let the main part handle input and printing. For output questions, never skip the difference between returned value and in-place change.

Final Answer: After append: [10, 20, 30, 40, [50, 60]]. After extend: [10, 20, 30, 40, [50, 60], 80, 90].

Q 9.3 What will be the output of this code segment: `myList = [1,2,3,4,5,6,7,8,9,10];`
`for i in range(0,len(myList)):` if `i%2 == 0:`
`print(myList[i])`?

SOLUTION

Quick solution

Source: NCERT Class 11 Computer Science Chapter 9 Exercise and Programming Problems, pages 204-206, Reprint 2026-27. Intent: Python list operations and programs.

Concept used. The loop visits all indexes. The condition `i`

Step 1. The valid indexes are 0 to 9.

Step 2. Even indexes are 0, 2, 4, 6 and 8.

Step 3. The elements at these indexes are 1, 3, 5, 7 and 9.

Step 4. Each value is printed on a new line

because `print()` is inside the loop.

 Exam tip

The condition tests the index `i`. It does not test whether `myList[i]` is even.

Final Answer: 1, 3, 5, 7 and 9 are printed on separate lines.

EXPERT'S SOLUTION : *Rohit Iyer, B.Tech
Computer Science, IIT Roorkee*

Scoring angle. List questions reward exact tracing. Write the index, method effect and final list state before the answer.

Step 1. Identify whether the question asks for output tracing, a concept difference or a program.

Step 2. The valid indexes are 0 to 9.

Step 3. Even indexes are 0, 2, 4, 6 and 8.

Step 4. The elements at these indexes are 1, 3, 5, 7 and 9.

Step 5. Each value is printed on a new line because `print()` is inside the loop.

Step 6. End with the exact list value, code statement or program result required

by the question.

For programs, keep the function small and let the main part handle input and printing. For output questions, never skip the difference between returned value and in-place change.

Final Answer: 1, 3, 5, 7 and 9 are printed on separate lines.

Q 9.4 What will be the output of the following code segments? (a) `del myList[3:]` then `print(myList)`. (b) `del myList[:5]` then `print(myList)`. (c) `del myList[::2]` then `print(myList)`, where each case starts with `myList = [1,2,3,4,5,6,7,8,9,10]`.

SOLUTION

Quick solution

Source: NCERT Class 11 Computer Science Chapter 9 Exercise and Programming Problems, pages 204-206, Reprint 2026-27. Intent: Python list operations and programs.

Concept used. The `del` statement removes the

slice from the list in-place. Slice boundaries follow the same start, stop and step rules used for reading.

Step 1. `del myList[3:]` removes elements from index 3 to the end. The remaining list is `[1, 2, 3]`.

Step 2. `del myList[:5]` removes indexes 0 to 4. The remaining list is `[6, 7, 8, 9, 10]`.

Step 3. `del myList[::2]` removes the elements at even indexes 0, 2, 4, 6 and 8.

Step 4. The values removed in part (c) are 1, 3, 5, 7 and 9, so the remaining list is `[2, 4, 6, 8, 10]`.

 Exam tip

In deletion with a step, Python decides the slice positions first, then removes those elements.

Final Answer: (a) `[1, 2, 3]`; (b) `[6, 7, 8, 9, 10]`; (c) `[2, 4, 6, 8, 10]`.

Scoring angle. List questions reward exact tracing. Write the index, method effect and final list state before the answer.

Step 1. Identify whether the question asks for output tracing, a concept difference or a program.

Step 2. `del myList[3:]` removes elements from index 3 to the end. The remaining list is [1, 2, 3].

Step 3. `del myList[:5]` removes indexes 0 to 4. The remaining list is [6, 7, 8, 9, 10].

Step 4. `del myList[::2]` removes the elements at even indexes 0, 2, 4, 6 and 8.

Step 5. The values removed in part (c) are 1, 3, 5, 7 and 9, so the remaining list is [2, 4, 6, 8, 10].

Step 6. End with the exact list value, code statement or program result required by the question.

For programs, keep the function small and let the main part handle input and printing. For output questions, never skip the difference between returned value and in-place change.

Final Answer: (a) [1, 2, 3]; (b) [6, 7, 8, 9, 10]; (c) [2, 4, 6, 8, 10].

Q 9.5 Differentiate between `append()` and `extend()` functions of list.

SOLUTION

Quick solution

Source: NCERT Class 11 Computer Science Chapter 9 Exercise and Programming Problems, pages 204-206, Reprint 2026-27. Intent: Python list operations and programs.

Concept used. Both methods add data at the end of a list, but they treat the argument differently.

Step 1. `append(x)` adds `x` as one new element at the end of the list.

Step 2. If `x` is another list, `append(x)` places that entire list as a nested element.

Step 3. `extend(iterable)` takes each element from the iterable and adds those elements one by one.

Step 4. For example,
[10,20].append([30,40]) gives
[10,20,[30,40]], while
[10,20].extend([30,40]) gives
[10,20,30,40].

 Exam tip

Use append when you want one extra item.
Use extend when you want to merge many items.

Final Answer: append() adds one object as a single element. extend() adds each element of another iterable separately.

EXPERT'S SOLUTION : *Kabir Sharma, B.Tech
Computer Science, DTU*

Scoring angle. List questions reward exact tracing. Write the index, method effect and final list state before the answer.

Step 1. Identify whether the question asks for output tracing, a concept difference or a program.

Step 2. append(x) adds x as one new element at the end of the list.

Step 3. If x is another list, append(x) places

that entire list as a nested element.

Step 4. `extend(iterable)` takes each element from the iterable and adds those elements one by one.

Step 5. For example,
`[10,20].append([30,40])` gives
`[10,20,[30,40]]`, while
`[10,20].extend([30,40])` gives
`[10,20,30,40]`.

Step 6. End with the exact list value, code statement or program result required by the question.

For programs, keep the function small and let the main part handle input and printing. For output questions, never skip the difference between returned value and in-place change.

Final Answer: `append()` adds one object as a single element. `extend()` adds each element of another iterable separately.

Q9.6 Consider `list1 = [6,7,8,9]`. What is the difference between `list1 * 2`, `list1 *= 2` and `list1 = list1 * 2`?

 Quick solution

Source: NCERT Class 11 Computer Science Chapter 9 Exercise and Programming Problems, pages 204-206, Reprint 2026-27. Intent: Python list operations and programs.

Concept used. The repetition operator creates repeated list content. Assignment decides whether the result is only returned, assigned, or applied in-place.

Step 1. `list1 * 2` returns `[6, 7, 8, 9, 6, 7, 8, 9]`, but `list1` remains `[6, 7, 8, 9]` if the result is not assigned.

Step 2. `list1 *= 2` updates `list1` in-place to `[6, 7, 8, 9, 6, 7, 8, 9]`. Existing aliases to the same list see the change.

Step 3. `list1 = list1 * 2` creates a repeated list and then binds the name `list1` to that new list.

Step 4. After either assignment form, `list1` stores the repeated list, but alias behaviour is different.

Exam tip

The star operation gives repeated content. Assignment controls whether the variable changes.

Final Answer: `list1 * 2` only returns a repeated list. `list1 *= 2` changes the same list in-place. `list1 = list1 * 2` assigns a new repeated list to `list1`.

EXPERT'S SOLUTION : *Nisha Verma, M.Sc
Information Technology, BHU*

Scoring angle. List questions reward exact tracing. Write the index, method effect and final list state before the answer.

Step 1. Identify whether the question asks for output tracing, a concept difference or a program.

Step 2. `list1 * 2` returns `[6, 7, 8, 9, 6, 7, 8, 9]`, but `list1` remains `[6, 7, 8, 9]` if the result is not assigned.

Step 3. `list1 *= 2` updates `list1` in-place to `[6, 7, 8, 9, 6, 7, 8, 9]`. Existing aliases to the same list see the change.

Step 4. `list1 = list1 * 2` creates a repeated list

and then binds the name list1 to that new list.

Step 5. After either assignment form, list1 stores the repeated list, but alias behaviour is different.

Step 6. End with the exact list value, code statement or program result required by the question.

For programs, keep the function small and let the main part handle input and printing. For output questions, never skip the difference between returned value and in-place change.

Final Answer: list1 * 2 only returns a repeated list. list1 *= 2 changes the same list in-place. list1 = list1 * 2 assigns a new repeated list to list1.

Q 9.7 The record stRecord = ['Raman','A-36',[56,98,99,72,69],78.8] stores a student record. Write Python statements to retrieve: (a) percentage, (b) marks in the fifth subject, (c) maximum marks, (d) roll no, (e) change the name from Raman to Raghav.

 Quick solution

Source: NCERT Class 11 Computer Science Chapter 9 Exercise and Programming Problems, pages 204-206, Reprint 2026-27. Intent: Python list operations and programs.

Concept used. A nested list is accessed with two indexes. The outer index selects the marks list, and the inner index selects one subject mark.

Step 1. Percentage is the fourth outer element, so use `stRecord[3]`.

Step 2. Marks are stored in `stRecord[2]`. The fifth subject has inner index 4, so use `stRecord[2][4]`.

Step 3. Maximum marks are found with `max(stRecord[2])`.

Step 4. Roll number is the second outer element, so use `stRecord[1]`.

Step 5. Lists are mutable, so `stRecord[0] = 'Raghav'` changes the name.

 Exam tip

For the marks list, first reach `stRecord[2]`, then index inside that list.

Final Answer: `stRecord[3],`
`stRecord[2][4], max(stRecord[2]),`
`stRecord[1],` and `stRecord[0] =`
`'Raghav'.`

EXPERT'S SOLUTION : *Arjun Iyer, M.Tech*
Computer Engineering, NIT Warangal

Scoring angle. List questions reward exact tracing. Write the index, method effect and final list state before the answer.

Step 1. Identify whether the question asks for output tracing, a concept difference or a program.

Step 2. Percentage is the fourth outer element, so use `stRecord[3]`.

Step 3. Marks are stored in `stRecord[2]`. The fifth subject has inner index 4, so use `stRecord[2][4]`.

Step 4. Maximum marks are found with `max(stRecord[2])`.

Step 5. Roll number is the second outer element, so use `stRecord[1]`.

Step 6. Lists are mutable, so `stRecord[0] = 'Raghav'` changes the name.

Step 7. End with the exact list value, code statement or program result required by the question.

For programs, keep the function small and let the main part handle input and printing. For output questions, never skip the difference between returned value and in-place change.

Final Answer: `stRecord[3],`
`stRecord[2][4], max(stRecord[2]),`
`stRecord[1], and stRecord[0] =`
`'Raghav'.`

Q 9.8 Write a program to find the number of times an element occurs in the list.

SOLUTION

Quick solution

Source: NCERT Class 11 Computer Science Chapter 9 Exercise and Programming Problems, pages 204-206, Reprint 2026-27. Intent: Python list operations and programs.

Concept used. A counter increases whenever the current list element matches the target

element.

Step 1. Read the number of elements and create the list.

Step 2. Input the element that has to be counted.

Step 3. Traverse the list one element at a time.

Step 4. Increase count whenever the element equals the target.

Step 5. Print the final count.

```
n = int(input('How many elements: '))
lst = []
for i in range(n):
    lst.append(int(input('Enter element: ')))
target = int(input('Element to count: '))
count = 0
for item in lst:
    if item == target:
        count += 1
print(target, 'occurs', count, 'times')
```

Exam tip

`list.count(target)` is shorter, but a loop shows the logic clearly.

Final Answer: The program traverses the list and prints how many entries match the target element.

EXPERT'S SOLUTION : *Farah Khan, M.C.A., Jamia Millia Islamia*

Scoring angle. List questions reward exact tracing. Write the index, method effect and final list state before the answer.

Step 1. Identify whether the question asks for output tracing, a concept difference or a program.

Step 2. Read the number of elements and create the list.

Step 3. Input the element that has to be counted.

Step 4. Traverse the list one element at a time.

Step 5. Increase count whenever the element equals the target.

Step 6. Print the final count.

Step 7. End with the exact list value, code statement or program result required by the question.

For programs, keep the function small and let the main part handle input and printing. For output questions, never skip the difference between returned value and in-place change.

Final Answer: The program traverses the list and prints how many entries match the target element.

[Download the Full Revision Notes PDF →](#)

Q 9.9 Write a program to read a list of n integers, positive as well as negative. Create two new lists, one having all positive numbers and the other having all negative numbers from the given list. Print all three lists.

Quick solution

Source: NCERT Class 11 Computer Science Chapter 9 Exercise and Programming Problems, pages 204-206, Reprint 2026-27. Intent: Python list operations and programs.

Concept used. Conditional traversal can place each value into a different output list.

- Step 1.** Read n and then read n integers into one list.
- Step 2.** Create two empty lists named positive and negative.
- Step 3.** For each number, append it to positive if it is greater than zero.
- Step 4.** Append it to negative if it is less than zero.
- Step 5.** Print the original, positive and negative lists. Zero is neither positive nor negative.

```
n = int(input('How many integers: '))
numbers = []
for i in range(n):
    numbers.append(int(input('Enter number: ')))
positive = []
negative = []
for num in numbers:
    if num > 0:
        positive.append(num)
    elif num < 0:
        negative.append(num)
print('Original list:', numbers)
print('Positive list:', positive)
print('Negative list:', negative)
```

Exam tip

Keep zero out unless the question asks for a separate zero list.

Final Answer: The program builds two new lists by checking each integer against zero.

EXPERT'S SOLUTION : *Vivek Bansal, M.Sc*
Computer Science, Pune University

Scoring angle. List questions reward exact tracing. Write the index, method effect and final list state before the answer.

Step 1. Identify whether the question asks for

output tracing, a concept difference or a program.

- Step 2.** Read n and then read n integers into one list.
- Step 3.** Create two empty lists named positive and negative.
- Step 4.** For each number, append it to positive if it is greater than zero.
- Step 5.** Append it to negative if it is less than zero.
- Step 6.** Print the original, positive and negative lists. Zero is neither positive nor negative.
- Step 7.** End with the exact list value, code statement or program result required by the question.

For programs, keep the function small and let the main part handle input and printing. For output questions, never skip the difference between returned value and in-place change.

Final Answer: The program builds two new lists by checking each integer against zero.

Q 9.10

Write a function that returns the largest element of the list passed as parameter.

SOLUTION **Quick solution**

Source: NCERT Class 11 Computer Science Chapter 9 Exercise and Programming Problems, pages 204-206, Reprint 2026-27. Intent: Python list operations and programs.

Concept used. A function can scan the list while keeping the largest value seen so far.

Step 1. Assume the first element is the largest.

Step 2. Compare each remaining element with the current largest.

Step 3. If an element is greater, update the largest value.

Step 4. Return the largest value after the loop ends.

```
def largest(lst):  
    big = lst[0]  
    for item in lst:  
        if item > big:  
            big = item  
    return big  
  
numbers = [12, 45, 7, 89, 34]  
print('Largest element:', largest(numbers))
```

Exam tip

Do not initialise the largest value as 0 because the list may contain only negative numbers.

Final Answer: The function returns the maximum element after comparing all list items.

EXPERT'S SOLUTION : *Neha Joshi, B.Tech
Computer Science, NIT Surat*

Scoring angle. List questions reward exact tracing. Write the index, method effect and final list state before the answer.

Step 1. Identify whether the question asks for output tracing, a concept difference or a program.

Step 2. Assume the first element is the largest.

- Step 3.** Compare each remaining element with the current largest.
- Step 4.** If an element is greater, update the largest value.
- Step 5.** Return the largest value after the loop ends.
- Step 6.** End with the exact list value, code statement or program result required by the question.

For programs, keep the function small and let the main part handle input and printing. For output questions, never skip the difference between returned value and in-place change.

Final Answer: The function returns the maximum element after comparing all list items.

Q 9.11 Write a function to return the second largest number from a list of numbers.

Quick solution

Source: NCERT Class 11 Computer Science Chapter 9 Exercise and Programming Problems, pages 204-206, Reprint 2026-27. Intent: Python list operations and programs.

Concept used. The second largest distinct value can be found by tracking the largest and second largest values during one traversal.

- Step 1.** Start largest and second largest as None.
- Step 2.** For each number, update largest when the number is greater than the current largest.
- Step 3.** When largest changes, the old largest becomes the second largest.
- Step 4.** If a number lies between largest and second largest, update only second largest.
- Step 5.** Ignore duplicates of the current largest for a distinct second largest answer.

```
def second_largest(lst):  
    largest = None  
    second = None  
    for num in lst:  
        if largest is None or num > largest:  
            second = largest  
            largest = num  
        elif num != largest and (second is None or num > second):  
            second = num  
    return second  
  
numbers = [20, 45, 45, 12, 39]  
print('Second largest:', second_largest(numbers))
```

Exam tip

If all values are equal, the function returns None because no distinct second largest value exists.

Final Answer: The function returns the second largest distinct number from the list.

EXPERT'S SOLUTION : *Samar Gupta, M.Tech
Software Systems, BITS Pilani*

Scoring angle. List questions reward exact tracing. Write the index, method effect and final list state before the answer.

Step 1. Identify whether the question asks for

output tracing, a concept difference or a program.

- Step 2.** Start largest and second largest as None.
- Step 3.** For each number, update largest when the number is greater than the current largest.
- Step 4.** When largest changes, the old largest becomes the second largest.
- Step 5.** If a number lies between largest and second largest, update only second largest.
- Step 6.** Ignore duplicates of the current largest for a distinct second largest answer.
- Step 7.** End with the exact list value, code statement or program result required by the question.

For programs, keep the function small and let the main part handle input and printing. For output questions, never skip the difference between returned value and in-place change.

Final Answer: The function returns the second largest distinct number from the list.

Q 9.12 Write a program to read a list of n integers and find their median.

SOLUTION

Quick solution

Source: NCERT Class 11 Computer Science Chapter 9 Exercise and Programming Problems, pages 204-206, Reprint 2026-27. Intent: Python list operations and programs.

Concept used. The median is the middle value after sorting. If there are two middle values, the median is their average.

Step 1. Read the list of integers.

Step 2. Sort the list in ascending order.

Step 3. If n is odd, take the element at index $n//2$.

Step 4. If n is even, average the elements at indexes $n//2 - 1$ and $n//2$.

Step 5. Print the sorted list and median.

```
n = int(input('How many integers: '))
lst = []
for i in range(n):
    lst.append(int(input('Enter integer: ')))
lst.sort()
mid = n // 2
if n % 2 == 1:
    median = lst[mid]
else:
    median = (lst[mid - 1] + lst[mid]) / 2
print('Sorted list:', lst)
print('Median:', median)
```

Exam tip

Median depends on sorted order, not on the input order.

Final Answer: The program sorts the list and then applies the odd or even median rule.

EXPERT'S SOLUTION : Aditi Sen, M.Sc Computer Applications, University of Mumbai

Scoring angle. List questions reward exact tracing. Write the index, method effect and final list state before the answer.

Step 1. Identify whether the question asks for output tracing, a concept difference or a program.

Step 2. Read the list of integers.

Step 3. Sort the list in ascending order.

Step 4. If n is odd, take the element at index $n//2$.

Step 5. If n is even, average the elements at indexes $n//2 - 1$ and $n//2$.

Step 6. Print the sorted list and median.

Step 7. End with the exact list value, code statement or program result required by the question.

For programs, keep the function small and let the main part handle input and printing. For output questions, never skip the difference between returned value and in-place change.

Final Answer: The program sorts the list and then applies the odd or even median rule.

Q9.13 Write a program to read a list of elements. Modify this list so that it does not contain any duplicate elements.

 Quick solution

Source: NCERT Class 11 Computer Science Chapter 9 Exercise and Programming Problems, pages 204-206, Reprint 2026-27. Intent: Python list operations and programs.

Concept used. A second list can store only the first occurrence of each element.

Step 1. Read all elements into a list.

Step 2. Create an empty list named unique.

Step 3. Traverse the original list.

Step 4. Append an element to unique only if it is not already present.

Step 5. Assign unique back to the original list name and print it.

```
n = int(input('How many elements: '))
lst = []
for i in range(n):
    lst.append(input('Enter element: '))
unique = []
for item in lst:
    if item not in unique:
        unique.append(item)
lst = unique
print('List without duplicates:', lst)
```

Exam tip

This method preserves the original order of first appearances.

Final Answer: The program keeps the first occurrence of each element and removes later duplicates.

EXPERT'S SOLUTION : *Rohan Das, B.Tech
Information Technology, Jadavpur University*

Scoring angle. List questions reward exact tracing. Write the index, method effect and final list state before the answer.

Step 1. Identify whether the question asks for output tracing, a concept difference or a program.

Step 2. Read all elements into a list.

Step 3. Create an empty list named unique.

Step 4. Traverse the original list.

Step 5. Append an element to unique only if it is not already present.

Step 6. Assign unique back to the original list name and print it.

Step 7. End with the exact list value, code

statement or program result required by the question.

For programs, keep the function small and let the main part handle input and printing. For output questions, never skip the difference between returned value and in-place change.

Final Answer: The program keeps the first occurrence of each element and removes later duplicates.

Q 9.14 Write a program to read a list of elements. Input an element from the user that has to be inserted in the list. Also input the position at which it is to be inserted. Write a user-defined function to insert the element at the desired position in the list.

Quick solution

Source: NCERT Class 11 Computer Science Chapter 9 Exercise and Programming Problems, pages 204-206, Reprint 2026-27. Intent: Python list operations and programs.

Concept used. A list is mutable, so a function can insert an element into the same list object.

Step 1. Read the list.

Step 2. Read the element and the required position.

Step 3. Pass the list, position and element to the function.

Step 4. Use `insert(position, element)` inside the function.

Step 5. Print the modified list.

```
def insert_element(lst, pos, element):  
    lst.insert(pos, element)  
  
n = int(input('How many elements: '))  
lst = []  
for i in range(n):  
    lst.append(input('Enter element: '))  
element = input('Element to insert: ')  
pos = int(input('Position/index: '))  
insert_element(lst, pos, element)  
print('Modified list:', lst)
```

Exam tip

The question uses position, but Python `insert()` expects a zero-based index. State this clearly in practical work.

Final Answer: The function inserts the element at the user-given index and the main program prints the changed list.

EXPERT'S SOLUTION : Kavya Iyer, M.C.A., Christ University

Scoring angle. List questions reward exact tracing. Write the index, method effect and final list state before the answer.

Step 1. Identify whether the question asks for output tracing, a concept difference or

a program.

Step 2. Read the list.

Step 3. Read the element and the required position.

Step 4. Pass the list, position and element to the function.

Step 5. Use `insert(position, element)` inside the function.

Step 6. Print the modified list.

Step 7. End with the exact list value, code statement or program result required by the question.

For programs, keep the function small and let the main part handle input and printing. For output questions, never skip the difference between returned value and in-place change.

Final Answer: The function inserts the element at the user-given index and the main program prints the changed list.

Q 9.15 Read elements of a list. (a) Ask for the position of the element to be deleted and write a function to delete the element at that position. (b) Ask for the value of the

element to be deleted and write a function to delete the element of this value from the list.

SOLUTION

Quick solution

Source: NCERT Class 11 Computer Science Chapter 9 Exercise and Programming Problems, pages 204-206, Reprint 2026-27. Intent: Python list operations and programs.

Concept used. `pop(index)` deletes by position. `remove(value)` deletes the first matching value.

Step 1. Create the list from user input.

Step 2. For position-based deletion, check that the index is valid and then use `pop(pos)`.

Step 3. For value-based deletion, check membership and then use `remove(value)`.

Step 4. Print clear messages when the index is invalid or the value is absent.

```
def delete_by_position(lst, pos):
    if 0 <= pos < len(lst):
        lst.pop(pos)
    else:
        print('Invalid position')

def delete_by_value(lst, value):
    if value in lst:
        lst.remove(value)
    else:
        print('Value not present')

lst = [10, 20, 30, 40, 50]
delete_by_position(lst, int(input('Position/index to delete: ')))
delete_by_value(lst, int(input('Value to delete: ')))
print('Modified list:', lst)
```

Exam tip

`remove()` deletes the first occurrence only. Use a loop if all occurrences must be deleted.

Final Answer: Use `pop()` for deletion by position and `remove()` for deletion by value.

Scoring angle. List questions reward exact tracing. Write the index, method effect and final list state before the answer.

Step 1. Identify whether the question asks for output tracing, a concept difference or a program.

Step 2. Create the list from user input.

Step 3. For position-based deletion, check that the index is valid and then use `pop(pos)`.

Step 4. For value-based deletion, check membership and then use `remove(value)`.

Step 5. Print clear messages when the index is invalid or the value is absent.

Step 6. End with the exact list value, code statement or program result required by the question.

For programs, keep the function small and let the main part handle input and printing. For output questions, never skip the difference between returned value and in-place change.

Final Answer: Use `pop()` for deletion by position and `remove()` for deletion by value.

Q 9.16 Read a list of n elements. Pass this list to a function which reverses this list in-place without creating a new list.

SOLUTION

Quick solution

Source: NCERT Class 11 Computer Science Chapter 9 Exercise and Programming Problems, pages 204-206, Reprint 2026-27. Intent: Python list operations and programs.

Concept used. In-place reversal swaps elements from the two ends and moves both indexes toward the centre.

Step 1. Set `left` to 0 and `right` to `len(lst)-1`.

Step 2. Swap `lst[left]` and `lst[right]`.

Step 3. Increase `left` and decrease `right`.

Step 4. Repeat while `left` is less than `right`.

Step 5. The same list object is reversed, so no

second list is created.

```
def reverse_in_place(lst):  
    left = 0  
    right = len(lst) - 1  
    while left < right:  
        lst[left], lst[right] = lst[right], lst[left]  
        left += 1  
        right -= 1  
  
lst = []  
n = int(input('How many elements: '))  
for i in range(n):  
    lst.append(input('Enter element: '))  
reverse_in_place(lst)  
print('Reversed list:', lst)
```

Exam tip

Do not use `lst[::-1]` here because that creates a new reversed list.

Final Answer: The function reverses the passed list by swapping elements inside the same list.

EXPERT'S SOLUTION : *Ishita Roy, M.Sc Computer Science, University of Hyderabad*

Scoring angle. List questions reward exact tracing. Write the index, method effect and final list state before the answer.

- Step 1.** Identify whether the question asks for output tracing, a concept difference or a program.
- Step 2.** Set left to 0 and right to $\text{len}(\text{lst})-1$.
- Step 3.** Swap `lst[left]` and `lst[right]`.
- Step 4.** Increase left and decrease right.
- Step 5.** Repeat while left is less than right.
- Step 6.** The same list object is reversed, so no second list is created.
- Step 7.** End with the exact list value, code statement or program result required by the question.

For programs, keep the function small and let the main part handle input and printing. For output questions, never skip the difference between returned value and in-place change.

Final Answer: The function reverses the passed list by swapping elements inside the same list.

Related Collegedunia Resources

Same chapter, other resources:

- [Class 11 Computer Science - All Chapters](#)

Continue learning:

- [Ch 8: Strings](#)
- [Class 11 Computer Science - All Chapters](#)

End of NCERT Solutions for Class 11 Computer Science, Chapter 9: Lists.