

1

main.py	Output
<pre> 1 test_cases = [[], [1], [7, 7, 7, 7], [-5, -1, -3, -2, -4]] 2 for case in test_cases: 3 case.sort(reverse=case[0] < 0 if case else False) 4 print(case) 5 </pre>	<pre> [] [1] [7, 7, 7, 7] [-1, -2, -3, -4, -5] === Code Execution Successful === </pre>

2

main.py	Output
<pre> 1 def selection_sort(arr): 2 n = len(arr) 3 for i in range(n): 4 min_idx = i 5 for j in range(i+1, n): 6 if arr[j] < arr[min_idx]: 7 min_idx = j 8 arr[i], arr[min_idx] = arr[min_idx], arr[i] 9 return arr 10 input_arr = [5, 2, 9, 1, 5, 6] 11 sorted_arr = selection_sort(input_arr) 12 print(sorted_arr) </pre>	<pre> [1, 2, 5, 5, 6, 9] === Code Execution Successful === </pre>

3

main.py	Output
<pre> 1 def bubbleSort(arr): 2 n = len(arr) 3 for i in range(n): 4 swapped = False 5 for j in range(0, n-i-1): 6 if arr[j] > arr[j+1]: 7 arr[j], arr[j+1] = arr[j+1], arr[j] 8 swapped = True 9 if (swapped == False): 10 break 11 if __name__ == "__main__": 12 arr = [64, 34, 25, 12, 22, 11, 90] 13 bubbleSort(arr) 14 print("Sorted array:") 15 for i in range(len(arr)): 16 print("%d" % arr[i], end=" ") 17 </pre>	<pre> Sorted array: 11 12 22 25 34 64 90 === Code Execution Successful === </pre>

5

main.py	Output
<pre> 1 def find_kth_missing(arr, k): 2 return next(x for x in range(1, arr[-1] + k + 1) if x not in arr) 3 arr = [2, 3, 4, 7, 11] 4 k = 5 5 output = find_kth_missing(arr, k) 6 print(output) 7 </pre>	<pre> 1 === Code Execution Successful === </pre>

4

main.py	Output
<pre> 1 def insertion_sort(arr): 2 for i in range(1, len(arr)): 3 key = arr[i] 4 j = i - 1 5 while j >= 0 and key < arr[j]: 6 arr[j + 1] = arr[j] 7 j -= 1 8 arr[j + 1] = key 9 return arr 10 array1 = [3, 1, 4, 1, 5, 9, 2, 6, 5, 3] 11 array2 = [5, 5, 5, 5, 5] 12 array3 = [2, 3, 1, 3, 2, 1, 1, 3] 13 sorted_array1 = insertion_sort(array1) 14 sorted_array2 = insertion_sort(array2) 15 sorted_array3 = insertion_sort(array3) 16 print(sorted_array1) 17 print(sorted_array2) 18 print(sorted_array3) </pre>	<pre> [1, 1, 2, 3, 3, 4, 5, 5, 6, 9] [5, 5, 5, 5, 5] [1, 1, 1, 2, 2, 3, 3, 3] === Code Execution Successful === </pre>

6

main.py	Output
<pre> 1 def findPeak(arr, n) : 2 if (n == 1) : 3 return 0 4 if (arr[0] >= arr[1]) : 5 return 0 6 if (arr[n - 1] >= arr[n - 2]) : 7 return n - 1 8 for i in range(1, n - 1) : 9 if (arr[i] >= arr[i - 1] and arr[i] >= arr[i + 1]) : 10 return i 11 arr = [1, 3, 20, 4, 1, 0] 12 n = len(arr) 13 print("Index of a peak point is", findPeak(arr, n)) </pre>	<pre> Index of a peak point is 2 === Code Execution Successful === </pre>

7

main.py	Output
<pre> 1 def findPeak(arr, n) : 2 if (n == 1) : 3 return 0 4 if (arr[0] >= arr[1]) : 5 return 0 6 if (arr[n - 1] >= arr[n - 2]) : 7 return n - 1 8 for i in range(1, n - 1) : 9 if (arr[i] >= arr[i - 1] and arr[i] >= arr[i + 1]) : 10 return i 11 arr = [1, 3, 20, 4, 1, 0] 12 n = len(arr) 13 print("Index of a peak point is", findPeak(arr, n)) </pre>	<pre> Index of a peak point is 2 === Code Execution Successful === </pre>

8

<pre> words = ["mass", "as", "hero", "superhero"] output = [word for word in words if any(word in other_word for other_word in words if word != other_word)] print(output) </pre>	<pre> ['as', 'hero'] === Code Execution Successful === </pre>
---	--