



programiz.com



Netflix



CSA0619-DAA/day2 - D...



Online C Compiler - Pro...



FREE AI Code Generator:...



Python Compiler (Interpr...



Python Online Compiler

Programiz PRO &gt;

main.py



Run

Output

Clear

```
1 empty_list = []
2 print(empty_list)
3
4 one_element_list = [42]
5 print(one_element_list)
6
7 identical_elements_list = [7] * 5
8 print(identical_elements_list)
9
10 negative_numbers_list = [-1, -2, -3, -4, -5]
11 print(negative_numbers_list)
```

```
[]
[42]
[7, 7, 7, 7, 7]
[-1, -2, -3, -4, -5]

=== Code Execution Successful ===
```



programiz.com



Netflix



CSA0619-DAA/day2 - D...



Online C Compiler - Pro...



FREE AI Code Generator:...



Python Compiler (Interpr...



Python Online Compiler

Programiz PRO &gt;

main.py



Run

Output

Clear

```
1 def selection_sort(arr):
2     n = len(arr)
3     for i in range(n):
4         min_index = i
5         for j in range(i + 1, n):
6             if arr[j] < arr[min_index]:
7                 min_index = j
8         arr[i], arr[min_index] = arr[min_index],
            arr[i]
9     return arr
10 random_array = [5, 2, 9, 1, 5, 6]
11 sorted_random_array = selection_sort(random_array)
12 print("Sorted Random Array:", sorted_random_array)
13 reverse_sorted_array = [10, 8, 6, 4, 2]
14 reverse_array = selection_sort(reverse_sorted_array)
15 print("Sorted Reverse Sorted Array:",
        sorted_reverse_array)
16 already_sorted= [1, 2, 3, 4, 5]
17 sorted_already= selection_sort(already_sorted_array)
18 print("Sorted Already Sorted Array:",
        sorted_already_sorted_array)
```

```
Sorted Random Array: [1, 2, 5, 5, 6, 9]
Sorted Reverse Sorted Array: [2, 4, 6, 8, 10]
Sorted Already Sorted Array: [1, 2, 3, 4, 5]
```

```
=== Code Execution Successful ===
```

main.py



Run

Output

Clear

```
1 def bubble_sort(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 not swapped:
10             break
11     return arr
12
13 arr = [64, 34, 25, 12, 22, 11, 90]
14 sorted_arr = bubble_sort(arr)
15 print("Sorted array:", sorted_arr)
16
```

Sorted array: [11, 12, 22, 25, 34, 64, 90]

=== Code Execution Successful ===



programiz.com



Netflix



CSA0619-DAA/day2 - D...



Online C Compiler - Pro...



FREE AI Code Generator:...



Python Compiler (Interpr...



Python Online Compiler

Programiz PRO >

main.py



Run

Output

Clear

```
1 def optimized_sort(arr):
2     return sorted(arr)
3
4 print(optimized_sort([64, 25, 12, 22, 11]))
5 print(optimized_sort([29, 10, 14, 37, 13]))
6 print(optimized_sort([3, 5, 2, 1, 4]))
7 print(optimized_sort([1, 2, 3, 4, 5]))
8
```

[11, 12, 22, 25, 64]

[10, 13, 14, 29, 37]

[1, 2, 3, 4, 5]

[1, 2, 3, 4, 5]

=== Code Execution Successful ===



programiz.com



Netflix



CSA0619-DAA/day2 - D...



Online C Compiler - Pro...



FREE AI Code Generator:...



Python Compiler (Interpr...



Python Online Compiler

Programiz PRO &gt;

main.py



Run

Output

Clear

```
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 print(insertion_sort([3, 1, 4, 1, 5, 9, 2, 6, 5, 3]))
11 print(insertion_sort([5, 5, 5, 5, 5]))
12 print(insertion_sort([2, 3, 1, 3, 2, 1, 1, 3]))
```

13

14

[1, 1, 2, 3, 3, 4, 5, 5, 6, 9]

[5, 5, 5, 5, 5]

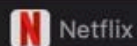
[1, 1, 1, 2, 2, 3, 3, 3]

=== Code Execution Successful ===





programiz.com



CSA0619-DAA/day2 - D...



Online C Compiler - Pro...



FREE AI Code Generator:...



Python Compiler (Interpr...



Python Online Compiler

Programiz PRO &gt;

main.py



Run

Output

Clear

```
1 def findKthPositive(arr, k):
2     missing_count = 0
3     current = 1
4     index = 0
5
6     while missing_count < k:
7         if index < len(arr) and arr[index] == current
8             :
9             index += 1
10        else:
11            missing_count += 1
12            if missing_count == k:
13                return current
14 print(findKthPositive([2, 3, 4, 7, 11], 5))
15 print(findKthPositive([1, 2, 3, 4], 2))
```

```
1
1
=== Code Execution Successful ===
```



programiz.com



Netflix



CSA0619-DAA/day2 - D...



Online C Compiler - Pro...



FREE AI Code Generator:...



Python Compiler (Interpr...



Python Online Compiler

Programiz PRO &gt;

main.py



Run

Output

Clear

```
1 def findPeakElement(nums):
2     left, right = 0, len(nums) - 1
3
4     while left < right:
5         mid = (left + right) // 2
6         if nums[mid] < nums[mid + 1]:
7             left = mid + 1
8         else:
9             right = mid
10
11     return left
12 print(findPeakElement([1, 2, 3, 1]))
```

2

```
=== Code Execution Successful ===
```



programiz.com



Netflix



CSA0619-DAA/day2 - D...



Online C Compiler - Pro...



FREE AI Code Generator:...



Python Compiler (Interpr...



Python Online Compiler

Programiz PRO &gt;

main.py



Run

Output

Clear

```
1 def strStr(haystack: str, needle: str) -> int:
2     return haystack.find(needle)
3
4 haystack1 = "sadbutsad"
5 needle1 = "sad"
6 output1 = strStr(haystack1, needle1)
7 print(output1)
8 haystack2 = "leetcode"
9 needle2 = "leeto"
10 output2 = strStr(haystack2, needle2)
11 print(output2)
12
```

0

-1

=== Code Execution Successful ===





programiz.com



Netflix



CSA0619-DAA/day2 - D...



Online C Compiler - Pro...



FREE AI Code Generator:...



Python Compiler (Interpr...



Programiz Python Online Compiler

Programiz PRO &gt;

main.py



Run

Output

Clear

```
1 def find_substrings(words):
2     result = []
3     for i in range(len(words)):
4         for j in range(len(words)):
5             if i != j and words[i] in words[j]:
6                 result.append(words[i])
7                 break
8     return list(set(result))
9
10 print(find_substrings(["mass","as","hero"
11                        ,"superhero"]))
11 print(find_substrings(["leetcode","et","code"]))
12 print(find_substrings(["blue","green","bu"]))
```

```
['hero', 'as']
['et', 'code']
[]
```

```
=== Code Execution Successful ===
```