Online Python Compiler (Interpre | DAY-5[1].pdf

https://www.programiz.com/python-programming/online-compiler/

Programiz
Python Online Compiler

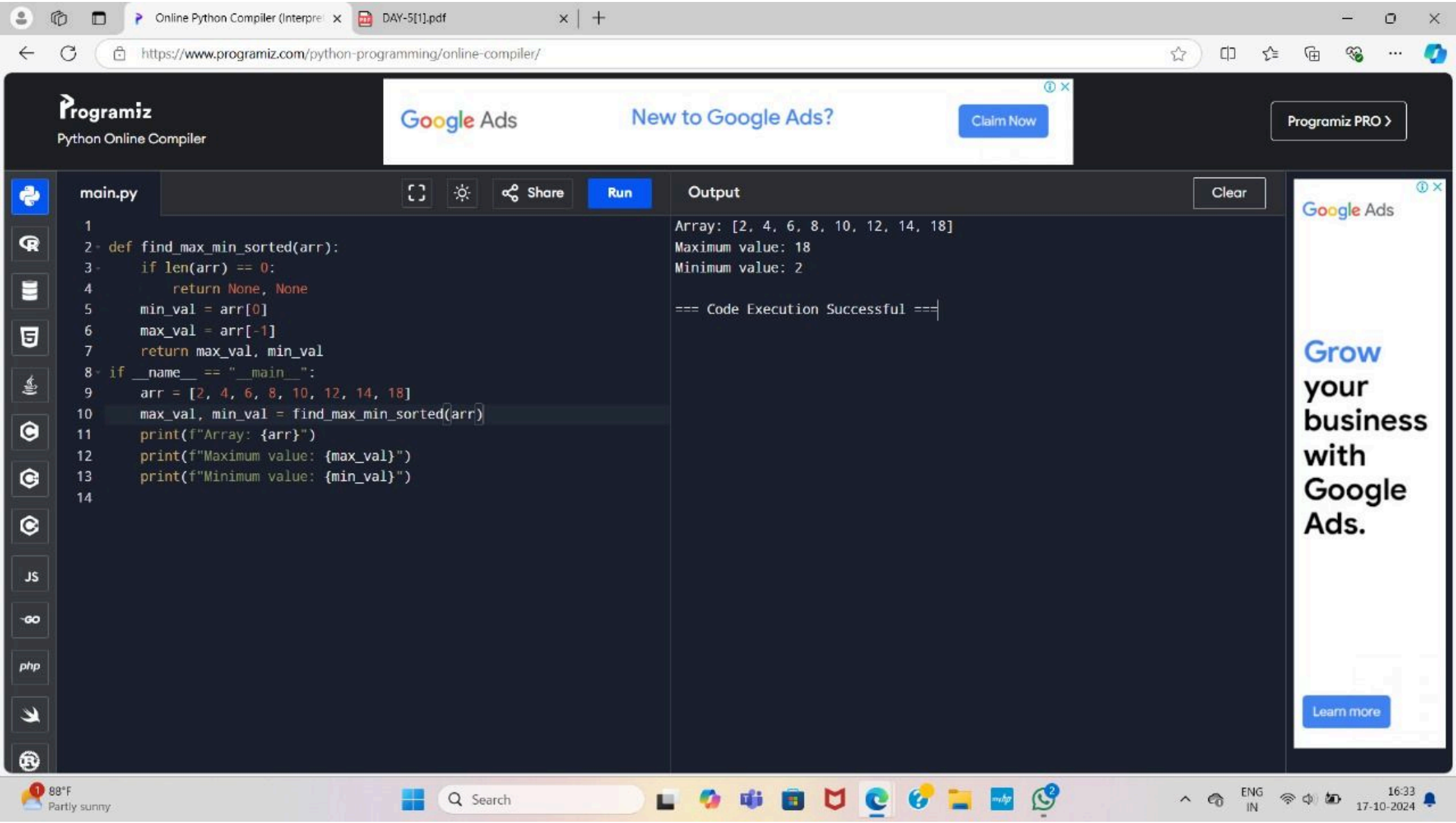
main.py | Share | Run

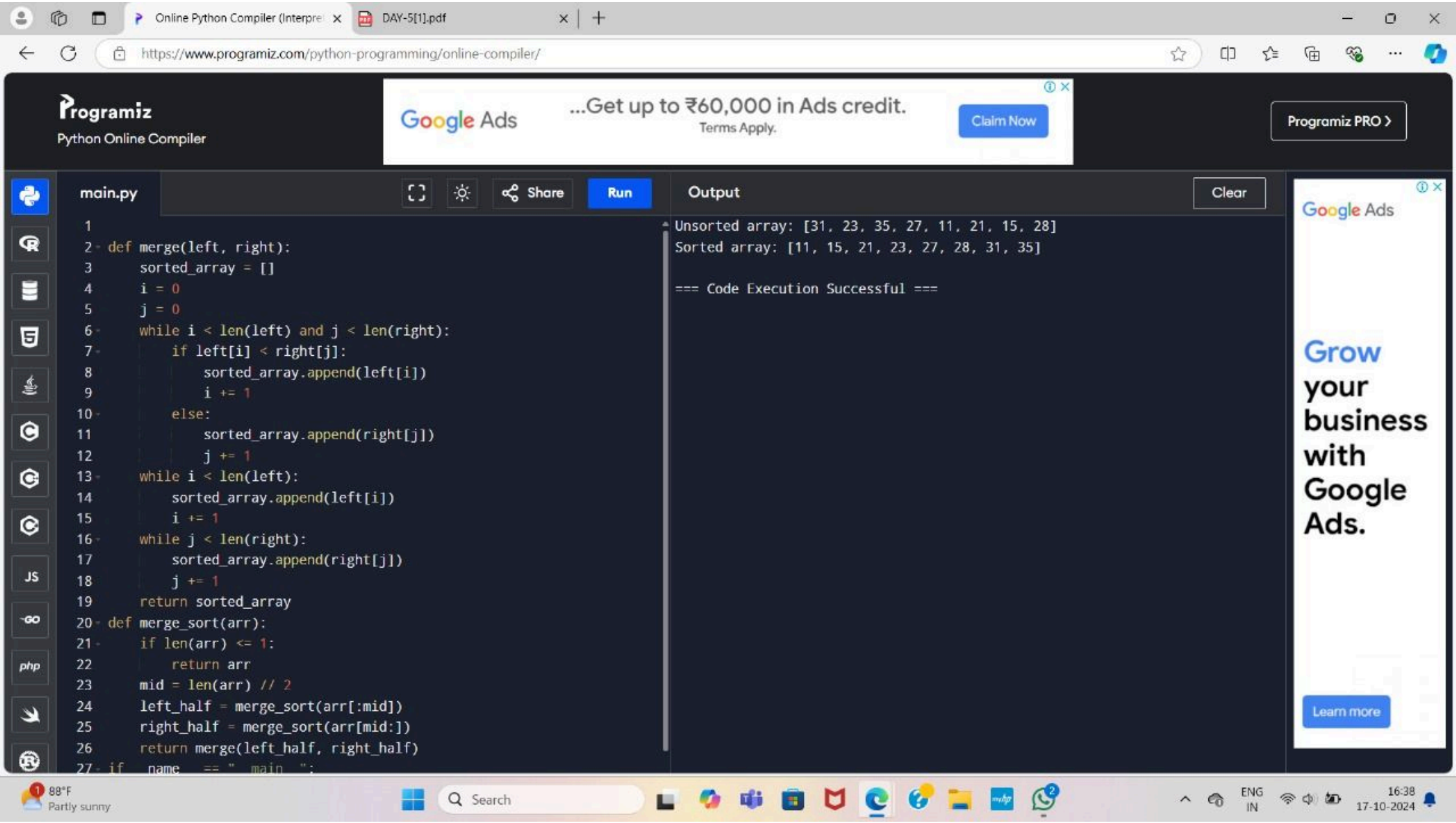
```
1
2  def find_max_min(arr):
3      if len(arr) == 0:
4          return None, None
5      max_val = arr[0]
6      min_val = arr[0]
7      for num in arr:
8          if num > max_val:
9              max_val = num
10         if num < min_val:
11             min_val = num
12
13     return max_val, min_val
14 if __name__ == "__main__":
15     arr = [10, 20, 5, 8, 25, 30, 2, 19]
16     max_val, min_val = find_max_min(arr)
17     print(f"Array: {arr}")
18     print(f"Maximum value: {max_val}")
19     print(f"Minimum value: {min_val}")
20
```

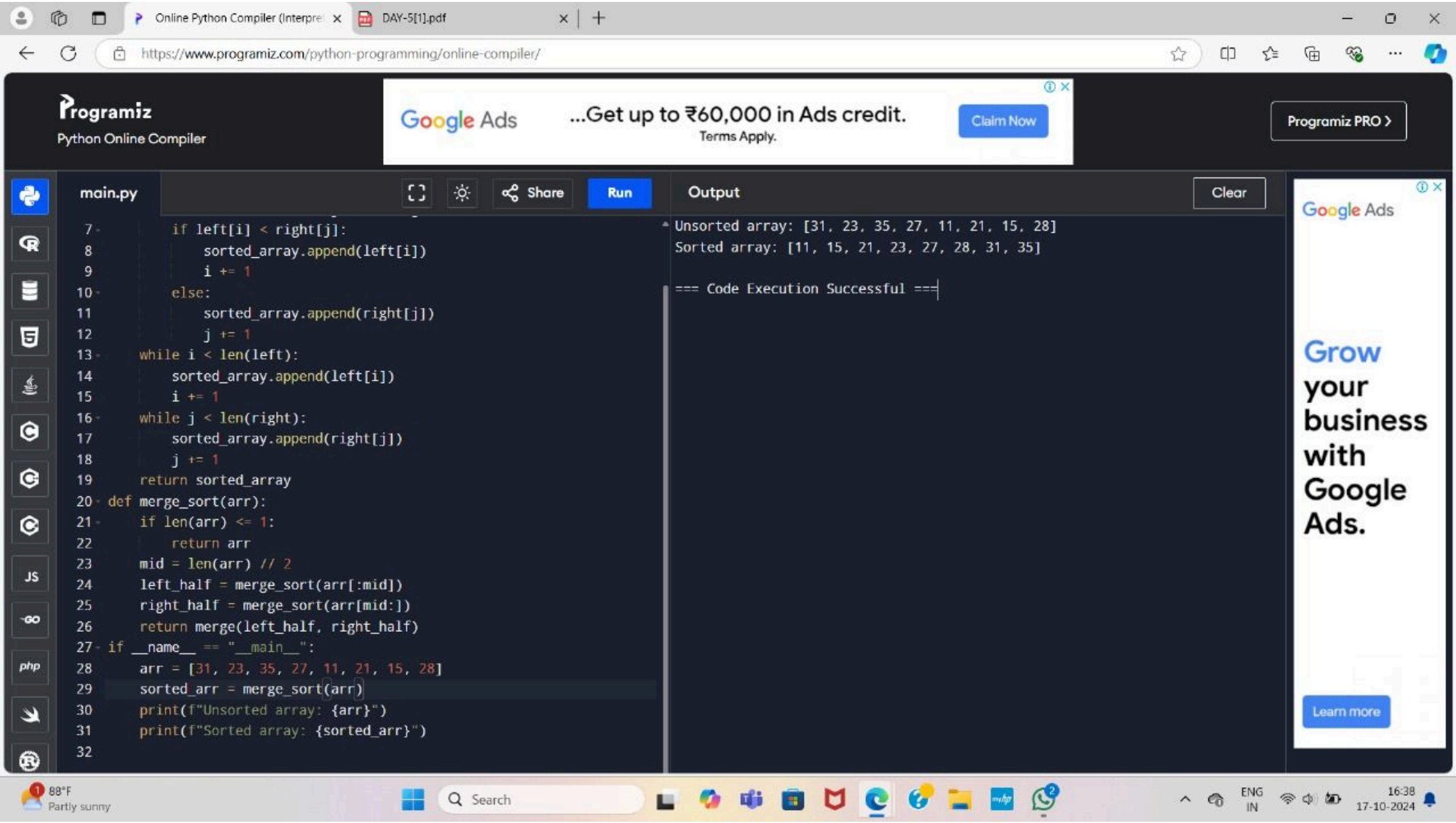
Output | Clear

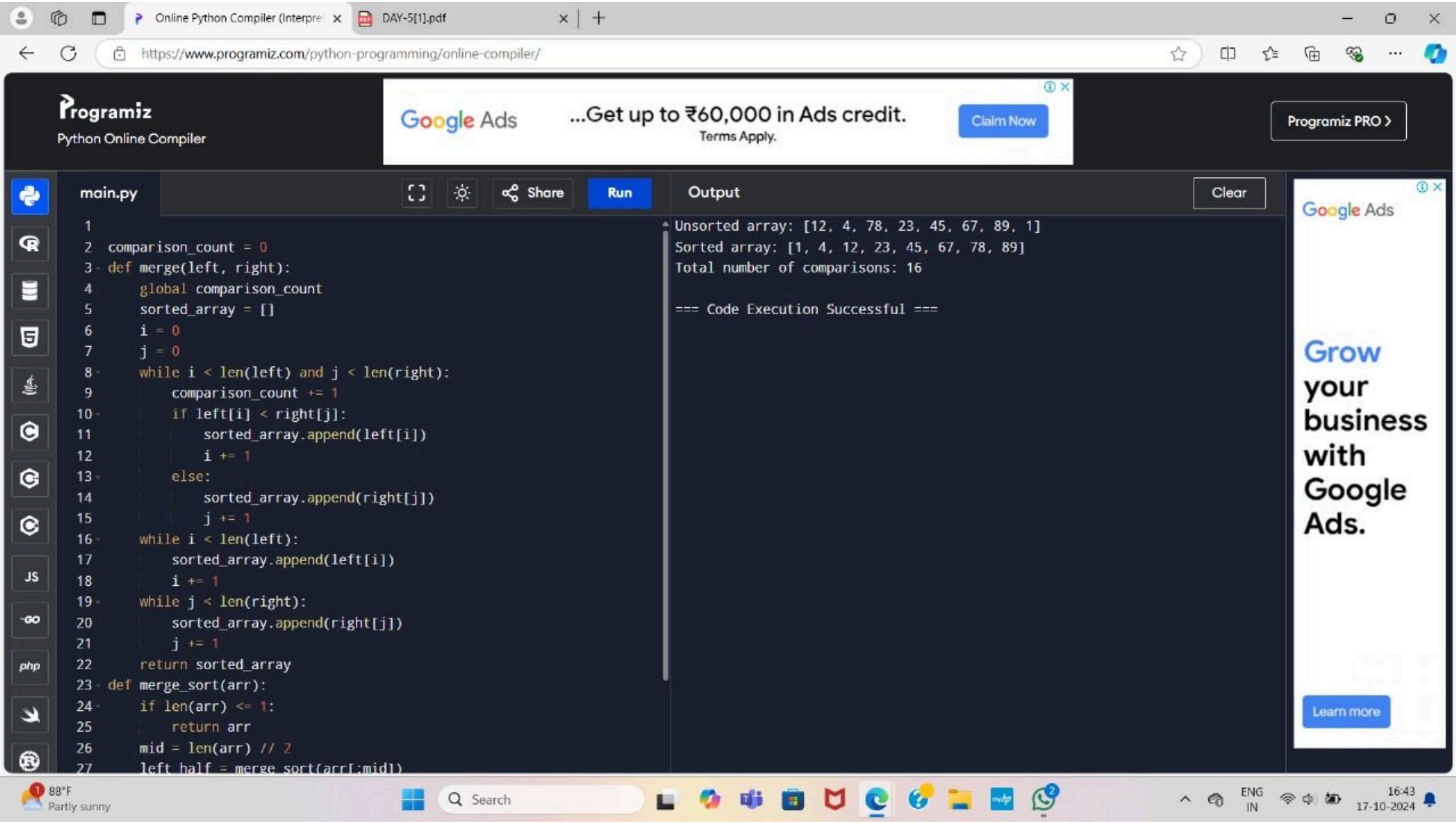
```
Array: [10, 20, 5, 8, 25, 30, 2, 19]
Maximum value: 30
Minimum value: 2

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

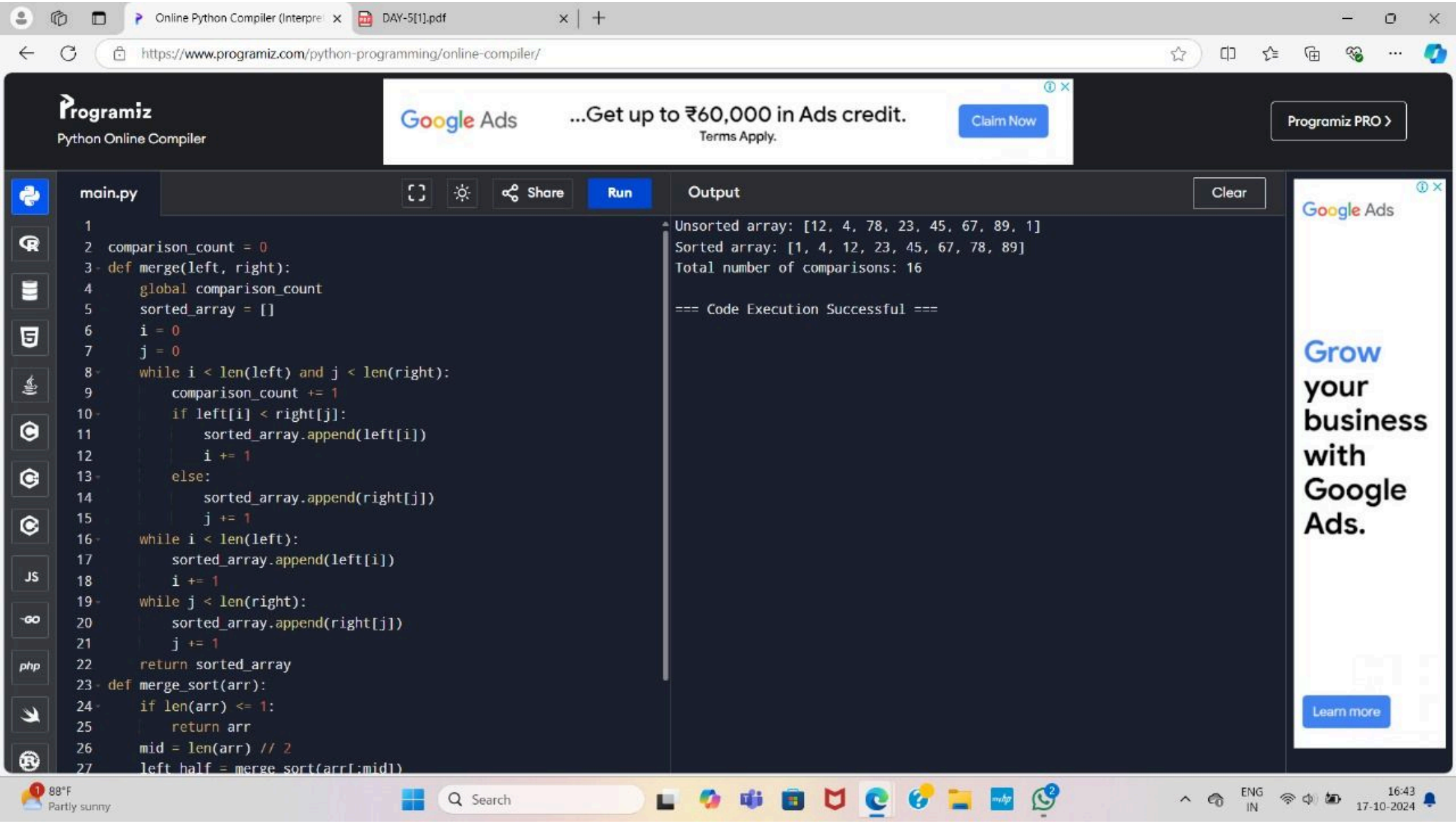

88°F Partly sunny | Search | ENG IN | 16:30 17-10-2024


Online Python Compiler (Interpre
DAY-5[1].pdf
https://www.programiz.com/python-programming/online-compiler/
Programiz
Python Online Compiler
Google Ads
New to Google Ads?
Claim Now
Programiz PRO
main.py
Share
Run
Output
Clear
def find_max_min_sorted(arr):
if len(arr) == 0:
return None, None
min_val = arr[0]
max_val = arr[-1]
return max_val, min_val
if __name__ == "__main__":
arr = [2, 4, 6, 8, 10, 12, 14, 18]
max_val, min_val = find_max_min_sorted(arr)
print(f"Array: {arr}")
print(f"Maximum value: {max_val}")
print(f"Minimum value: {min_val}")
Array: [2, 4, 6, 8, 10, 12, 14, 18]
Maximum value: 18
Minimum value: 2
=== Code Execution Successful ===
Google Ads
Grow your business with Google Ads.
Learn more
88°F
Partly sunny
Search
ENG
IN
16:33
17-10-2024


Online Python Compiler (Interpre
DAY-5[1].pdf
https://www.programiz.com/python-programming/online-compiler/
Programiz
Python Online Compiler
Google Ads
...Get up to ₹60,000 in Ads credit.
Terms Apply.
Claim Now
Programiz PRO
main.py
Share
Run
Output
Clear
def merge(left, right):
sorted_array = []
i = 0
j = 0
while i < len(left) and j < len(right):
if left[i] < right[j]:
sorted_array.append(left[i])
i += 1
else:
sorted_array.append(right[j])
j += 1
while i < len(left):
sorted_array.append(left[i])
i += 1
while j < len(right):
sorted_array.append(right[j])
j += 1
return sorted_array
def merge_sort(arr):
if len(arr) <= 1:
return arr
mid = len(arr) // 2
left_half = merge_sort(arr[:mid])
right_half = merge_sort(arr[mid:])
return merge(left_half, right_half)
Unsorted array: [31, 23, 35, 27, 11, 21, 15, 28]
Sorted array: [11, 15, 21, 23, 27, 28, 31, 35]
=== Code Execution Successful ===
Google Ads
Grow your business with Google Ads.
Learn more
88°F
Partly sunny
Search
ENG
IN
16:38
17-10-2024


Online Python Compiler (Interpre
DAY-5[1].pdf
https://www.programiz.com/python-programming/online-compiler/
Programiz
Python Online Compiler
Google Ads
...Get up to ₹60,000 in Ads credit.
Terms Apply.
Claim Now
Programiz PRO
main.py
Share
Run
Output
Clear
if left[i] < right[j]:
sorted_array.append(left[i])
i += 1
else:
sorted_array.append(right[j])
j += 1
while i < len(left):
sorted_array.append(left[i])
i += 1
while j < len(right):
sorted_array.append(right[j])
j += 1
return sorted_array
def merge_sort(arr):
if len(arr) <= 1:
return arr
mid = len(arr) // 2
left_half = merge_sort(arr[:mid])
right_half = merge_sort(arr[mid:])
return merge(left_half, right_half)
if __name__ == "__main__":
arr = [31, 23, 35, 27, 11, 21, 15, 28]
sorted_arr = merge_sort(arr)
print(f"Unsorted array: {arr}")
print(f"Sorted array: {sorted_arr}")
Unsorted array: [31, 23, 35, 27, 11, 21, 15, 28]
Sorted array: [11, 15, 21, 23, 27, 28, 31, 35]
=== Code Execution Successful ===
Google Ads
Grow your business with Google Ads.
Learn more
88°F
Partly sunny
Search
ENG
IN
16:38
17-10-2024


Online Python Compiler (Interpre
DAY-5[1].pdf
https://www.programiz.com/python-programming/online-compiler/
Programiz
Python Online Compiler
Google Ads
...Get up to ₹60,000 in Ads credit.
Terms Apply.
Claim Now
Programiz PRO
main.py
Share
Run
Output
Clear
comparison_count = 0
def merge(left, right):
global comparison_count
sorted_array = []
i = 0
j = 0
while i < len(left) and j < len(right):
comparison_count += 1
if left[i] < right[j]:
sorted_array.append(left[i])
i += 1
else:
sorted_array.append(right[j])
j += 1
while i < len(left):
sorted_array.append(left[i])
i += 1
while j < len(right):
sorted_array.append(right[j])
j += 1
return sorted_array
def merge_sort(arr):
if len(arr) <= 1:
return arr
mid = len(arr) // 2
left_half = merge_sort(arr[:mid])
Unsorted array: [12, 4, 78, 23, 45, 67, 89, 1]
Sorted array: [1, 4, 12, 23, 45, 67, 78, 89]
Total number of comparisons: 16
=== Code Execution Successful ===
Google Ads
Grow your business with Google Ads.
Learn more
88°F
Partly sunny
Search
ENG
IN
16:43
17-10-2024


Online Python Compiler (Interpre
DAY-5[1].pdf
https://www.programiz.com/python-programming/online-compiler/
Programiz
Python Online Compiler
Google Ads
...Get up to ₹60,000 in Ads credit.
Terms Apply.
Claim Now
Programiz PRO
main.py
Share
Run
Output
Clear
comparison_count = 0
def merge(left, right):
global comparison_count
sorted_array = []
i = 0
j = 0
while i < len(left) and j < len(right):
comparison_count += 1
if left[i] < right[j]:
sorted_array.append(left[i])
i += 1
else:
sorted_array.append(right[j])
j += 1
while i < len(left):
sorted_array.append(left[i])
i += 1
while j < len(right):
sorted_array.append(right[j])
j += 1
return sorted_array
def merge_sort(arr):
if len(arr) <= 1:
return arr
mid = len(arr) // 2
left_half = merge_sort(arr[:mid])
Unsorted array: [12, 4, 78, 23, 45, 67, 89, 1]
Sorted array: [1, 4, 12, 23, 45, 67, 78, 89]
Total number of comparisons: 16
=== Code Execution Successful ===
Google Ads
Grow your business with Google Ads.
Learn more
88°F
Partly sunny
Search
ENG
IN
16:43
17-10-2024

Online Python Compiler (Interpre | DAY-5[1].pdf

https://www.programiz.com/python-programming/online-compiler/

Programiz
Python Online Compiler

main.py | Share | Run

```
            i += 1
        else:
            sorted_array.append(right[j])
            j += 1
    while i < len(left):
        sorted_array.append(left[i])
        i += 1
    while j < len(right):
        sorted_array.append(right[j])
        j += 1
    return sorted_array
def merge_sort(arr):
    if len(arr) <= 1:
        return arr
    mid = len(arr) // 2
    left_half = merge_sort(arr[:mid])
    right_half = merge_sort(arr[mid:])
    return merge(left_half, right_half)
if __name__ == "__main__":
    arr = [12, 4, 78, 23, 45, 67, 89, 1]
    comparison_count = 0
    sorted_arr = merge_sort(arr)
    print(f"Unsorted array: {arr}")
    print(f"Sorted array: {sorted_arr}")
    print(f"Total number of comparisons: {comparison_count}")

```

Output | Clear

```
Unsorted array: [12, 4, 78, 23, 45, 67, 89, 1]
Sorted array: [1, 4, 12, 23, 45, 67, 78, 89]
Total number of comparisons: 16

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


88°F Partly sunny | Search | ENG IN | 16:43 17-10-2024

Online Python Compiler (Interpre | DAY-5[1].pdf

https://www.programiz.com/python-programming/online-compiler/

Programiz
Python Online Compiler

main.py

Share | Run

```python
# Function to perform partitioning
def partition(arr, low, high):
    pivot = arr[low]  # Choose the first element as the pivot
    left = low + 1
    right = high

    while True:
        # Move the left pointer to the right until a value greater than the pivot is found
        while left <= right and arr[left] <= pivot:
            left += 1
        # Move the right pointer to the left until a value smaller than the pivot is found
        while left <= right and arr[right] >= pivot:
            right -= 1
        # If the pointers cross, partitioning is done
        if left > right:
            break
        else:
            # Swap elements at left and right pointers
            arr[left], arr[right] = arr[right], arr[left]

    # Swap the pivot with the element at the right pointer
    arr[low], arr[right] = arr[right], arr[low]
    return right

# Function to perform Quick Sort recursively
```

Output | Clear

```
Initial unsorted array: [10, 16, 8, 12, 15, 6, 3, 9, 5]
Array after partitioning with pivot 10: [6, 5, 8, 9, 3, 10, 15, 12, 16]
Array after partitioning with pivot 6: [3, 5, 6, 9, 8, 10, 15, 12, 16]
Array after partitioning with pivot 3: [3, 5, 6, 9, 8, 10, 15, 12, 16]
Array after partitioning with pivot 9: [3, 5, 6, 8, 9, 10, 15, 12, 16]
Array after partitioning with pivot 15: [3, 5, 6, 8, 9, 10, 12, 15, 16]
Sorted array: [3, 5, 6, 8, 9, 10, 12, 15, 16]

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


88°F Partly sunny | Search | ENG IN | 16:47 17-10-2024

Online Python Compiler (Interpre | DAY-5[1].pdf

https://www.programiz.com/python-programming/online-compiler/

Programiz
Python Online Compiler

main.py | Share | Run

```
24
25 # Function to perform Quick Sort recursively
26 def quick_sort(arr, low, high):
27     if low < high:
28         # Perform partitioning
29         pivot_index = partition(arr, low, high)
30         print(f"Array after partitioning with pivot
               {arr[pivot_index]}: {arr}")
31
32         # Recursively apply Quick Sort on the left and right sub
               -arrays
33         quick_sort(arr, low, pivot_index - 1)
34         quick_sort(arr, pivot_index + 1, high)
35
36 # Test the function with the given unsorted array
37 if __name__ == "__main__":
38     # Unsorted array
39     arr = [10, 16, 8, 12, 15, 6, 3, 9, 5]
40     print(f"Initial unsorted array: {arr}")
41
42     # Perform Quick Sort
43     quick_sort(arr, 0, len(arr) - 1)
44
45     # Display the final sorted array
46     print(f"Sorted array: {arr}")
47
```

Output | Clear

```
Initial unsorted array: [10, 16, 8, 12, 15, 6, 3, 9, 5]
Array after partitioning with pivot 10: [6, 5, 8, 9, 3, 10, 15, 12, 16]
Array after partitioning with pivot 6: [3, 5, 6, 9, 8, 10, 15, 12, 16]
Array after partitioning with pivot 3: [3, 5, 6, 9, 8, 10, 15, 12, 16]
Array after partitioning with pivot 9: [3, 5, 6, 8, 9, 10, 15, 12, 16]
Array after partitioning with pivot 15: [3, 5, 6, 8, 9, 10, 12, 15, 16]
Sorted array: [3, 5, 6, 8, 9, 10, 12, 15, 16]

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


88°F Partly sunny | Search | ENG IN | 16:47 17-10-2024

Online Python Compiler (Interpre | DAY-5[1].pdf

https://www.programiz.com/python-programming/online-compiler/

Programiz
Python Online Compiler

main.py

Share | Run

```
1
2  def binary_search(arr, target):
3      low = 0
4      high = len(arr) - 1
5      comparison_count = 0
6      while low <= high:
7          comparison_count += 1
8          mid = (low + high) // 2
9          if arr[mid] == target:
10             print(f"Element {target} found at index {mid}")
11             print(f"Total number of comparisons: {comparison_count}")
12             return mid
13         elif arr[mid] < target:
14             low = mid + 1
15         else:
16             high = mid - 1
17     print(f"Element {target} not found")
18     print(f"Total number of comparisons: {comparison_count}")
19     return -1
20 if __name__ == "__main__":
21     arr = [5, 10, 15, 20, 25, 30, 35, 40, 45]  # Sorted array
22     target = 20
23     index = binary_search(arr, target)
24
```

Output | Clear

```
Element 20 found at index 3
Total number of comparisons: 4

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


88°F Partly sunny

Search

ENG IN

16:48 17-10-2024

Online Python Compiler (Interpre | DAY-5[1].pdf

https://www.programiz.com/python-programming/online-compiler/

Programiz
Python Online Compiler

Programiz PRO >

main.py | Share | Run

```python
def binary_search(arr, target):
    low = 0
    high = len(arr) - 1
    comparison_count = 0

    while low <= high:
        comparison_count += 1
        mid = (low + high) // 2
        print(f"Step {comparison_count}: mid-point index = {mid},
            mid-point value = {arr[mid]}")
        if arr[mid] == target:
            print(f"\nElement {target} found at index {mid}")
            print(f"Total number of comparisons: {comparison_count}"
                )
            return mid
        elif arr[mid] < target:
            low = mid + 1
            print(f"Element {target} is greater than mid-point value
                , moving to right half.")
        else:
            high = mid - 1
            print(f"Element {target} is less than mid-point value,
                moving to left half.")
    print(f"\nElement {target} not found in the array.")
    print(f"Total number of comparisons: {comparison_count}")
    return -1
if __name__ == "__main__":
```

Output | Clear

```
Step 1: mid-point index = 4, mid-point value = 25
Element 31 is greater than mid-point value, moving to right half.
Step 2: mid-point index = 6, mid-point value = 42
Element 31 is less than mid-point value, moving to left half.
Step 3: mid-point index = 5, mid-point value = 31

Element 31 found at index 5
Total number of comparisons: 3

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

88°F Partly sunny | Search | ENG IN | 16:51 17-10-2024

Online Python Compiler (Interpre | DAY-5[1].pdf

https://www.programiz.com/python-programming/online-compiler/

Programiz
Python Online Compiler

main.py

Share | Run

```
5
6   while low <= high:
7       comparison_count += 1
8       mid = (low + high) // 2
9       print(f"Step {comparison_count}: mid-point index = {mid},
            mid-point value = {arr[mid]}")
10      if arr[mid] == target:
11          print(f"\nElement {target} found at index {mid}")
12          print(f"Total number of comparisons: {comparison_count}"
                )
13          return mid
14      elif arr[mid] < target:
15          low = mid + 1
16          print(f"Element {target} is greater than mid-point value
                , moving to right half.")
17      else:
18          high = mid - 1
19          print(f"Element {target} is less than mid-point value,
                moving to left half.")
20  print(f"\nElement {target} not found in the array.")
21  print(f"Total number of comparisons: {comparison_count}")
22  return -1
23 if __name__ == "__main__":
24      arr = [3, 9, 14, 19, 25, 31, 42, 47, 53]
25      index = binary_search(arr, target)
26
```

Output | Clear

```
Step 1: mid-point index = 4, mid-point value = 25
Element 31 is greater than mid-point value, moving to right half.
Step 2: mid-point index = 6, mid-point value = 42
Element 31 is less than mid-point value, moving to left half.
Step 3: mid-point index = 5, mid-point value = 31

Element 31 found at index 5
Total number of comparisons: 3

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


88°F Partly sunny | Search | ENG IN | 16:51 17-10-2024

Online Python Compiler (Interpre | DAY-5[1].pdf

https://www.programiz.com/python-programming/online-compiler/

Programiz
Python Online Compiler

main.py | Share | Run

```python
import math

def k_closest_points(points, k):

    points_with_distance = [(math.sqrt(x**2 + y**2), (x, y)) for x, y
        in points]


    points_with_distance.sort(key=lambda x: x[0])


    closest_points = [point[1] for point in points_with_distance[:k]]

    return closest_points


if __name__ == "__main__":
    points = [[1, 2], [2, 3], [3, 1], [5, 4], [0, 1], [1, 1]]
    k = 3
    closest_points = k_closest_points(points, k)
    print(f"The {k} closest points to the origin are:
        {closest_points}")

```

Output | Clear

```
The 3 closest points to the origin are: [(0, 1), (1, 1), (1, 2)]

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


88°F Partly sunny | Search | ENG IN | 16:52 17-10-2024

Online Python Compiler (Interpre | DAY-5[1].pdf

https://www.programiz.com/python-programming/online-compiler/

Programiz
Python Online Compiler

main.py | Share | Run | Output | Clear

```python
from collections import defaultdict

def four_sum_count(A, B, C, D):
    count_ab = defaultdict(int)

    # Count sums of A and B
    for a in A:
        for b in B:
            count_ab[a + b] += 1

    total_tuples = 0

    # Count sums of C and D and check against count_ab
    for c in C:
        for d in D:
            total_tuples += count_ab[-(c + d)]

    return total_tuples

# Test the function with example lists
if __name__ == "__main__":
    A = [1, 2]
    B = [-2, -1]
    C = [-1, 2]
    D = [0, 2]

    result = four_sum_count(A, B, C, D)
```

```
Number of tuples (i, j, k, l) such that A[i] + B[j] + C[k] + D[l] = 0: 2

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


88°F Partly sunny | Search | ENG IN | 16:59 17-10-2024

Online Python Compiler (Interpre | DAY-5[1].pdf

https://www.programiz.com/python-programming/online-compiler/

Programiz
Python Online Compiler

main.py

Share | Run

```
5
6       # Count sums of A and B
7       for a in A:
8           for b in B:
9               count_ab[a + b] += 1
10
11      total_tuples = 0
12
13      # Count sums of C and D and check against count_ab
14      for c in C:
15          for d in D:
16              total_tuples += count_ab[-(c + d)]
17
18      return total_tuples
19
20  # Test the function with example lists
21  if __name__ == "__main__":
22      A = [1, 2]
23      B = [-2, -1]
24      C = [-1, 2]
25      D = [0, 2]
26
27      result = four_sum_count(A, B, C, D)
28      print(f"Number of tuples (i, j, k, l) such that A[i] + B[j] +
            C[k] + D[l] = 0: {result}")
29
```

Output | Clear

```
Number of tuples (i, j, k, l) such that A[i] + B[j] + C[k] + D[l] = 0: 2

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


88°F Partly sunny | Search | ENG IN | 16:59 17-10-2024