File Edit Format Run Options Window Help

```python
from itertools import combinations

def total_value(items, values):
    return sum(values[i] for i in items)

def is_feasible(items, weights, capacity):
    total_weight = sum(weights[i] for i in items)
    return total_weight <= capacity

def knapsack(weights, values, capacity):
    n = len(weights)
    best_value = 0
    best_selection = []
    for r in range(n + 1):
        for combo in combinations(range(n), r):
            if is_feasible(combo, weights, capacity):
                val = total_value(combo, values)
                if val > best_value:
                    best_value = val
                    best_selection = list(combo)
    return best_selection, best_value

weights1 = [2, 3, 1]
values1 = [4, 5, 3]
capacity1 = 4
selection1, value1 = knapsack(weights1, values1, capacity1)
print("Test Case 1:")
print("Optimal Selection:", selection1)
print("Total Value:", value1)
```

Ln: 30 Col: 0

IDLE Shell 3.13.7

File Edit Shell Debug Options Window Help

```
Python 3.13.7 (tags/v3.13.7:bcee1c3, Aug 14 2025, 14:15:11) [MSC v.1944 64 bit (
AMD64)] on win32
Enter "help" below or click "Help" above for more information.
>>>
========== RESTART: C:/Users/SUPRAJA/.py ==========
Test Case 1:
Optimal Selection: [1, 2]
Total Value: 8
>>>
```

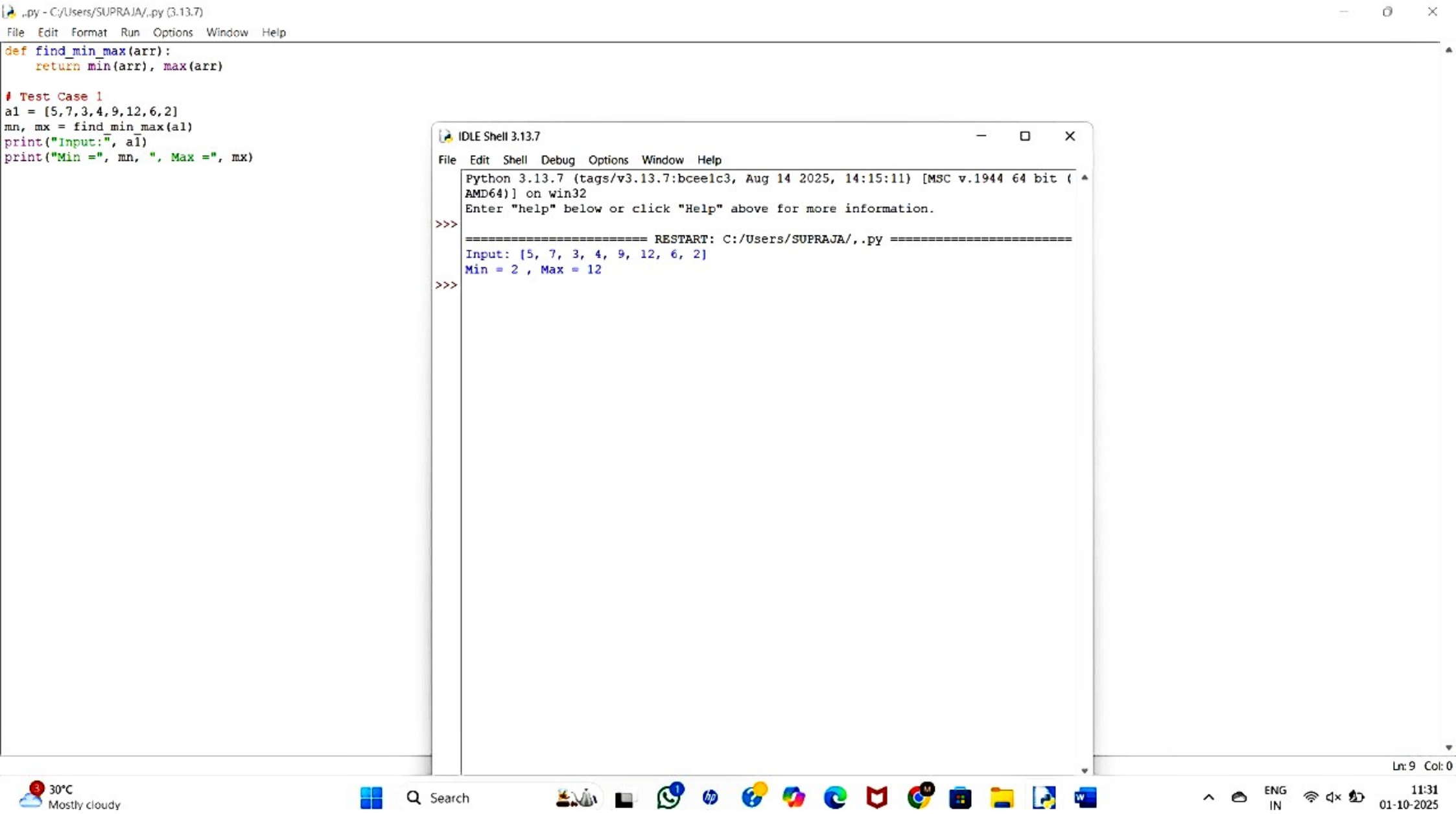
Ln: 8 Col: 0

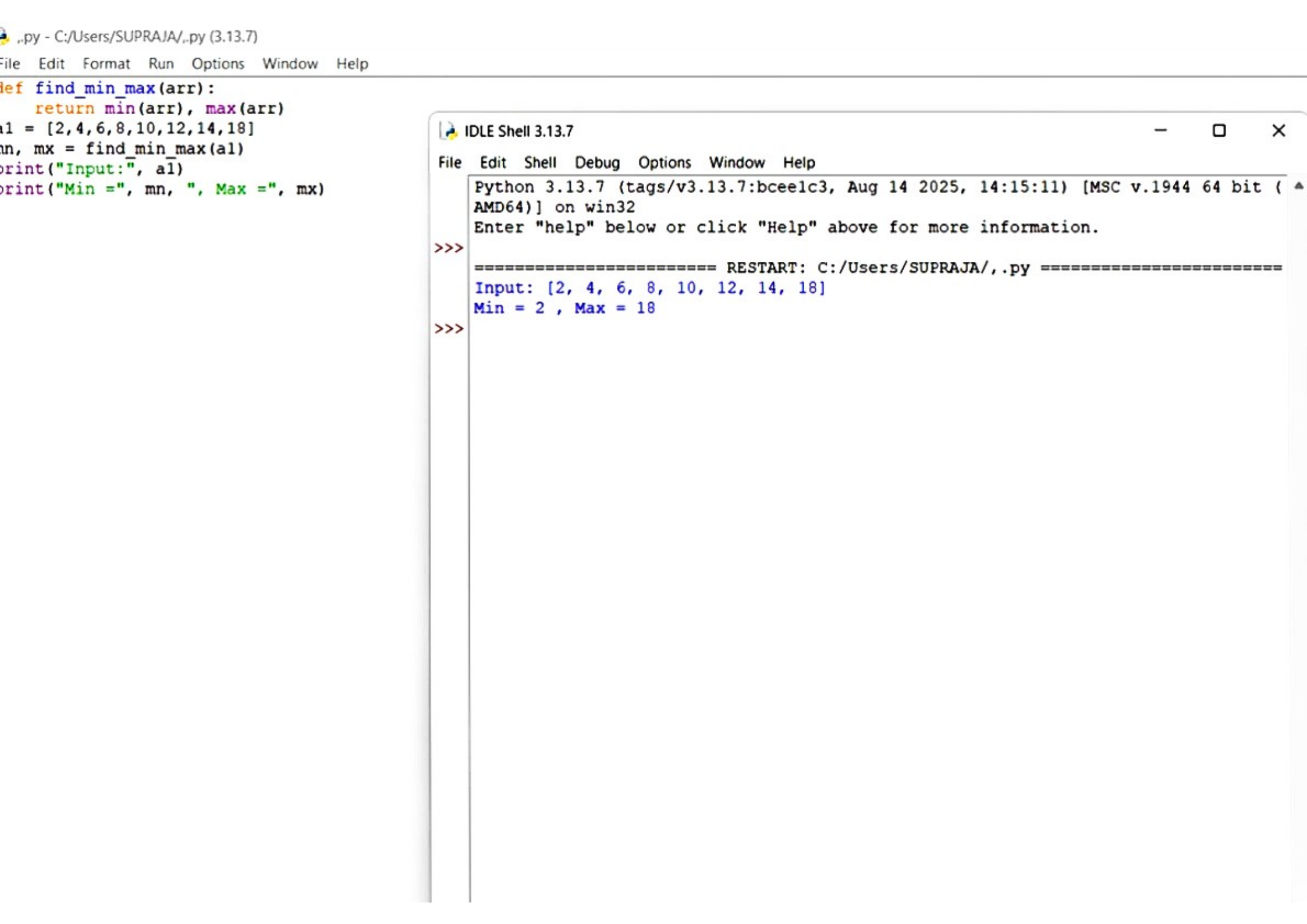
30°C Mostly cloudy

Search

ENG IN

11:38 01-10-2025


..py - C:/Users/SUPRAJA/..py (3.13.7)
File Edit Format Run Options Window Help
def find_min_max(arr):
return min(arr), max(arr)
# Test Case 1
a1 = [5,7,3,4,9,12,6,2]
mn, mx = find_min_max(a1)
print("Input:", a1)
print("Min =", mn, ", Max =", mx)
IDLE Shell 3.13.7
File Edit Shell Debug Options Window Help
Python 3.13.7 (tags/v3.13.7:bcee1c3, Aug 14 2025, 14:15:11) [MSC v.1944 64 bit (
AMD64)] on win32
Enter "help" below or click "Help" above for more information.
>>>
========== RESTART: C:/Users/SUPRAJA/,.py ==========
Input: [5, 7, 3, 4, 9, 12, 6, 2]
Min = 2 , Max = 12
>>>
Ln: 9 Col: 0
30°C
Mostly cloudy
Search
ENG
IN
11:31
01-10-2025


,.py - C:/Users/SUPRAJA/,.py (3.13.7)
File Edit Format Run Options Window Help
def find_min_max(arr):
return min(arr), max(arr)
a1 = [2,4,6,8,10,12,14,18]
mn, mx = find_min_max(a1)
print("Input:", a1)
print("Min =", mn, ", Max =", mx)
IDLE Shell 3.13.7
File Edit Shell Debug Options Window Help
Python 3.13.7 (tags/v3.13.7:bcee1c3, Aug 14 2025, 14:15:11) [MSC v.1944 64 bit (
AMD64)] on win32
Enter "help" below or click "Help" above for more information.
>>>
========== RESTART: C:/Users/SUPRAJA/,.py ==========
Input: [2, 4, 6, 8, 10, 12, 14, 18]
Min = 2 , Max = 18
>>>

File Edit Format Run Options Window Help

```
def merge_sort(arr):
    if len(arr) <= 1:
        return arr
    mid = len(arr) // 2
    left = merge_sort(arr[:mid])
    right = merge_sort(arr[mid:])
    return merge(left, right)
def merge(left, right):
    result = []
    i = j = 0
    while i < len(left) and j < len(right):
        if left[i] < right[j]:
            result.append(left[i])
            i += 1
        else:
            result.append(right[j])
            j += 1
    result.extend(left[i:])
    result.extend(right[j:])
    return result
a1 = [31,23,35,27,11,21,15,28]
print("Input:", a1)
print("Sorted:", merge_sort(a1))
```

IDLE Shell 3.13.7

File Edit Shell Debug Options Window Help

```
Python 3.13.7 (tags/v3.13.7:bcee1c3, Aug 14 2025, 14:15:11) [MSC v.1944 64 bit (
AMD64)] on win32
Enter "help" below or click "Help" above for more information.
>>>
========== RESTART: C:/Users/SUPRAJA/.py ==========
Input: [31, 23, 35, 27, 11, 21, 15, 28]
Sorted: [11, 15, 21, 23, 27, 28, 31, 35]
>>>
```

Ln: 7 Col: 0

Ln: 6 Col: 33

30°C Mostly cloudy

Search

ENG IN

11:34 01-10-2025

File Edit Format Run Options Window Help

```
comparison_count = 0
def merge_sort(arr):
    if len(arr) <= 1:
        return arr
    mid = len(arr) // 2
    left = merge_sort(arr[:mid])
    right = merge_sort(arr[mid:])
    return merge(left, right)
def merge(left, right):
    global comparison_count
    result = []
    i = j = 0
    while i < len(left) and j < len(right):
        comparison_count += 1  # counting each comparison
        if left[i] < right[j]:
            result.append(left[i])
            i += 1
        else:
            result.append(right[j])
            j += 1

    result.extend(left[i:])
    result.extend(right[j:])
    return result
arr1 = [12,4,78,23,45,67,89,1]
comparison_count = 0
sorted_arr1 = merge_sort(arr1)
print("Input:", arr1)
print("Sorted:", sorted_arr1)
print("Comparisons:", comparison_count)
```

IDLE Shell 3.13.7

File Edit Shell Debug Options Window Help

```
Python 3.13.7 (tags/v3.13.7:bcee1c3, Aug 14 2025, 14:15:11) [MSC v.1944 64 bit (
AMD64)] on win32
Enter "help" below or click "Help" above for more information.
>>>
========== RESTART: C:/Users/SUPRAJA/.py ==========
Input: [31, 23, 35, 27, 11, 21, 15, 28]
Sorted: [11, 15, 21, 23, 27, 28, 31, 35]
>>>
========== RESTART: C:/Users/SUPRAJA/.py ==========
Input: [12, 4, 78, 23, 45, 67, 89, 1]
Sorted: [1, 4, 12, 23, 45, 67, 78, 89]
Comparisons: 16
>>>
```

Ln: 12 Col: 0

Ln: 17 Col: 18

30°C Mostly cloudy

Search

ENG IN

11:35 01-10-2025

File Edit Format Run Options Window Help

```
def quick_sort(arr, low, high):
    if low < high:
        pi = partition(arr, low, high)
        print("After partition with pivot", arr[pi], ":", arr)
        quick_sort(arr, low, pi - 1)
        quick_sort(arr, pi + 1, high)

def partition(arr, low, high):
    pivot = arr[low]
    left = low + 1
    right = high

    while True:
        while left <= right and arr[left] <= pivot:
            left += 1
        while left <= right and arr[right] > pivot:
            right -= 1
        if left > right:
            break
        arr[left], arr[right] = arr[right], arr[left]

    arr[low], arr[right] = arr[right], arr[low]
    return right
arr1 = [10,16,8,12,15,6,3,9,5]
print("Input:", arr1)
quick_sort(arr1, 0, len(arr1)-1)
print("Sorted:", arr1)
```

IDLE Shell 3.13.7

File Edit Shell Debug Options Window Help

```
Python 3.13.7 (tags/v3.13.7:bcee1c3, Aug 14 2025, 14:15:11) [MSC v.1944 64 bit (
AMD64)] on win32
Enter "help" below or click "Help" above for more information.
>>>
========== RESTART: C:/Users/SUPRAJA/.py ==========
Input: [10, 16, 8, 12, 15, 6, 3, 9, 5]
After partition with pivot 10 : [6, 5, 8, 9, 3, 10, 15, 12, 16]
After partition with pivot 6 : [3, 5, 6, 9, 8, 10, 15, 12, 16]
After partition with pivot 3 : [3, 5, 6, 9, 8, 10, 15, 12, 16]
After partition with pivot 9 : [3, 5, 6, 8, 9, 10, 15, 12, 16]
After partition with pivot 15 : [3, 5, 6, 8, 9, 10, 12, 15, 16]
Sorted: [3, 5, 6, 8, 9, 10, 12, 15, 16]
>>>
```

Ln: 12 Col: 0

Ln: 23 Col: 16

File Edit Format Run Options Window Help

```
def quick_sort(arr, low, high):
    if low < high:
        pi = partition(arr, low, high)
        print("After partition with pivot", arr[pi], ":", arr)
        quick_sort(arr, low, pi - 1)
        quick_sort(arr, pi + 1, high)

def partition(arr, low, high):
    mid = (low + high) // 2
    pivot = arr[mid]
    arr[mid], arr[high] = arr[high], arr[mid]  # move pivot to end
    i = low - 1
    for j in range(low, high):
        if arr[j] <= pivot:
            i += 1
            arr[i], arr[j] = arr[j], arr[i]
    arr[i+1], arr[high] = arr[high], arr[i+1]
    return i + 1
arr1 = [19,72,35,46,58,91,22,31]
print("Input:", arr1)
quick_sort(arr1, 0, len(arr1)-1)
print("Sorted:", arr1)
```

Ln: 18 Col: 16

IDLE Shell 3.13.7

File Edit Shell Debug Options Window Help

```
Python 3.13.7 (tags/v3.13.7:bcee1c3, Aug 14 2025, 14:15:11) [MSC v.1944 64 bit (
AMD64)] on win32
Enter "help" below or click "Help" above for more information.
>>>
========== RESTART: C:/Users/SUPRAJA/.py ==========
Input: [19, 72, 35, 46, 58, 91, 22, 31]
After partition with pivot 46 : [19, 35, 31, 22, 46, 91, 72, 58]
After partition with pivot 35 : [19, 22, 31, 35, 46, 91, 72, 58]
After partition with pivot 22 : [19, 22, 31, 35, 46, 91, 72, 58]
After partition with pivot 72 : [19, 22, 31, 35, 46, 58, 72, 91]
Sorted: [19, 22, 31, 35, 46, 58, 72, 91]
>>>
```

Ln: 11 Col: 0

Gold +1.20%

Search

ENG IN

11:46 01-10-2025

File Edit Format Run Options Window Help

```
def binary_search(arr, key):
    low, high = 0, len(arr) - 1
    comparisons = 0
    while low <= high:
        mid = (low + high) // 2
        comparisons += 1
        if arr[mid] == key:
            return mid + 1, comparisons
        elif arr[mid] < key:
            low = mid + 1
        else:
            high = mid - 1
    return -1, comparisons
arr1 = [5,10,15,20,25,30,35,40,45]
pos, comps = binary_search(arr1, 20)
print("Input:", arr1, " Search key: 20")
print("Position:", pos, " Comparisons:", comps)
```

Ln: 12 Col: 26

IDLE Shell 3.13.7

File Edit Shell Debug Options Window Help

```
Python 3.13.7 (tags/v3.13.7:bcee1c3, Aug 14 2025, 14:15:11) [MSC v.1944 64 bit (
AMD64)] on win32
Enter "help" below or click "Help" above for more information.
>>>
========== RESTART: C:/Users/SUPRAJA/.py ==========
Input: [5, 10, 15, 20, 25, 30, 35, 40, 45]  Search key: 20
Position: 4  Comparisons: 4
>>>
```

Ln: 7 Col: 0

Gold +1.20%

Search

ENG IN

11:48 01-10-2025

File Edit Format Run Options Window Help

```
def binary_search_steps(arr, key):
    low, high = 0, len(arr) - 1
    steps = []

    while low <= high:
        mid = (low + high) // 2
        steps.append((low, mid, high, arr[mid]))
        if arr[mid] == key:
            return mid + 1, steps   # +1 for 1-based index
        elif arr[mid] < key:
            low = mid + 1
        else:
            high = mid - 1

    return -1, steps
arr1 = [3,9,14,19,25,31,42,47,53]
key1 = 31
pos, steps = binary_search_steps(arr1, key1)
print("Input:", arr1, "Search key:", key1)
for s in steps:
    print(f"low={s[0]}, mid={s[1]}, high={s[2]}, arr[mid]={s[3]}")
print("Position:", pos)
```

IDLE Shell 3.13.7

File Edit Shell Debug Options Window Help

```
Python 3.13.7 (tags/v3.13.7:bcee1c3, Aug 14 2025, 14:15:11) [MSC v.1944 64 bit (
AMD64)] on win32
Enter "help" below or click "Help" above for more information.
>>>
========== RESTART: C:/Users/SUPRAJA/,.py ==========
Input: [3, 9, 14, 19, 25, 31, 42, 47, 53] Search key: 31
low=0, mid=4, high=8, arr[mid]=25
low=5, mid=6, high=8, arr[mid]=42
low=5, mid=5, high=5, arr[mid]=31
Position: 6
>>>
```

Ln: 10 Col: 0

Ln: 23 Col: 0

File Edit Format Run Options Window Help

```
def k_closest_points(points, k):
    points_sorted = sorted(points, key=lambda p: p[0]**2 + p[1]**2)
    return points_sorted[:k]
points1 = [[1,3],[-2,2],[5,8],[0,1]]
k1 = 2
print("Input:", points1, "k =", k1)
print("Output:", k_closest_points(points1, k1))
```

IDLE Shell 3.13.7

File Edit Shell Debug Options Window Help

```
Python 3.13.7 (tags/v3.13.7:bcee1c3, Aug 14 2025, 14:15:11) [MSC v.1944 64 bit (
AMD64)] on win32
Enter "help" below or click "Help" above for more information.
>>>
========== RESTART: C:/Users/SUPRAJA/,.py ==========
Input: [[1, 3], [-2, 2], [5, 8], [0, 1]] k = 2
Output: [[0, 1], [-2, 2]]
>>>
```

Ln: 7 Col: 0

Ln: 1 Col: 32

30°C Mostly cloudy

Search

ENG IN

11:52 01-10-2025