Introducing ChatGPT | OpenAI | ChatGPT | Online Python Compiler (Interp

programiz.com/python-programming/online-compiler/

Paused

Programiz
Python Online Compiler

Programiz PRO ›

main.py

Save Run

```python
def find_max_min(nums):
    if not nums:
        print("List is empty")
        return

    max_num = nums[0]
    min_num = nums[0]

    for num in nums[1:]:
        if num > max_num:
            max_num = num
        elif num < min_num:
            min_num = num

    print(f"Maximum number: {max_num}")
    print(f"Minimum number: {min_num}")
numbers = [5, 3, 8, 1, 9, 2]
find_max_min(numbers)
```

Output

Clear

```
Maximum number: 9
Minimum number: 1

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

Waiting for currency.prebid.org...

30° Search ENG IN 09:28 17-06-2024

Introducing ChatGPT | OpenAI | ChatGPT | Online Python Compiler (Interp

programiz.com/python-programming/online-compiler/

Paused

Programiz
Python Online Compiler

Programiz PRO ›

main.py

Save | Run

```
11
12      while i < len(L) and j < len(R):
13          if L[i] < R[j]:
14              arr[k] = L[i]
15              i += 1
16          else:
17              arr[k] = R[j]
18              j += 1
19          k += 1
20
21      while i < len(L):
22          arr[k] = L[i]
23          i += 1
24          k += 1
25
26      while j < len(R):
27          arr[k] = R[j]
28          j += 1
29          k += 1
30
31  # Input
32  arr = [12, 11, 13, 5, 6, 7]
33  merge_sort(arr)
34
35  # Output
36  print(f"Sorted array: {arr}")
```

Output | Clear

```
Sorted array: [5, 6, 7, 11, 12, 13]

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

30° Search ENG IN 09:35 17-06-2024

Introducing ChatGPT | OpenAI | ChatGPT | Online Python Compiler (Interp

programiz.com/python-programming/online-compiler/

Paused

Programiz
Python Online Compiler

Programiz PRO ›

main.py

Save | Run

Output | Clear

```python
def partition(arr, low, high):
    pivot = arr[high]
    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

def quick_sort(arr, low, high):
    if low < high:
        pi = partition(arr, low, high)
        quick_sort(arr, low, pi - 1)
        quick_sort(arr, pi + 1, high)

# Input
arr = [10, 7, 8, 9, 1, 5]
quick_sort(arr, 0, len(arr) - 1)

# Output
print(f"Sorted array: {arr}")
```

```
Sorted array: [1, 5, 7, 8, 9, 10]

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

30°

Search

ENG
IN

09:36
17-06-2024

Introducing ChatGPT | OpenAI | ChatGPT | Online Python Compiler (Interp

programiz.com/python-programming/online-compiler/

Paused

Programiz
Python Online Compiler

Programiz PRO ›

main.py

Save | Run

```python
def binary_search(arr, low, high, target):
    if high >= low:
        mid = (high + low) // 2
        if arr[mid] == target:
            return mid
        elif arr[mid] > target:
            return binary_search(arr, low, mid - 1, target)
        else:
            return binary_search(arr, mid + 1, high, target)
    else:
        return -1

# Input
arr = [2, 3, 4, 10, 40]
target = 10
result = binary_search(arr, 0, len(arr) - 1, target)

# Output
if result != -1:
    print(f"Element found at index {result}")
else:
    print("Element not found")
```

Output | Clear

```
Element found at index 3

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

30° Search ENG IN 09:37 17-06-2024

Introducing ChatGPT | OpenAI | ChatGPT | Online Python Compiler (Interp

programiz.com/python-programming/online-compiler/

Paused

Programiz
Python Online Compiler

Programiz PRO ›

main.py | Save | Run | Output | Clear

```
        :, mid:]

    M1 = strassen_multiply(A11 + A22, B11 + B22)
    M2 = strassen_multiply(A21 + A22, B11)
    M3 = strassen_multiply(A11, B12 - B22)
    M4 = strassen_multiply(A22, B21 - B11)
    M5 = strassen_multiply(A11 + A12, B22)
    M6 = strassen_multiply(A21 - A11, B11 + B12)
    M7 = strassen_multiply(A12 - A22, B21 + B22)

    C11 = M1 + M4 - M5 + M7
    C12 = M3 + M5
    C21 = M2 + M4
    C22 = M1 - M2 + M3 + M6

    C = np.vstack((np.hstack((C11, C12)), np.hstack((C21, C22))))

    return C

# Input
A = np.array([[1, 2], [3, 4]])
B = np.array([[5, 6], [7, 8]])
C = strassen_multiply(A, B)

# Output
print(f"Product matrix:\n{C}")
```

```
Product matrix:
[[19 22]
 [43 50]]

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

30° Search ENG IN 09:37 17-06-2024

Introducing ChatGPT | OpenAI | ChatGPT | Online Python Compiler (Interp

programiz.com/python-programming/online-compiler/

Paused

Programiz
Python Online Compiler

Programiz PRO ›

main.py

Save | Run

```python
def karatsuba(x, y):
    if x < 10 or y < 10:
        return x * y
    n = max(len(str(x)), len(str(y)))
    m = n // 2

    x1, x0 = divmod(x, 10**m)
    y1, y0 = divmod(y, 10**m)

    z2 = karatsuba(x1, y1)
    z0 = karatsuba(x0, y0)
    z1 = karatsuba(x1 + x0, y1 + y0) - z2 - z0

    return z2 * 10*(2*m) + z1 * 10*m + z0

# Input
x = 1234
y = 5678
result = karatsuba(x, y)

# Output
print(f"Product: {result}")
```

Output | Clear

```
Product: 11052

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

Waiting for cache...

30° Search ENG IN 09:38 17-06-2024

Introducing ChatGPT | OpenAI | ChatGPT | Online Python Compiler (Interp

programiz.com/python-programming/online-compiler/

Paused

Programiz
Python Online Compiler

Programiz PRO ›

main.py

Save Run

Output Clear

```
38      in_strip = []
39      for p in Py:
40          if abs(p[0] - midpoint) < d:
41              in_strip.append(p)
42
43      min_d_strip = d
44      strip_pair = min_pair
45      for i in range(len(in_strip)):
46          for j in range(i+1, min(i+7, len(in_strip))):
47              if dist(in_strip[i], in_strip[j]) < min_d_strip:
48                  min_d_strip = dist(in_strip[i], in_strip[j])
49                  strip_pair = (in_strip[i], in_strip[j])
50
51      if min_d_strip < d:
52          return min_d_strip, strip_pair
53      else:
54          return d, min_pair
55
56  # Input
57  points = [(2.1, 3.1), (5.4, 1.2), (1.1, 2.5), (6.7, 3.3), (3.5, 1.7)]
58  Px = sorted(points, key=lambda x: x[0])
59  Py = sorted(points, key=lambda x: x[1])
60  distance, pair = closest_pair(Px, Py)
61
62  # Output
63  print(f"The closest pair of points is {pair} with a distance of {distance:.2f}"
```

```
ERROR!
Traceback (most recent call last):
  File "<main.py>", line 60, in <module>
  File "<main.py>", line 28, in closest_pair
  File "<main.py>", line 18, in closest_pair
  File "<main.py>", line 11, in brute_force
  File "<main.py>", line 4, in dist
ValueError: math domain error

=== Code Exited With Errors ===
```

30° Search ENG IN 09:39 17-06-2024

Introducing ChatGPT | OpenAI | ChatGPT | Online Python Compiler (Interp

programiz.com/python-programming/online-compiler/

Paused

Programiz
Python Online Compiler

Programiz PRO ›

main.py | Save | Run | Output | Clear

```
4
5       medians = []
6       for i in range(0, len(arr), 5):
7           group = arr[i:i + 5]
8           medians.append(sorted(group)[len(group) // 2])
9
10      pivot = median_of_medians(medians, len(medians) // 2)
11
12      low = [x for x in arr if x < pivot]
13      high = [x for x in arr if x > pivot]
14      equal = [x for x in arr if x == pivot]
15
16      if k < len(low):
17          return median_of_medians(low, k)
18      elif k < len(low) + len(equal):
19          return pivot
20      else:
21          return median_of_medians(high, k - len(low) - len(equal))
22
23  # Input
24  arr = [12, 3, 5, 7, 4, 19, 26]
25  k = len(arr) // 2
26  median = median_of_medians(arr, k)
27
28  # Output
29  print(f"The median is {median}")
```

```
The median is 7

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

30° Search ENG IN 09:40 17-06-2024

Introducing ChatGPT | OpenAI | ChatGPT | Online Python Compiler (Interp

programiz.com/python-programming/online-compiler/

Paused

Programiz

Python Online Compiler

Programiz PRO ›

main.py

Save | Run

Output | Clear

```python
from itertools import combinations

def subset_sum(arr, target):
    n = len(arr)
    left_half = arr[:n//2]
    right_half = arr[n//2:]

    left_sums = set(sum(comb) for i in range(len(left_half) + 1) for comb in
        combinations(left_half, i))
    right_sums = set(sum(comb) for i in range(len(right_half) + 1) for comb in
        combinations(right_half, i))

    for s in left_sums:
        if target - s in right_sums:
            return True

    return False

# Input
arr = [3, 34, 4, 12, 5, 2]
target = 9
result = subset_sum(arr, target)

# Output
print(f"Subset with the given sum exists: {result}")
```

```
Subset with the given sum exists: True

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

Waiting for match.deepintent.com...

30°

Search

ENG
IN

09:40
17-06-2024