ChatGPT - Code Copilot | Online Python Compiler (Interp

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

All Bookmarks

Programiz
Python Online Compiler

Programiz PRO >

main.py | Share | Run

```python
from itertools import combinations

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

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

def knapsack(weights, values, capacity):
    n = len(weights)
    max_value = 0
    best_selection = []

    for r in range(1, n + 1):
        for subset in combinations(range(n), r):
            if is_feasible(subset, weights, capacity):
                value = total_value(subset, values)
                if value > max_value:
                    max_value = value
                    best_selection = subset

    return best_selection, max_value

weights = [2, 3, 1]
values = [4, 5, 3]
capacity = 4

result = knapsack(weights, values, capacity)
print("Optimal Selection:", result[0])
print("Total Value:", result[1])
```

Output | Clear

```
Optimal Selection: (1, 2)
Total Value: 8

--- Code Execution Successful ---
```


27°C Haze | Search | ENG IN | 11:41 20-02-2025

ChatGPT - Code Copilot | Online Python Compiler (Interp

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

All Bookmarks

Programiz
Python Online Compiler

Programiz PRO ›

main.py

Share | Run

```python
from itertools import permutations

def total_cost(assignment, cost_matrix):
    return sum(cost_matrix[i][assignment[i]] for i in range(len(assignment)))

def assignment_problem(cost_matrix):
    n = len(cost_matrix)
    min_cost = float('inf')
    best_assignment = None

    for perm in permutations(range(n)):
        cost = total_cost(perm, cost_matrix)
        if cost < min_cost:
            min_cost = cost
            best_assignment = perm

    return best_assignment, min_cost

cost_matrix = [
    [3, 10, 7],
    [8, 5, 12],
    [4, 6, 9]
]

result = assignment_problem(cost_matrix)
print("Optimal Assignment:", result[0])
print("Total Cost:", result[1])
```

Output | Clear

```
Optimal Assignment: (2, 1, 0)
Total Cost: 16

--- Code Execution Successful ---
```

27°C Haze | Search | ENG IN | 11:41 20-02-2025

ChatGPT - Code Copilot | Online Python Compiler (Interp

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

All Bookmarks

Programiz
Python Online Compiler

Programiz PRO >

main.py

Share | Run

```python
from itertools import permutations
import math

def euclidean_distance(p1, p2):
    return math.sqrt((p1[0] - p2[0]) ** 2 + (p1[1] - p2[1]) ** 2)

def tsp(cities):
    min_distance = float('inf')
    best_path = None

    for perm in permutations(cities[1:]):
        path = [cities[0]] + list(perm) + [cities[0]]
        distance = sum(euclidean_distance(path[i], path[i+1]) for i in range(len(path) - 1))

        if distance < min_distance:
            min_distance = distance
            best_path = path

    return min_distance, best_path

cities = [(1, 2), (4, 5), (7, 1), (3, 6)]
result = tsp(cities)
print("Shortest Distance:", result[0])
print("Shortest Path:", result[1])
```

Output | Clear

```
Shortest Distance: 16.969112047670894
Shortest Path: [(1, 2), (7, 1), (4, 5), (3, 6), (1, 2)]

--- Code Execution Successful ---
```


27°C Haze

Search

ENG IN

11:41 20-02-2025

ChatGPT - Code Copilot | Online Python Compiler (Interp

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

All Bookmarks

Programiz
Python Online Compiler

Programiz PRO >

main.py

Share | Run

```python
def cross_product(o, a, b):
    return (a[0] - o[0]) * (b[1] - o[1]) - (a[1] - o[1]) * (b[0] - o[0])

def convex_hull_brute_force(points):
    n = len(points)
    if n < 3:
        return points

    hull = set()

    for i in range(n):
        for j in range(n):
            if i == j:
                continue
            valid = True
            for k in range(n):
                if k != i and k != j and cross_product(points[i], points[j], points[k]) > 0:
                    valid = False
                    break
            if valid:
                hull.add(points[i])
                hull.add(points[j])

    return sorted(hull)

points = [(1, 1), (4, 6), (8, 1), (0, 0), (3, 3)]
result = convex_hull_brute_force(points)
print("Convex Hull:", result)
```

Output | Clear

```
Convex Hull: [(0, 0), (4, 6), (8, 1)]

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


27°C Haze | Search | ENG IN | 11:41 20-02-2025

ChatGPT - Code Copilot | Online Python Compiler (Interp

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

All Bookmarks

Programiz
Python Online Compiler

Programiz PRO ›

main.py

Share | Run

```python
import math

def euclidean_distance(p1, p2):
    return math.sqrt((p1[0] - p2[0]) ** 2 + (p1[1] - p2[1]) ** 2)

def closest_pair_brute_force(points):
    min_dist = float('inf')
    closest_points = None
    n = len(points)

    for i in range(n):
        for j in range(i + 1, n):
            dist = euclidean_distance(points[i], points[j])
            if dist < min_dist:
                min_dist = dist
                closest_points = (points[i], points[j])

    return closest_points, min_dist

points = [(1, 2), (4, 5), (7, 8), (3, 1)]
result = closest_pair_brute_force(points)
print("Closest pair:", result[0])
print("Minimum distance:", result[1])
```

Output | Clear

```
Closest pair: ((1, 2), (3, 1))
Minimum distance: 2.23606797749979

--- Code Execution Successful ---
```


27°C Haze | Search | ENG IN | 11:40 20-02-2025