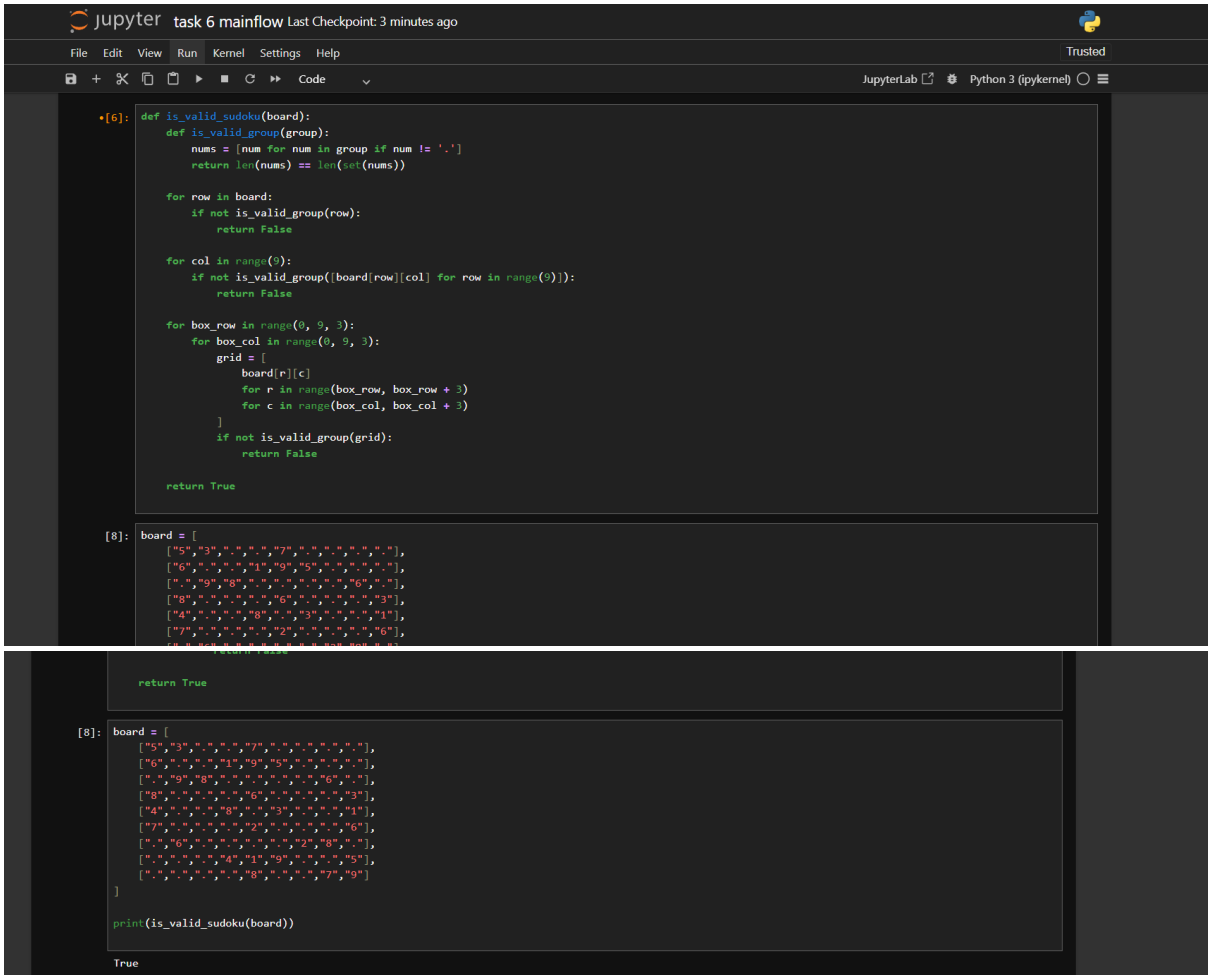
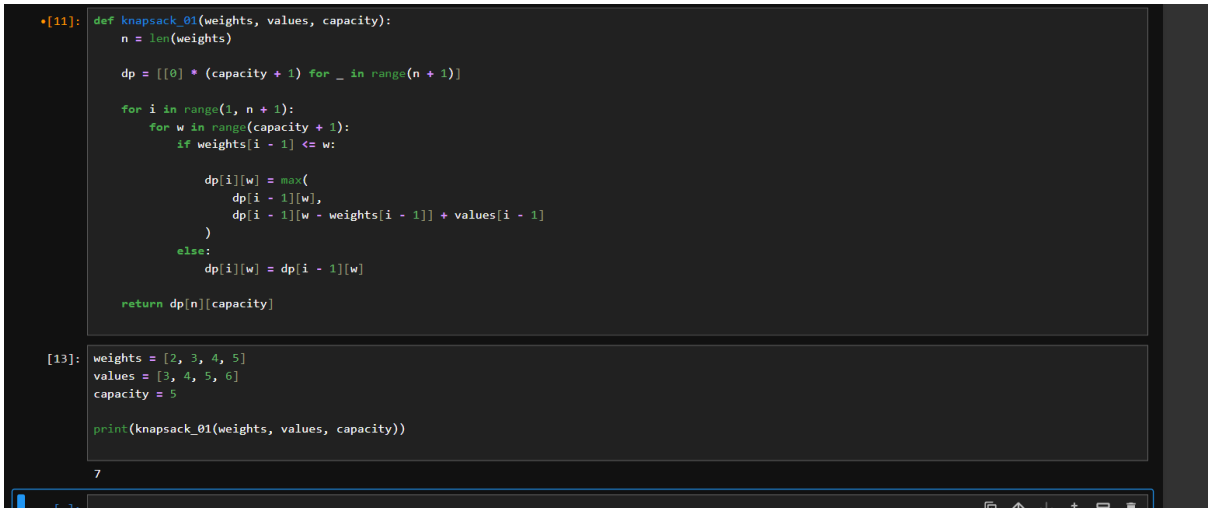
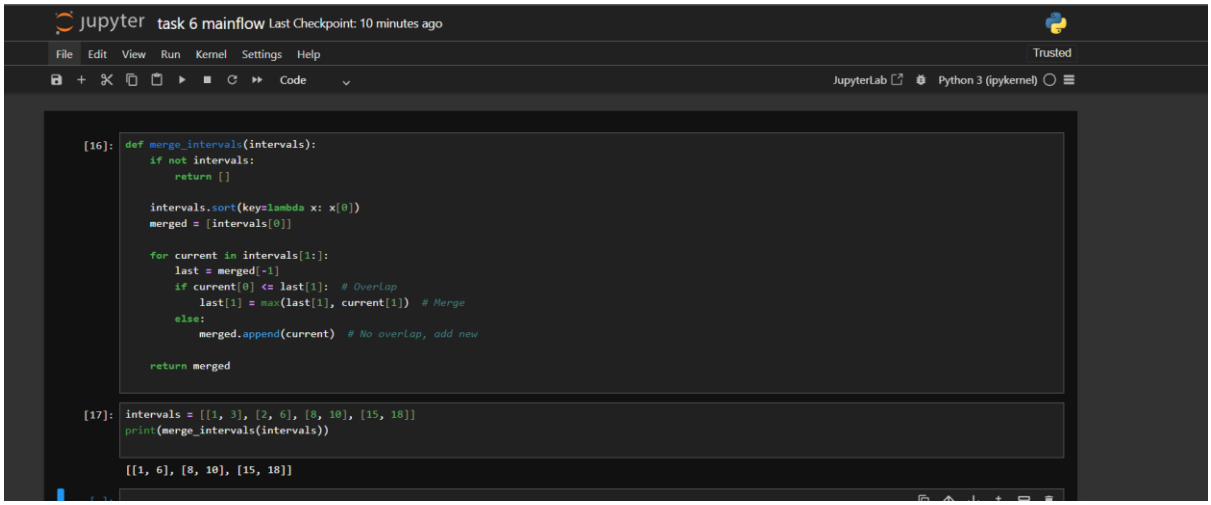
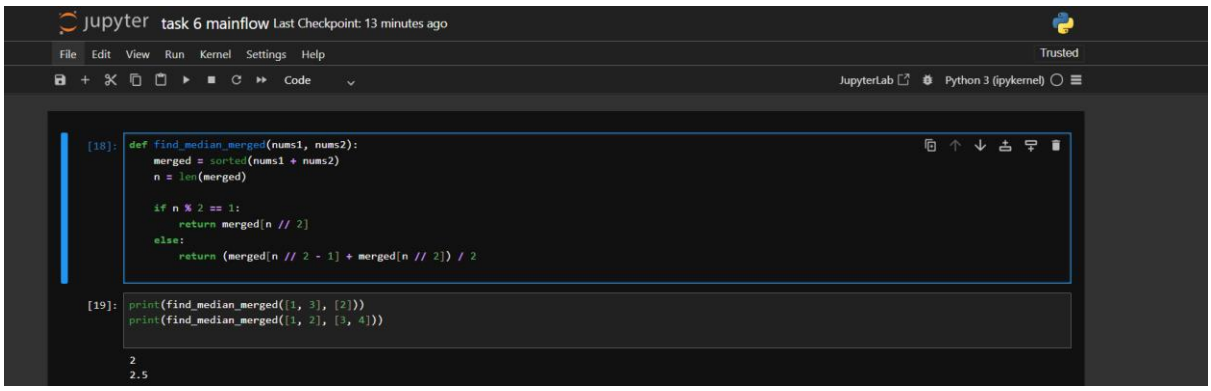

jupyter task 6 mainflow Last Checkpoint: 3 minutes ago
File Edit View Run Kernel Settings Help
Trusted
Code
JupyterLab
Python 3 (ipykernel)
[6]:
def is_valid_sudoku(board):
def is_valid_group(group):
nums = [num for num in group if num != '.']
return len(nums) == len(set(nums))
for row in board:
if not is_valid_group(row):
return False
for col in range(9):
if not is_valid_group([board[row][col] for row in range(9)]):
return False
for box_row in range(0, 9, 3):
for box_col in range(0, 9, 3):
grid = [
board[r][c]
for r in range(box_row, box_row + 3)
for c in range(box_col, box_col + 3)
]
if not is_valid_group(grid):
return False
return True
[8]:
board = [

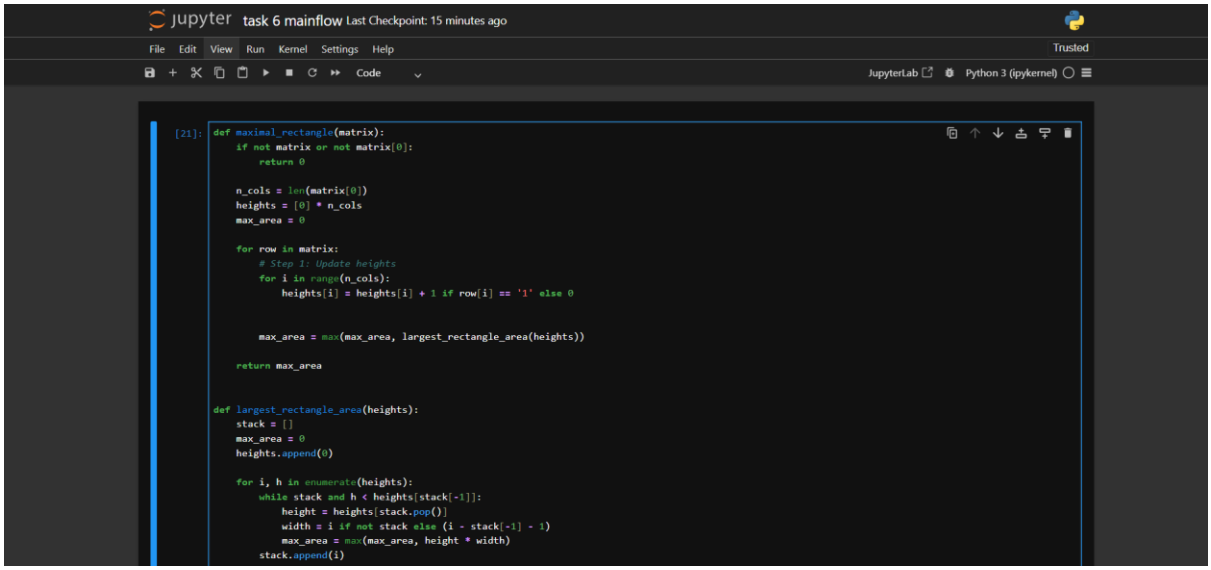
return True
[8]:
board = [
print(is_valid_sudoku(board))
True

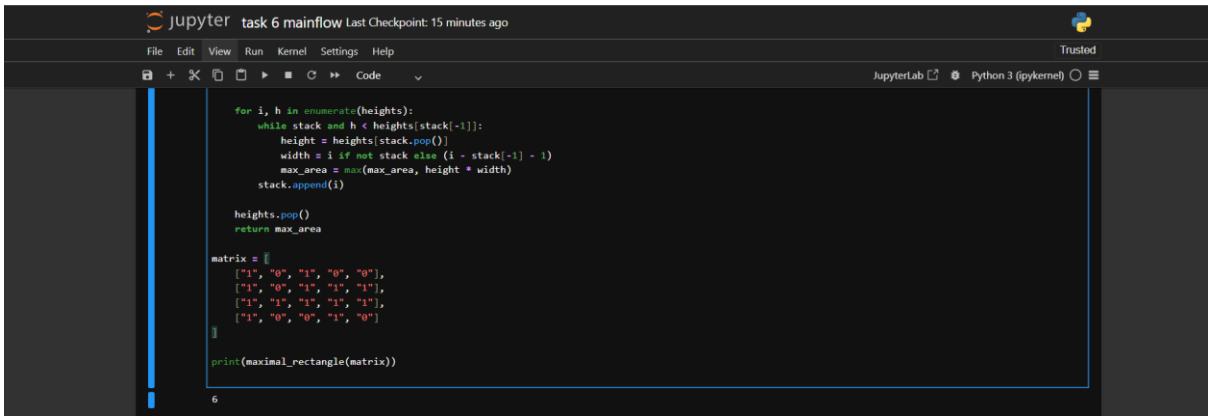

jupyter task 6 mainflow Last Checkpoint: 7 minutes ago
File Edit View Run Kernel Settings Help
Trusted
Code
JupyterLab Python 3 (ipykernel)
print(is_valid_sudoku(board))
True
[9]:
def word_frequency(text):
words = text.lower().split() # convert to lowercase and split by spaces
freq = {}
for word in words:
word = word.strip(".,!?;:'\"()[]{}") # remove common punctuation
if word: # skip empty strings
freq[word] = freq.get(word, 0) + 1
return freq
[10]:
text = "Hola, aunty! hows it going? going good? hola uncle!"
print(word_frequency(text))
{'hola': 2, 'aunty': 1, 'hows': 1, 'it': 1, 'going': 2, 'good': 1, 'uncle': 1}

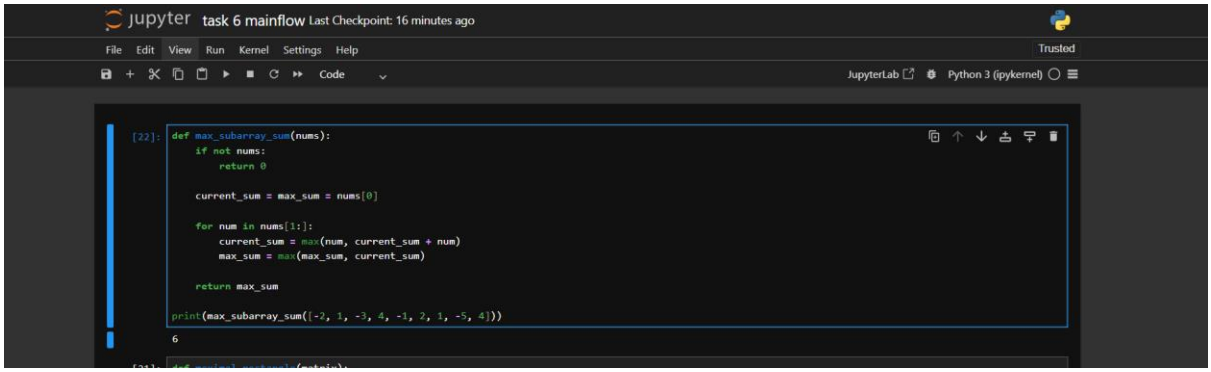

[11]:
def knapsack_01(weights, values, capacity):
n = len(weights)
dp = [[0] * (capacity + 1) for _ in range(n + 1)]
for i in range(1, n + 1):
for w in range(capacity + 1):
if weights[i - 1] <= w:
dp[i][w] = max(
dp[i - 1][w],
dp[i - 1][w - weights[i - 1]] + values[i - 1]
)
else:
dp[i][w] = dp[i - 1][w]
return dp[n][capacity]
[13]:
weights = [2, 3, 4, 5]
values = [3, 4, 5, 6]
capacity = 5
print(knapsack_01(weights, values, capacity))
7

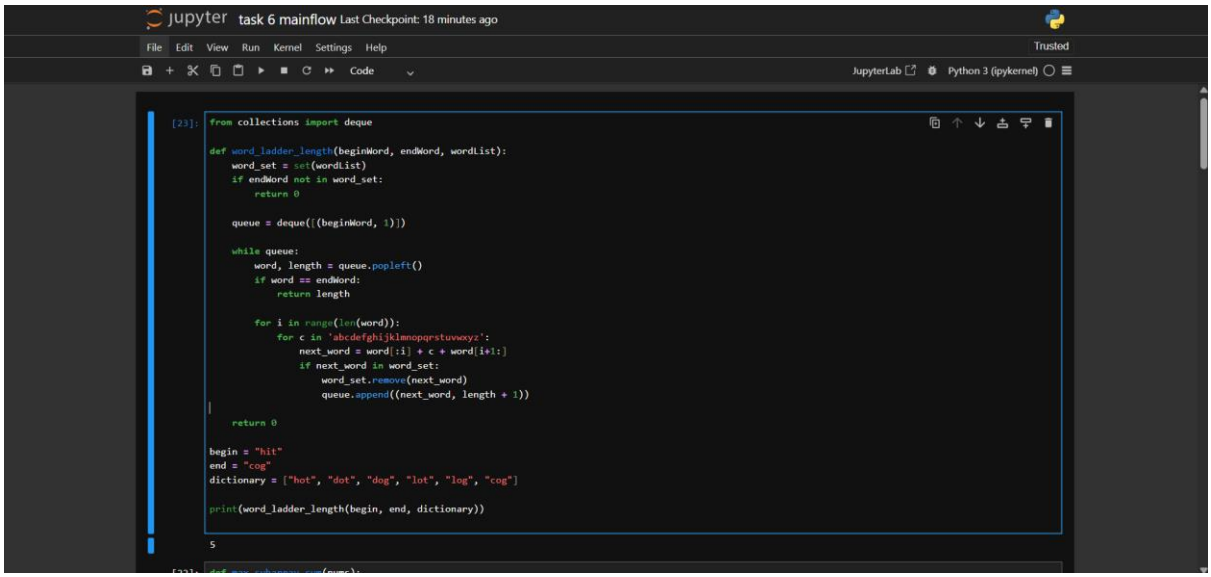

jupyter task 6 mainflow Last Checkpoint: 10 minutes ago
File Edit View Run Kernel Settings Help
Trusted
Code
JupyterLab Python 3 (ipykernel)
[16]:
def merge_intervals(intervals):
if not intervals:
return []
intervals.sort(key=lambda x: x[0])
merged = [intervals[0]]
for current in intervals[1:]:
last = merged[-1]
if current[0] <= last[1]: # Overlap
last[1] = max(last[1], current[1]) # Merge
else:
merged.append(current) # No overlap, add new
return merged
[17]:
intervals = [[1, 3], [2, 6], [8, 10], [15, 18]]
print(merge_intervals(intervals))
[[1, 6], [8, 10], [15, 18]]

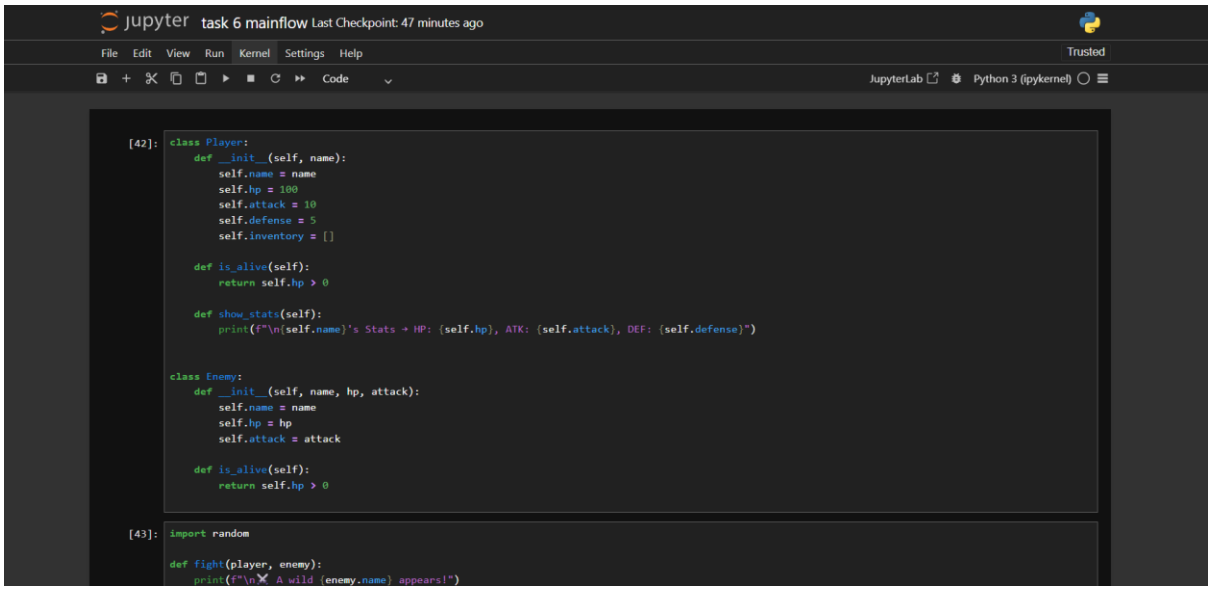

jupyter task 6 mainflow Last Checkpoint: 13 minutes ago
File Edit View Run Kernel Settings Help
Trusted
Code
JupyterLab Python 3 (ipykernel)
[18]:
def find_median_merged(nums1, nums2):
merged = sorted(nums1 + nums2)
n = len(merged)
if n % 2 == 1:
return merged[n // 2]
else:
return (merged[n // 2 - 1] + merged[n // 2]) / 2
[19]:
print(find_median_merged([1, 3], [2]))
print(find_median_merged([1, 2], [3, 4]))
2
2.5

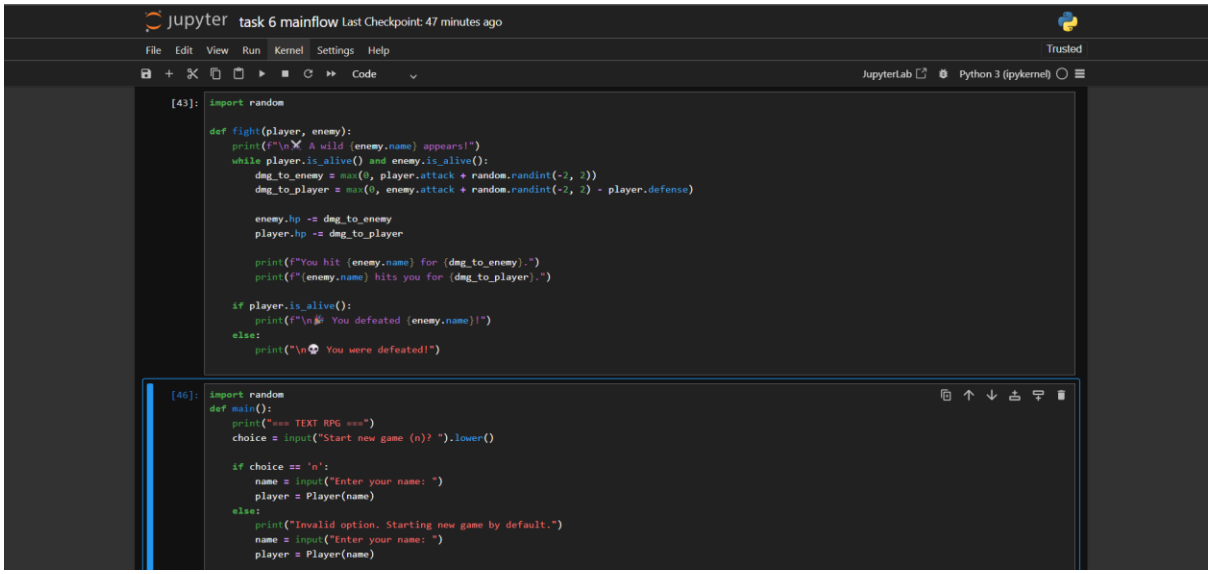

jupyter task 6 mainflow Last Checkpoint: 15 minutes ago
File Edit View Run Kernel Settings Help
Trusted
Code
JupyterLab Python 3 (ipykernel)
[21]:
def maximal_rectangle(matrix):
if not matrix or not matrix[0]:
return 0
n_cols = len(matrix[0])
heights = [0] * n_cols
max_area = 0
for row in matrix:
# Step 1: Update heights
for i in range(n_cols):
heights[i] = heights[i] + 1 if row[i] == '1' else 0
max_area = max(max_area, largest_rectangle_area(heights))
return max_area
def largest_rectangle_area(heights):
stack = []
max_area = 0
heights.append(0)
for i, h in enumerate(heights):
while stack and h < heights[stack[-1]]:
height = heights[stack.pop()]
width = i if not stack else (i - stack[-1] - 1)
max_area = max(max_area, height * width)
stack.append(i)

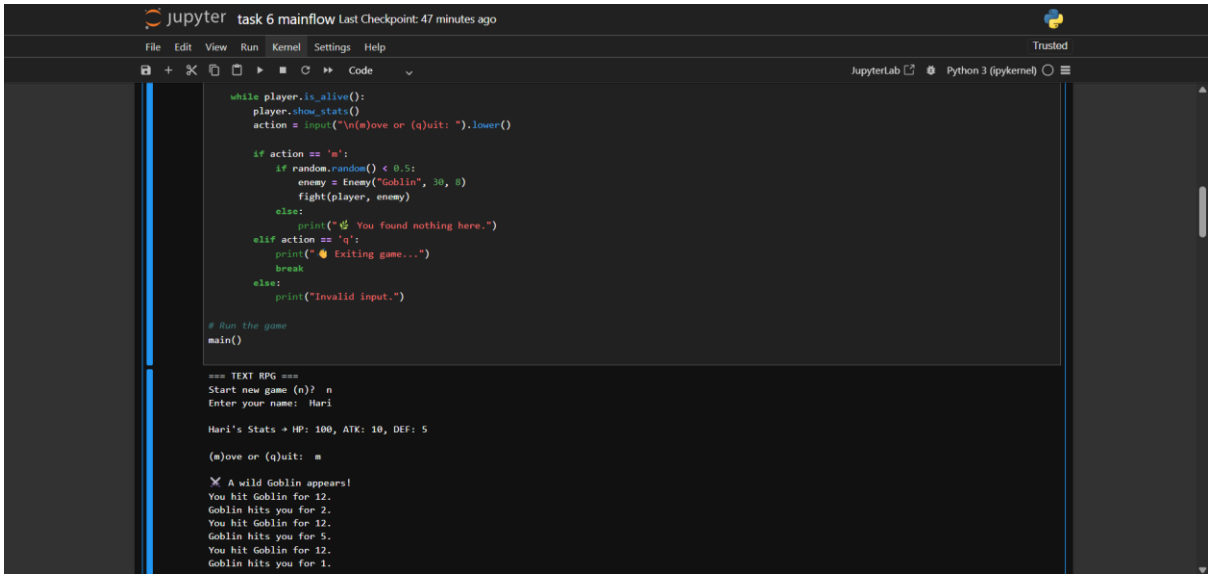

jupyter task 6 mainflow Last Checkpoint: 15 minutes ago
File Edit View Run Kernel Settings Help
Trusted
Code
JupyterLab Python 3 (ipykernel)
for i, h in enumerate(heights):
while stack and h < heights[stack[-1]]:
height = heights[stack.pop()]
width = i if not stack else (i - stack[-1] - 1)
max_area = max(max_area, height * width)
stack.append(i)
heights.pop()
return max_area
matrix = [
["1", "0", "1", "0", "0"],
["1", "0", "1", "1", "1"],
["1", "1", "1", "1", "1"],
["1", "0", "0", "1", "0"]
]
print(maximal_rectangle(matrix))
6

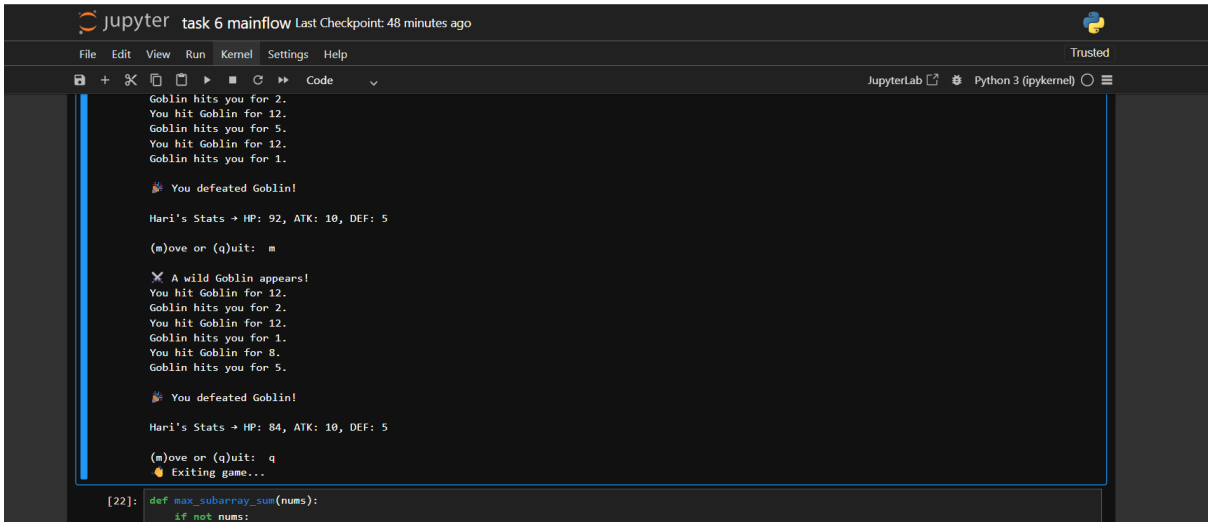

jupyter task 6 mainflow Last Checkpoint: 16 minutes ago
File Edit View Run Kernel Settings Help
Trusted
Code
JupyterLab Python 3 (ipykernel)
[22]:
def max_subarray_sum(nums):
if not nums:
return 0
current_sum = max_sum = nums[0]
for num in nums[1:]:
current_sum = max(num, current_sum + num)
max_sum = max(max_sum, current_sum)
return max_sum
print(max_subarray_sum([-2, 1, -3, 4, -1, 2, 1, -5, 4]))
6


jupyter task 6 mainflow Last Checkpoint: 18 minutes ago
File Edit View Run Kernel Settings Help
Trusted
Code
JupyterLab Python 3 (ipykernel)
[23]:
from collections import deque
def word_ladder_length(beginWord, endWord, wordList):
word_set = set(wordList)
if endWord not in word_set:
return 0
queue = deque([(beginWord, 1)])
while queue:
word, length = queue.popleft()
if word == endWord:
return length
for i in range(len(word)):
for c in 'abcdefghijklmnopqrstuvwxyz':
next_word = word[:i] + c + word[i+1:]
if next_word in word_set:
word_set.remove(next_word)
queue.append((next_word, length + 1))
return 0
begin = "hit"
end = "cog"
dictionary = ["hot", "dot", "dog", "lot", "log", "cog"]
print(word_ladder_length(begin, end, dictionary))
5


jupyter task 6 mainflow Last Checkpoint: 47 minutes ago
File Edit View Run Kernel Settings Help
Trusted
Code
JupyterLab Python 3 (ipykernel)
[42]:
class Player:
def __init__(self, name):
self.name = name
self.hp = 100
self.attack = 10
self.defense = 5
self.inventory = []
def is_alive(self):
return self.hp > 0
def show_stats(self):
print(f"\n{self.name}'s Stats → HP: {self.hp}, ATK: {self.attack}, DEF: {self.defense}")
class Enemy:
def __init__(self, name, hp, attack):
self.name = name
self.hp = hp
self.attack = attack
def is_alive(self):
return self.hp > 0
[43]:
import random
def fight(player, enemy):
print(f"\n⚔ A wild {enemy.name} appears!")


jupyter task 6 mainflow Last Checkpoint: 47 minutes ago
File Edit View Run Kernel Settings Help
Trusted
Code
JupyterLab
Python 3 (ipykernel)
[43]:
import random
def fight(player, enemy):
print(f"\n⚔️ A wild {enemy.name} appears!")
while player.is_alive() and enemy.is_alive():
dmg_to_enemy = max(0, player.attack + random.randint(-2, 2))
dmg_to_player = max(0, enemy.attack + random.randint(-2, 2) - player.defense)
enemy.hp -= dmg_to_enemy
player.hp -= dmg_to_player
print(f"You hit {enemy.name} for {dmg_to_enemy}.")
print(f"{enemy.name} hits you for {dmg_to_player}.")
if player.is_alive():
print(f"\n🎉 You defeated {enemy.name}!")
else:
print("\n💀 You were defeated!")
[46]:
import random
def main():
print("=== TEXT RPG ===")
choice = input("Start new game (n)? ").lower()
if choice == 'n':
name = input("Enter your name: ")
player = Player(name)
else:
print("Invalid option. Starting new game by default.")
name = input("Enter your name: ")
player = Player(name)


jupyter task 6 mainflow Last Checkpoint: 47 minutes ago
File Edit View Run Kernel Settings Help
Trusted
Code
JupyterLab
Python 3 (ipykernel)
while player.is_alive():
player.show_stats()
action = input("\n(m)ove or (q)uit: ").lower()
if action == 'm':
if random.random() < 0.5:
enemy = Enemy("Goblin", 30, 8)
fight(player, enemy)
else:
print("🌿 You found nothing here.")
elif action == 'q':
print("👋 Exiting game...")
break
else:
print("Invalid input.")
# Run the game
main()
=== TEXT RPG ===
Start new game (n)? n
Enter your name: Hari
Hari's Stats → HP: 100, ATK: 10, DEF: 5
(m)ove or (q)uit: m
⚔️ A wild Goblin appears!
You hit Goblin for 12.
Goblin hits you for 2.
You hit Goblin for 12.
Goblin hits you for 5.
You hit Goblin for 12.
Goblin hits you for 1.


jupyter task 6 mainflow Last Checkpoint: 48 minutes ago
File Edit View Run Kernel Settings Help
Trusted
Code
JupyterLab
Python 3 (ipykernel)
Goblin hits you for 2.
You hit Goblin for 12.
Goblin hits you for 5.
You hit Goblin for 12.
Goblin hits you for 1.
🎉 You defeated Goblin!
Hari's Stats → HP: 92, ATK: 10, DEF: 5
(m)ove or (q)uit: m
⚔️ A wild Goblin appears!
You hit Goblin for 12.
Goblin hits you for 2.
You hit Goblin for 12.
Goblin hits you for 1.
You hit Goblin for 8.
Goblin hits you for 5.
🎉 You defeated Goblin!
Hari's Stats → HP: 84, ATK: 10, DEF: 5
(m)ove or (q)uit: q
👋 Exiting game...
[22]:
def max_subarray_sum(nums):
if not nums: