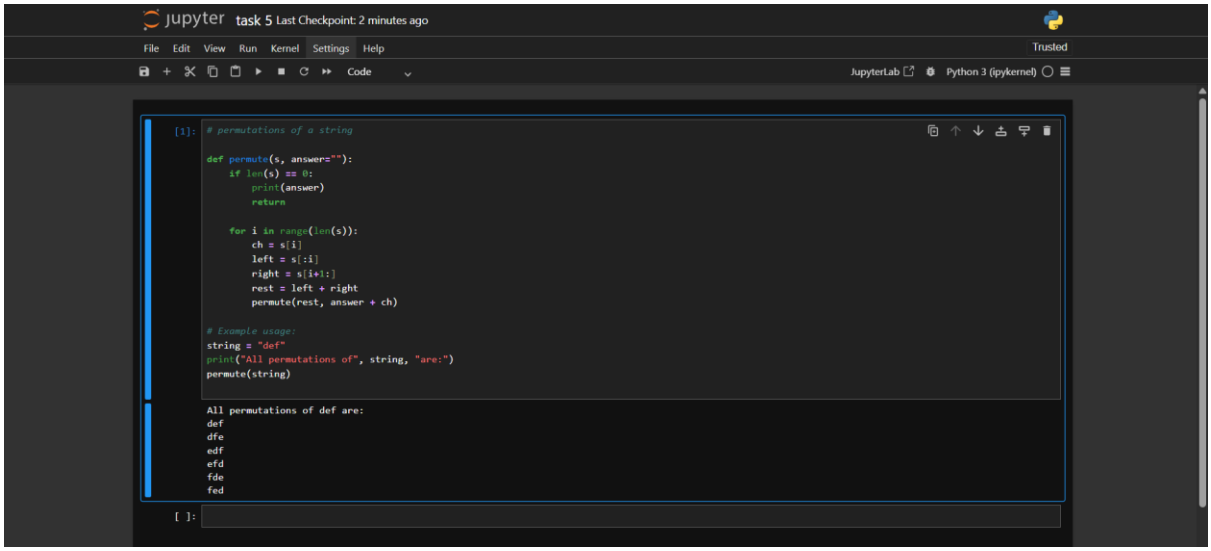
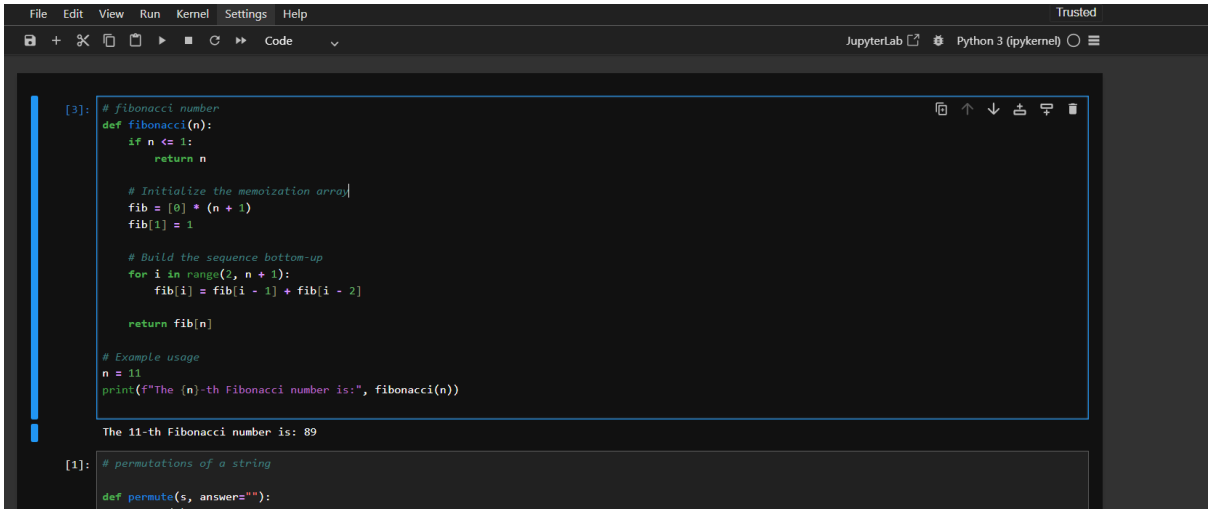
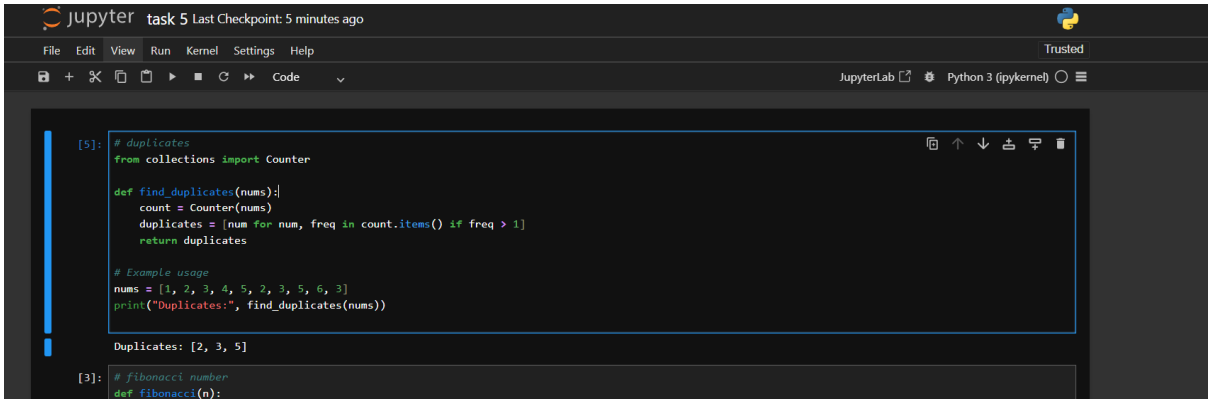
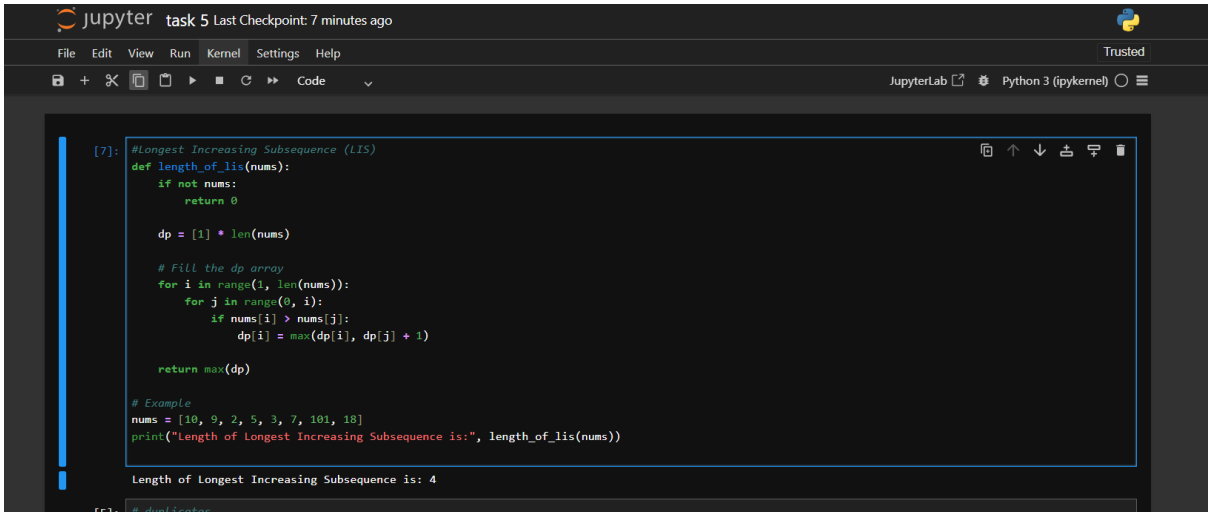
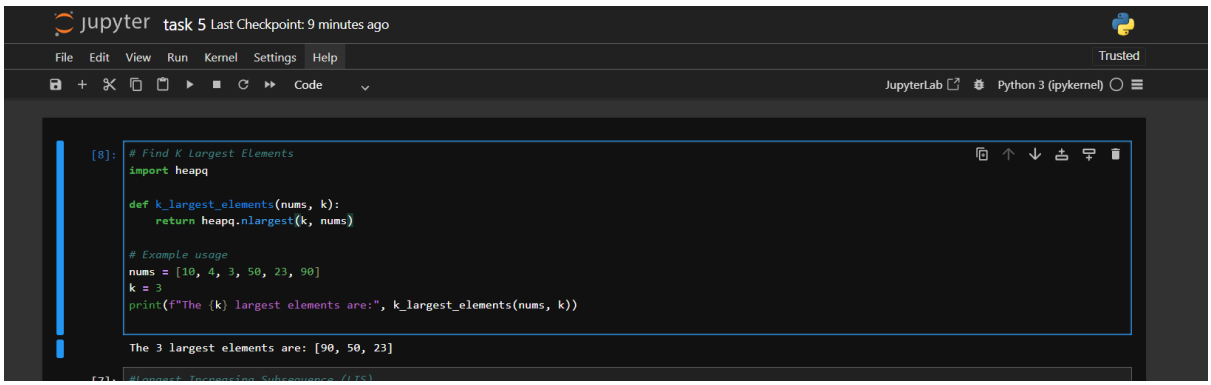
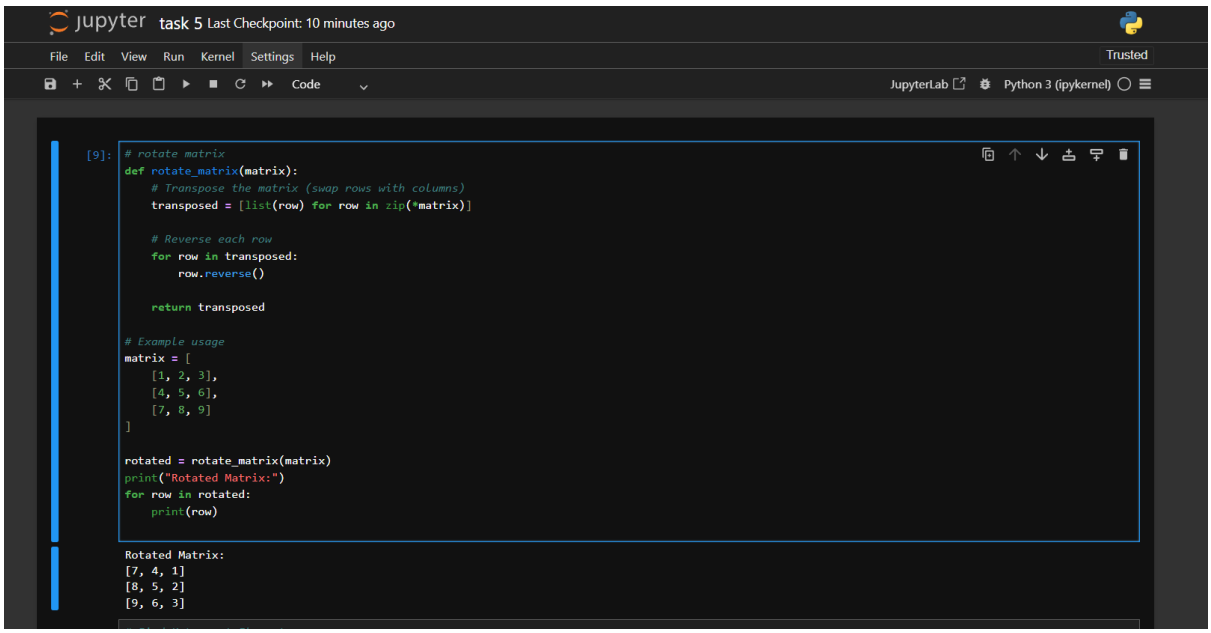

jupyter task 5 Last Checkpoint: 2 minutes ago
File Edit View Run Kernel Settings Help
Trusted
Code
JupyterLab
Python 3 (ipykernel)
[1]:
# permutations of a string
def permute(s, answer=""):
if len(s) == 0:
print(answer)
return
for i in range(len(s)):
ch = s[i]
left = s[:i]
right = s[i+1:]
rest = left + right
permute(rest, answer + ch)
# Example usage:
string = "def"
print("All permutations of", string, "are:")
permute(string)
All permutations of def are:
def
dfe
edf
efd
fde
fed
[ ]:

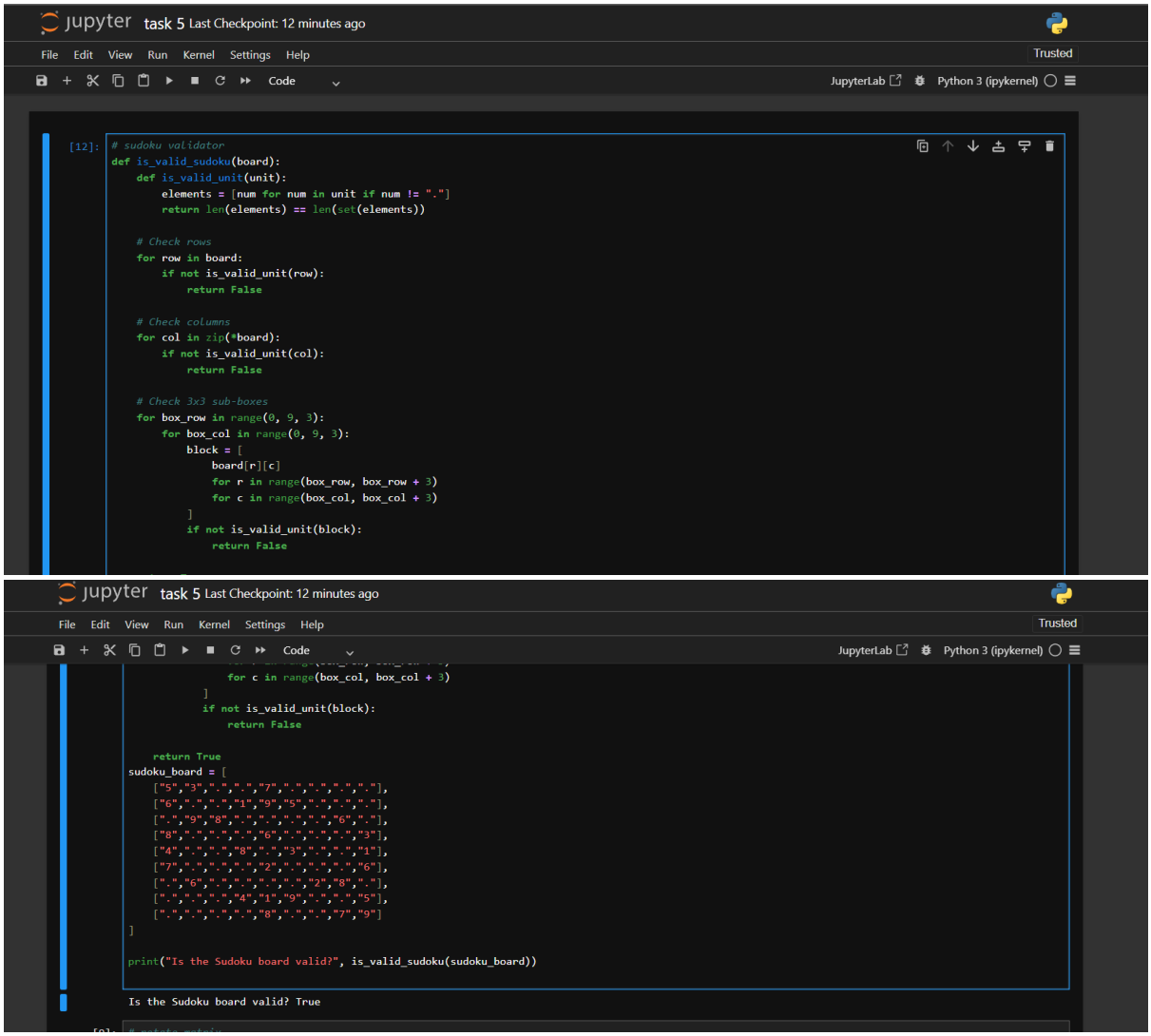

File Edit View Run Kernel Settings Help
Trusted
Code
JupyterLab
Python 3 (ipykernel)
[3]:
# fibonacci number
def fibonacci(n):
if n <= 1:
return n
# Initialize the memoization array
fib = [0] * (n + 1)
fib[1] = 1
# Build the sequence bottom-up
for i in range(2, n + 1):
fib[i] = fib[i - 1] + fib[i - 2]
return fib[n]
# Example usage
n = 11
print(f"The {n}-th Fibonacci number is:", fibonacci(n))
The 11-th Fibonacci number is: 89
[1]:
# permutations of a string
def permute(s, answer=""):


jupyter task 5 Last Checkpoint: 5 minutes ago
File Edit View Run Kernel Settings Help
Trusted
Code
JupyterLab
Python 3 (ipykernel)
[5]:
# duplicates
from collections import Counter
def find_duplicates(nums):
count = Counter(nums)
duplicates = [num for num, freq in count.items() if freq > 1]
return duplicates
# Example usage
nums = [1, 2, 3, 4, 5, 2, 3, 5, 6, 3]
print("Duplicates:", find_duplicates(nums))
Duplicates: [2, 3, 5]
[3]:
# fibonacci number
def fibonacci(n):

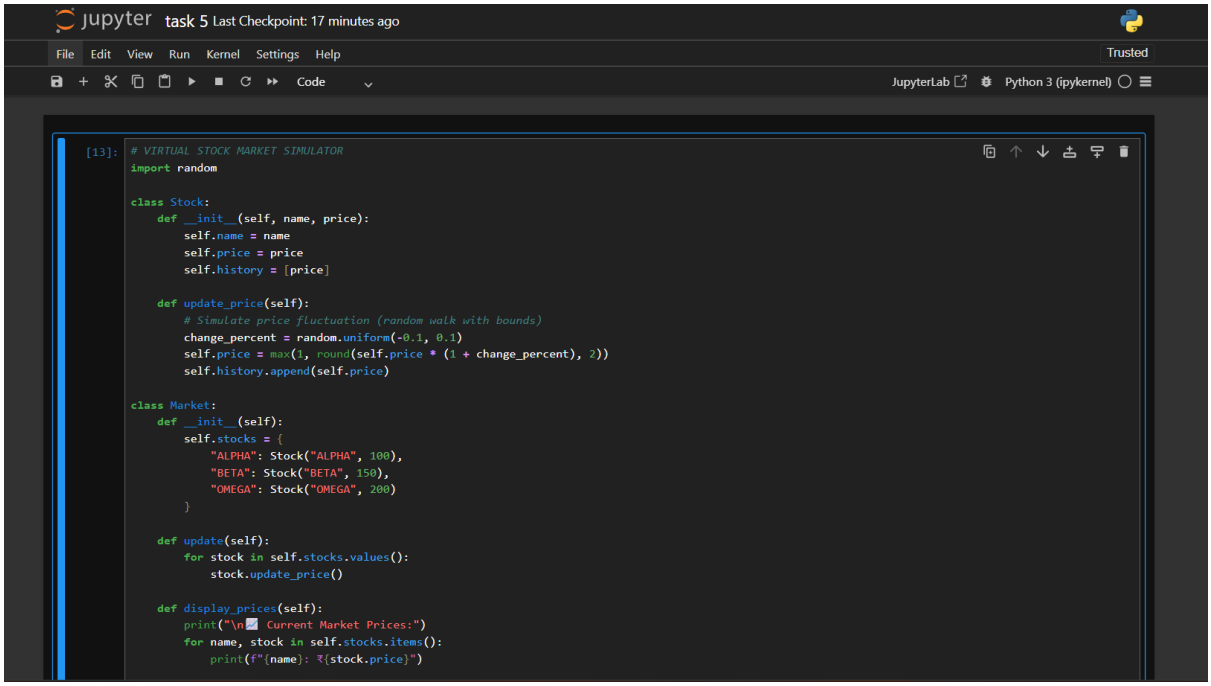

jupyter task 5 Last Checkpoint: 7 minutes ago
File Edit View Run Kernel Settings Help
Trusted
Code
JupyterLab Python 3 (ipykernel)
[7]:
#Longest Increasing Subsequence (LIS)
def length_of_lis(nums):
if not nums:
return 0
dp = [1] * len(nums)
# Fill the dp array
for i in range(1, len(nums)):
for j in range(0, i):
if nums[i] > nums[j]:
dp[i] = max(dp[i], dp[j] + 1)
return max(dp)
# Example
nums = [10, 9, 2, 5, 3, 7, 101, 18]
print("Length of Longest Increasing Subsequence is:", length_of_lis(nums))
Length of Longest Increasing Subsequence is: 4

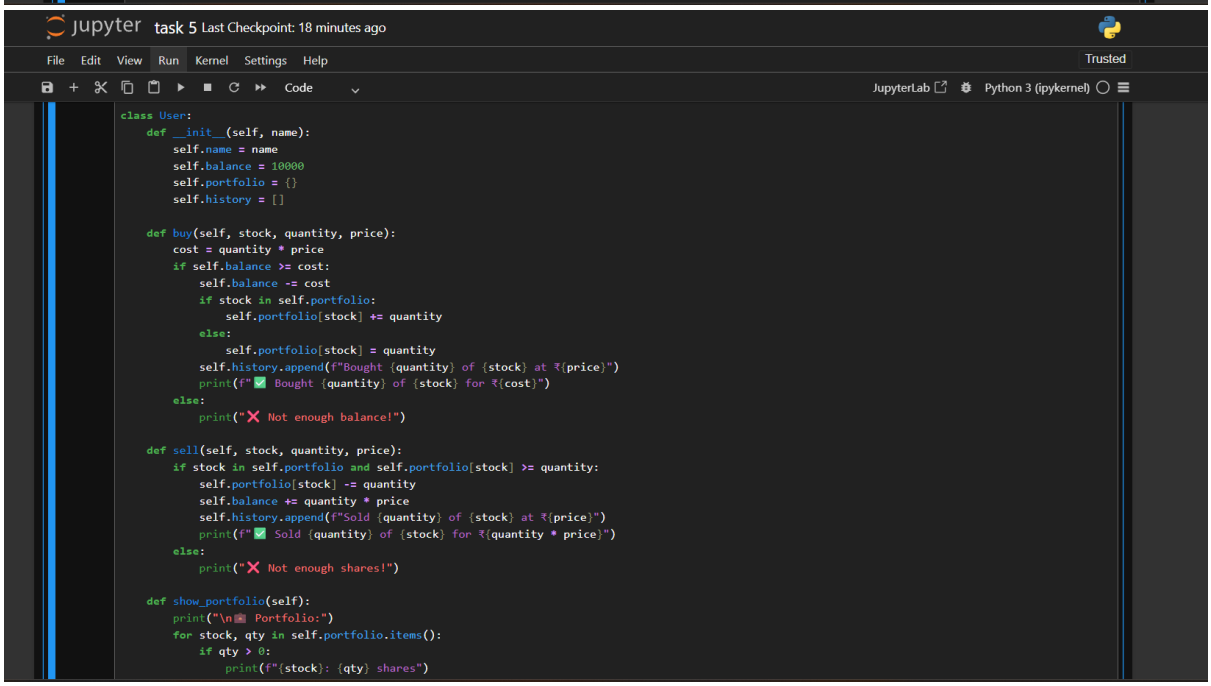

jupyter task 5 Last Checkpoint: 9 minutes ago
File Edit View Run Kernel Settings Help
Trusted
Code
JupyterLab Python 3 (ipykernel)
[8]:
# Find K Largest Elements
import heapq
def k_largest_elements(nums, k):
return heapq.nlargest(k, nums)
# Example usage
nums = [10, 4, 3, 50, 23, 90]
k = 3
print(f"The {k} largest elements are:", k_largest_elements(nums, k))
The 3 largest elements are: [90, 50, 23]

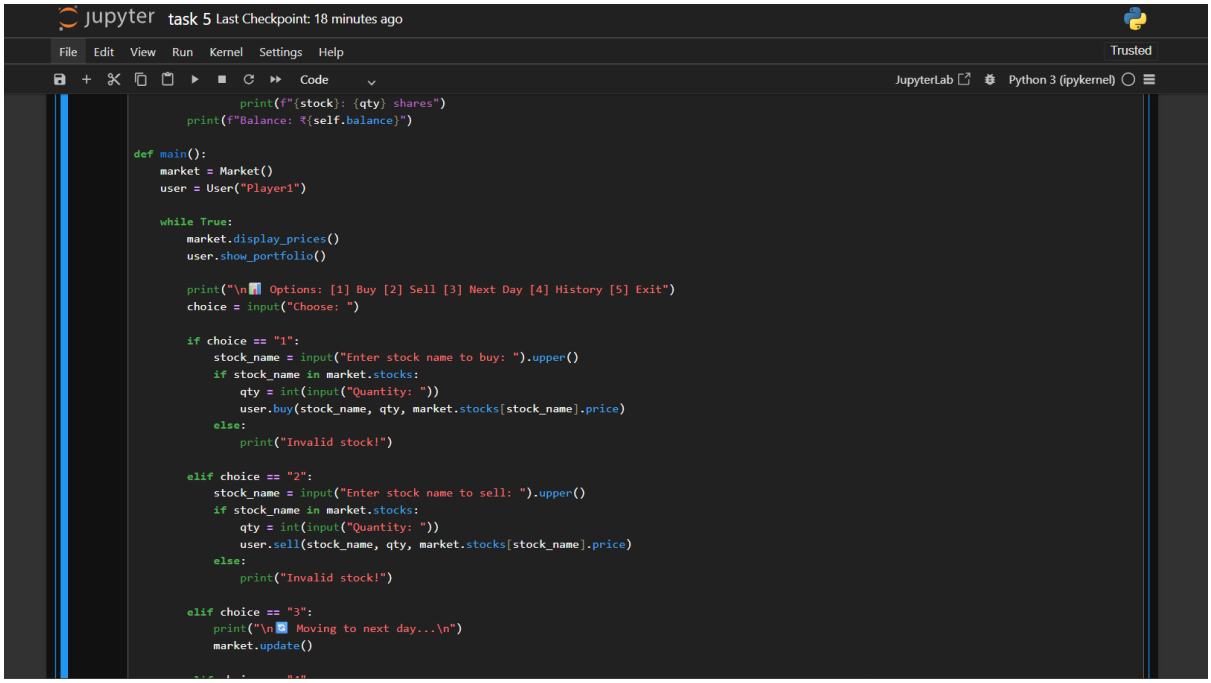

jupyter task 5 Last Checkpoint: 10 minutes ago
File Edit View Run Kernel Settings Help
Trusted
Code
JupyterLab Python 3 (ipykernel)
[9]:
# rotate matrix
def rotate_matrix(matrix):
# Transpose the matrix (swap rows with columns)
transposed = [list(row) for row in zip(*matrix)]
# Reverse each row
for row in transposed:
row.reverse()
return transposed
# Example usage
matrix = [
[1, 2, 3],
[4, 5, 6],
[7, 8, 9]
]
rotated = rotate_matrix(matrix)
print("Rotated Matrix:")
for row in rotated:
print(row)
Rotated Matrix:
[7, 4, 1]
[8, 5, 2]
[9, 6, 3]

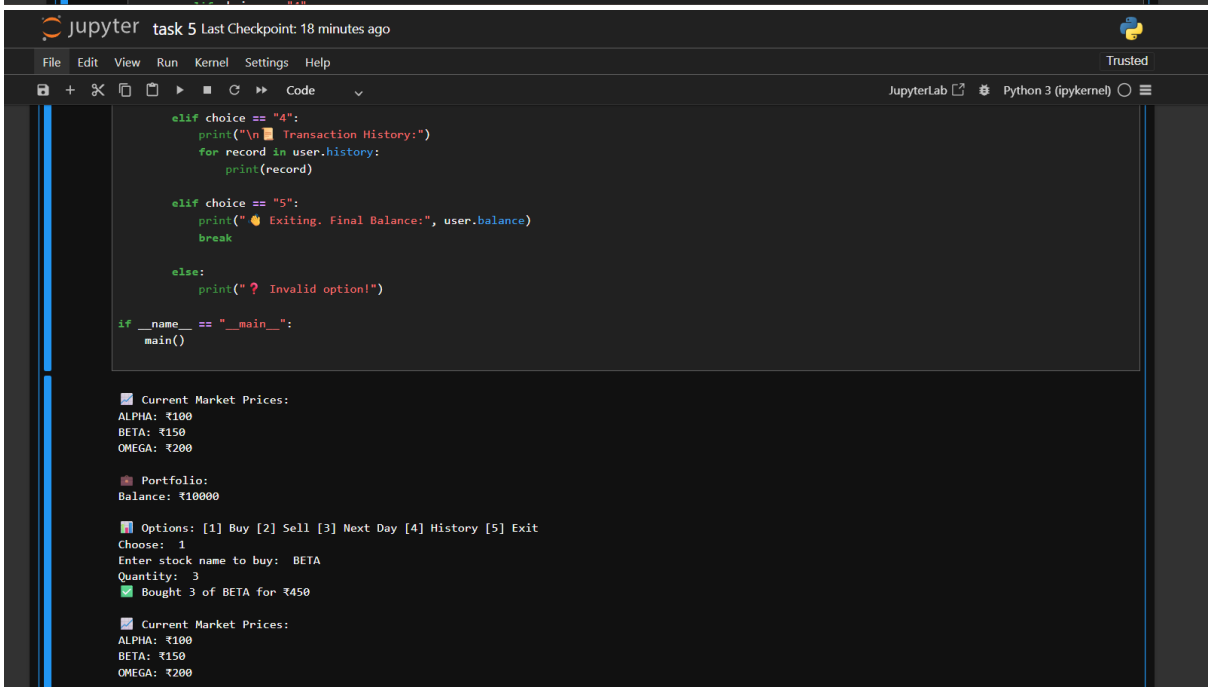

jupyter task 5 Last Checkpoint: 12 minutes ago
File Edit View Run Kernel Settings Help
Trusted
Code
JupyterLab
Python 3 (ipykernel)
[12]:
# sudoku validator
def is_valid_sudoku(board):
def is_valid_unit(unit):
elements = [num for num in unit if num != "."]
return len(elements) == len(set(elements))
# Check rows
for row in board:
if not is_valid_unit(row):
return False
# Check columns
for col in zip(*board):
if not is_valid_unit(col):
return False
# Check 3x3 sub-boxes
for box_row in range(0, 9, 3):
for box_col in range(0, 9, 3):
block = [
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_unit(block):
return False

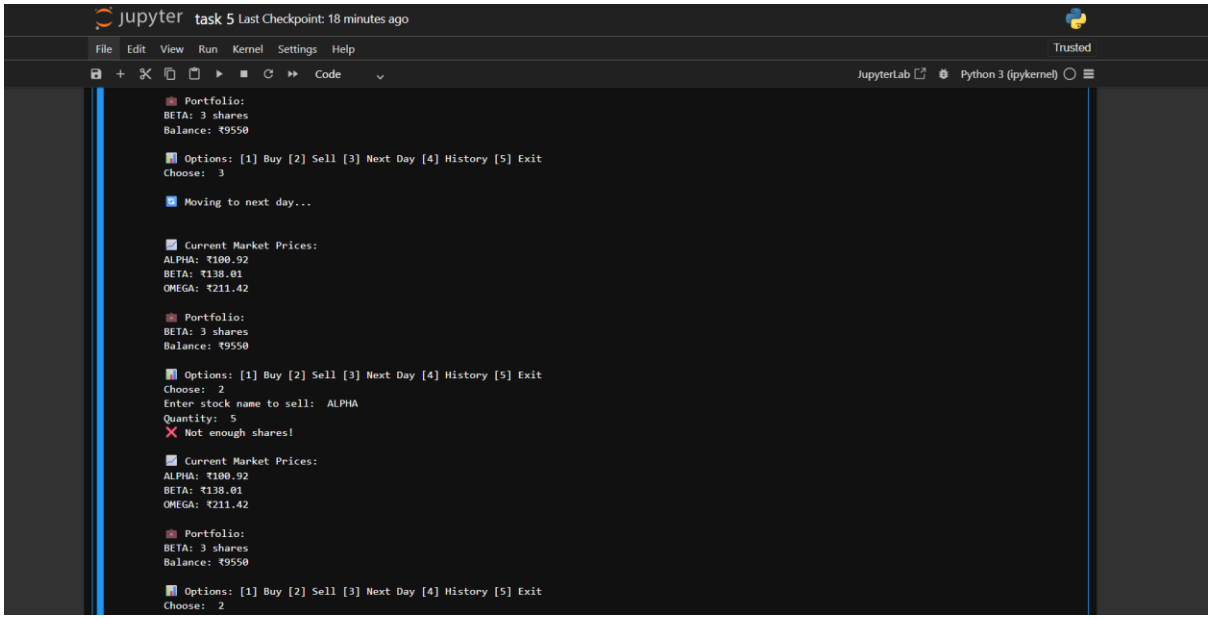
jupyter task 5 Last Checkpoint: 12 minutes ago
File Edit View Run Kernel Settings Help
Trusted
Code
JupyterLab
Python 3 (ipykernel)
for c in range(box_col, box_col + 3)
]
if not is_valid_unit(block):
return False
return True
sudoku_board = [
["5","3",".",".","7",".",".",".","."],
["6",".",".","1","9","5",".",".","."],
[".","9","8",".",".",".",".","6","."],
["8",".",".",".","6",".",".",".","3"],
["4",".",".","8",".","3",".",".","1"],
["7",".",".",".","2",".",".",".","6"],
[".","6",".",".",".",".","2","8","."],
[".",".",".","4","1","9",".",".","5"],
[".",".",".",".","8",".",".","7","9"]
]
print("Is the Sudoku board valid?", is_valid_sudoku(sudoku_board))
Is the Sudoku board valid? True

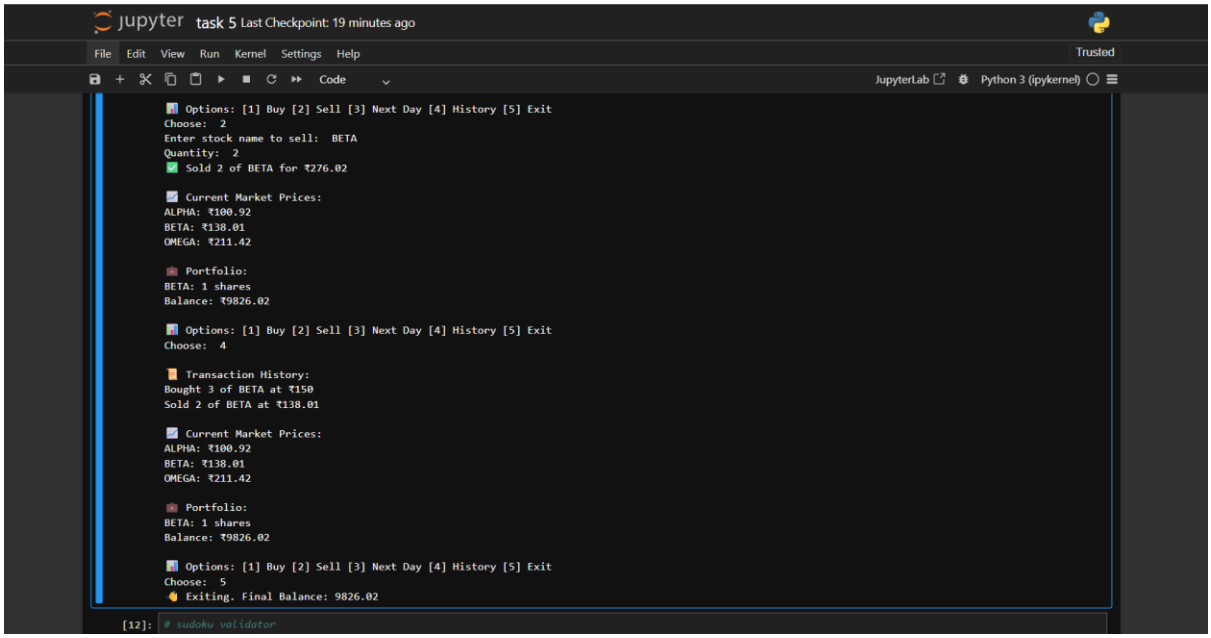

jupyter task 5 Last Checkpoint: 17 minutes ago
File Edit View Run Kernel Settings Help
Trusted
Code
JupyterLab
Python 3 (ipykernel)
[13]:
# VIRTUAL STOCK MARKET SIMULATOR
import random
class Stock:
def __init__(self, name, price):
self.name = name
self.price = price
self.history = [price]
def update_price(self):
# Simulate price fluctuation (random walk with bounds)
change_percent = random.uniform(-0.1, 0.1)
self.price = max(1, round(self.price * (1 + change_percent), 2))
self.history.append(self.price)
class Market:
def __init__(self):
self.stocks = {
"ALPHA": Stock("ALPHA", 100),
"BETA": Stock("BETA", 150),
"OMEGA": Stock("OMEGA", 200)
}
def update(self):
for stock in self.stocks.values():
stock.update_price()
def display_prices(self):
print("\n📈 Current Market Prices:")
for name, stock in self.stocks.items():
print(f"{name}: ₹{stock.price}")


jupyter task 5 Last Checkpoint: 18 minutes ago
File Edit View Run Kernel Settings Help
Trusted
Code
JupyterLab
Python 3 (ipykernel)
class User:
def __init__(self, name):
self.name = name
self.balance = 10000
self.portfolio = {}
self.history = []
def buy(self, stock, quantity, price):
cost = quantity * price
if self.balance >= cost:
self.balance -= cost
if stock in self.portfolio:
self.portfolio[stock] += quantity
else:
self.portfolio[stock] = quantity
self.history.append(f"Bought {quantity} of {stock} at ₹{price}")
print(f"✅ Bought {quantity} of {stock} for ₹{cost}")
else:
print("❌ Not enough balance!")
def sell(self, stock, quantity, price):
if stock in self.portfolio and self.portfolio[stock] >= quantity:
self.portfolio[stock] -= quantity
self.balance += quantity * price
self.history.append(f"Sold {quantity} of {stock} at ₹{price}")
print(f"✅ Sold {quantity} of {stock} for ₹{quantity * price}")
else:
print("❌ Not enough shares!")
def show_portfolio(self):
print("\n💼 Portfolio:")
for stock, qty in self.portfolio.items():
if qty > 0:
print(f"{stock}: {qty} shares")


jupyter task 5 Last Checkpoint: 18 minutes ago
File Edit View Run Kernel Settings Help
Trusted
Code
JupyterLab
Python 3 (ipykernel)
print(f"{stock}: {qty} shares")
print(f"Balance: ₹{self.balance}")
def main():
market = Market()
user = User("Player1")
while True:
market.display_prices()
user.show_portfolio()
print("\n📊 Options: [1] Buy [2] Sell [3] Next Day [4] History [5] Exit")
choice = input("Choose: ")
if choice == "1":
stock_name = input("Enter stock name to buy: ").upper()
if stock_name in market.stocks:
qty = int(input("Quantity: "))
user.buy(stock_name, qty, market.stocks[stock_name].price)
else:
print("Invalid stock!")
elif choice == "2":
stock_name = input("Enter stock name to sell: ").upper()
if stock_name in market.stocks:
qty = int(input("Quantity: "))
user.sell(stock_name, qty, market.stocks[stock_name].price)
else:
print("Invalid stock!")
elif choice == "3":
print("\n🔄 Moving to next day...\n")
market.update()


jupyter task 5 Last Checkpoint: 18 minutes ago
File Edit View Run Kernel Settings Help
Trusted
Code
JupyterLab
Python 3 (ipykernel)
elif choice == "4":
print("\n📜 Transaction History:")
for record in user.history:
print(record)
elif choice == "5":
print("👋 Exiting. Final Balance:", user.balance)
break
else:
print("❓ Invalid option!")
if __name__ == "__main__":
main()
📈 Current Market Prices:
ALPHA: ₹100
BETA: ₹150
OMEGA: ₹200
💼 Portfolio:
Balance: ₹10000
📊 Options: [1] Buy [2] Sell [3] Next Day [4] History [5] Exit
Choose: 1
Enter stock name to buy: BETA
Quantity: 3
✅ Bought 3 of BETA for ₹450
📈 Current Market Prices:
ALPHA: ₹100
BETA: ₹150
OMEGA: ₹200


jupyter task 5 Last Checkpoint: 18 minutes ago
File Edit View Run Kernel Settings Help
Trusted
Code
JupyterLab Python 3 (ipykernel)
💼 Portfolio:
BETA: 3 shares
Balance: ₹9550
📊 Options: [1] Buy [2] Sell [3] Next Day [4] History [5] Exit
Choose: 3
⏭️ Moving to next day...
📈 Current Market Prices:
ALPHA: ₹100.92
BETA: ₹138.01
OMEGA: ₹211.42
💼 Portfolio:
BETA: 3 shares
Balance: ₹9550
📊 Options: [1] Buy [2] Sell [3] Next Day [4] History [5] Exit
Choose: 2
Enter stock name to sell: ALPHA
Quantity: 5
❌ Not enough shares!
📈 Current Market Prices:
ALPHA: ₹100.92
BETA: ₹138.01
OMEGA: ₹211.42
💼 Portfolio:
BETA: 3 shares
Balance: ₹9550
📊 Options: [1] Buy [2] Sell [3] Next Day [4] History [5] Exit
Choose: 2


jupyter task 5 Last Checkpoint: 19 minutes ago
File Edit View Run Kernel Settings Help
Trusted
Code
JupyterLab Python 3 (ipykernel)
📊 Options: [1] Buy [2] Sell [3] Next Day [4] History [5] Exit
Choose: 2
Enter stock name to sell: BETA
Quantity: 2
✅ Sold 2 of BETA for ₹276.02
📈 Current Market Prices:
ALPHA: ₹100.92
BETA: ₹138.01
OMEGA: ₹211.42
💼 Portfolio:
BETA: 1 shares
Balance: ₹9826.02
📊 Options: [1] Buy [2] Sell [3] Next Day [4] History [5] Exit
Choose: 4
📜 Transaction History:
Bought 3 of BETA at ₹150
Sold 2 of BETA at ₹138.01
📈 Current Market Prices:
ALPHA: ₹100.92
BETA: ₹138.01
OMEGA: ₹211.42
💼 Portfolio:
BETA: 1 shares
Balance: ₹9826.02
📊 Options: [1] Buy [2] Sell [3] Next Day [4] History [5] Exit
Choose: 5
👋 Exiting. Final Balance: 9826.02
[12]: # sudoku validator