

Exp-11

Cutting tickets and show chair threads

Mohammed Sadriwala, 28, Batch-2

```
from threading import *
```

```
from time import *
```

```
class Theatre:
```

```
    def __init__(self, str):
```

```
        self.str=str
```

```
    def movieshow(self):
```

```
        for i in range(1,6):
```

```
            print(self.str,":",i)
```

```
            sleep(0.5)
```

```
obj1=Theatre("Cut Ticket ")
```

```
obj2=Theatre("Show Chair ")
```

```
t1=Thread(target=obj1.movieshow)
```

```
t2=Thread(target=obj2.movieshow)
```

```
t1.start()
```

```
t2.start()
```



```
Written by Hussain
Cut Ticket : 1
Show Chair : 1
Cut Ticket : 2
Show Chair : 2
Cut Ticket : 3
Show Chair : 3
Cut Ticket : 4
Show Chair : 4
Cut Ticket : 5
Show Chair : 5

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

Post-Lab question

Single thread

Mohammed Sadriwala, 28, Batch-2

```
from time import sleep
```

```
def task1():
```

```
    print("Task 1: Starting...")
```

```
    sleep(2)
```

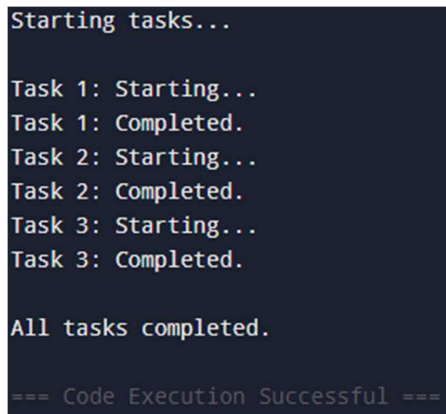
```
    print("Task 1: Completed.")
```

```
def task2():
```

```

    print("Task 2: Starting...")
    sleep(3)
    print("Task 2: Completed.")
def task3():
    print("Task 3: Starting...")
    sleep(1)
    print("Task 3: Completed.")
def main():
    print("Starting tasks...\n")
    task1()
    task2()
    task3()
    print("\nAll tasks completed.")
main()

```



```

Starting tasks...

Task 1: Starting...
Task 1: Completed.
Task 2: Starting...
Task 2: Completed.
Task 3: Starting...
Task 3: Completed.

All tasks completed.

=== Code Execution Successful ===

```

Multiple thread

Mohammed Sadriwala, 28, Batch-2

```
from threading import Thread
```

```
from time import sleep
```

```
def task1():
```

```
    print("Task 1: Starting...")
```

```
    sleep(2)
```

```
    print("Task 1: Completed.")
```

```
def task2():
```

```
    print("Task 2: Starting...")
```

```
    sleep(3)
```

```
    print("Task 2: Completed.")
```

```
def task3():
```

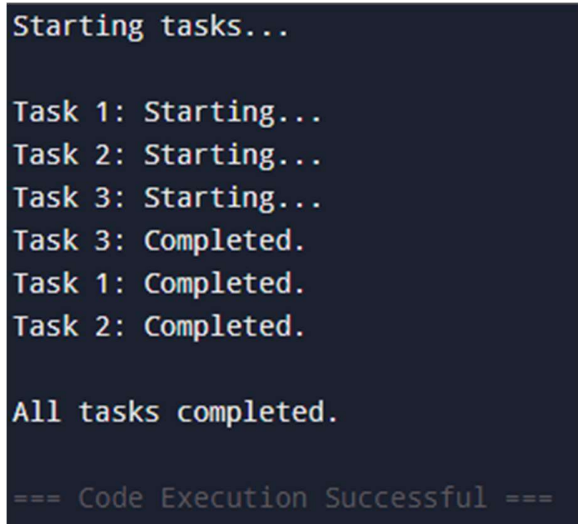
```
    print("Task 3: Starting...")
```

```
    sleep(1)
```

```

    print("Task 3: Completed.")
def main():
    print("Starting tasks...\n")
    thread1 = Thread(target=task1)
    thread2 = Thread(target=task2)
    thread3 = Thread(target=task3)
    thread1.start()
    thread2.start()
    thread3.start()
    thread1.join()
    thread2.join()
    thread3.join()
    print("\nAll tasks completed.")
main()

```



```

Starting tasks...

Task 1: Starting...
Task 2: Starting...
Task 3: Starting...
Task 3: Completed.
Task 1: Completed.
Task 2: Completed.

All tasks completed.

=== Code Execution Successful ===

```

Exp-12 NumPy program

Mohammed Sadriwala, 28, Batch-2

```
import numpy as np
```

```

def main():
    while True:
        choice = input("\nMenu:\n1. Polynomial Operations\n2. Integer Operations\n3. Exit\nEnter your choice: ")
        if choice == '1':
            poly1 = np.poly1d(list(map(int, input("Enter first polynomial coefficients: ").split()))))
            poly2 = np.poly1d(list(map(int, input("Enter second polynomial coefficients: ").split()))))
            operation = input("\n1. Add\n2. Subtract\n3. Multiply\n4. Divide\nChoose operation: ")
            if operation == '1': print("Result:", poly1 + poly2)

```

```

elif operation == '2': print("Result:", poly1 - poly2)
elif operation == '3': print("Result:", poly1 * poly2)
elif operation == '4': print("Result:", np.polydiv(poly1, poly2)[0])
else: print("Invalid operation")

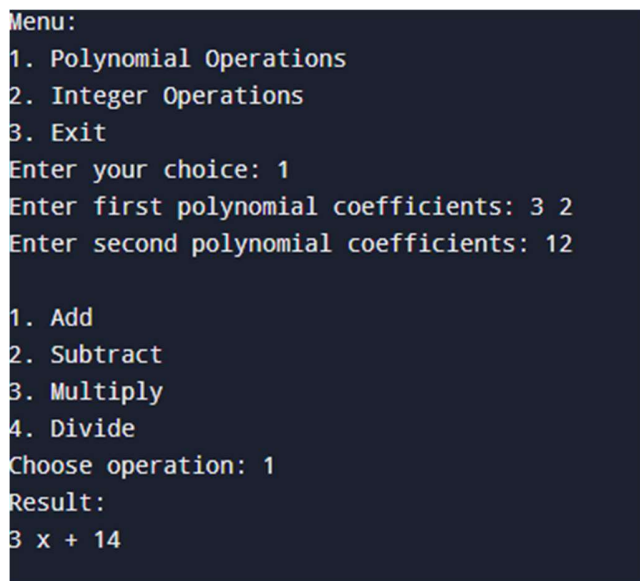
elif choice == '2':
    num1, num2 = int(input("Enter first integer: ")), int(input("Enter second integer: "))
    operation = input("\n1. Add\n2. Subtract\n3. Multiply\n4. Divide\nChoose operation: ")
    if operation == '1': print("Result:", num1 + num2)
    elif operation == '2': print("Result:", num1 - num2)
    elif operation == '3': print("Result:", num1 * num2)
    elif operation == '4': print("Result:", num1 / num2 if num2 != 0 else "Cannot divide by zero")
    else: print("Invalid operation")

elif choice == '3': break
else: print("Invalid choice.")

```

```
if __name__ == "__main__":
```

```
    main()
```



```

Menu:
1. Polynomial Operations
2. Integer Operations
3. Exit
Enter your choice: 1
Enter first polynomial coefficients: 3 2
Enter second polynomial coefficients: 12

1. Add
2. Subtract
3. Multiply
4. Divide
Choose operation: 1
Result:
3 x + 14

```

Ep-13

Mohammed Sadriwala, 28, Batch-2

```
from numpy import *
```

```
r, c = map(int, input("Enter rows and columns: ").split())
```

```
matrix_elements = list(map(int, input("Enter matrix elements: ").split()))
```

```
x = array(matrix_elements).reshape(r, c)
```

```
print("Original matrix:"); print(x)
```

```
print("Transpose matrix:"); print(x.T)
```

```
print("Diagonal of the matrix:"); print(diagonal(x))
```

```
print("Sum of diagonal elements:"); print(sum(diagonal(x)))
```

```
Enter rows and columns: 2 2
Enter matrix elements: 1 2 3 4
Original matrix:
[[1 2]
 [3 4]]
Transpose matrix:
[[1 3]
 [2 4]]
Diagonal of the matrix:
[1 4]
Sum of diagonal elements:
5

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

Post-Lab

Mohammed Sadriwala, 28, Batch-2

```
from numpy import *
```

```
n = int(input("Enter the size of square matrices (or 0 for different sizes): "))
```

```
if n > 0:
```

```
    A = array(list(map(int, input("Enter elements of first square matrix: ").split()))).reshape(n, n)
```

```
    B = array(list(map(int, input("Enter elements of second square matrix: ").split()))).reshape(n, n)
```

```
else:
```

```
    r1, c1 = map(int, input("Enter rows and columns of first matrix: ").split())
```

```
    A = array(list(map(int, input("Enter elements of first matrix: ").split()))).reshape(r1, c1)
```

```
    r2, c2 = map(int, input("Enter rows and columns of second matrix: ").split())
```

```
    B = array(list(map(int, input("Enter elements of second matrix: ").split()))).reshape(r2, c2)
```

```
if A.shape[1] != B.shape[0]:
```

```
    print("Matrix multiplication not possible")
```

```
else:
```

```
    print("Resultant matrix:"); print(dot(A, B))
```

```
Enter the size of square matrices (or 0 for different sizes): 2
Enter elements of first square matrix: 1 2 3 4
Enter elements of second square matrix: 5 6 7 8
Resultant matrix:
[[19 22]
 [43 50]]

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

Transpose matrix

Mohammed Sadriwala, 28, Batch-2

```
from numpy import *
```

```
r, c = map(int, input("Enter rows and columns of the matrix: ").split())
```

```
A = array(list(map(int, input("Enter matrix elements: ").split()))).reshape(r, c)
```

```
print("Original matrix:"); print(A)
```

```
print("Transpose of the matrix:"); print(A.T)
```

```
Enter rows and columns of the matrix: 2 2
Enter matrix elements: 1 2 3 4
Original matrix:
[[1 2]
 [3 4]]
Transpose of the matrix:
[[1 3]
 [2 4]]

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