HEAPS:

class MaxHeap:

def \_\_init\_\_(self):

self.heap = []

def \_heapify\_up(self, index):

"""Move the element at index up to maintain heap property."""

while index > 0:

parent\_index = (index - 1) // 2

if self.heap[index][0] > self.heap[parent\_index][0]:

self.heap[index], self.heap[parent\_index] = self.heap[parent\_index], self.heap[index]

index = parent\_index

else:

break

def \_heapify\_down(self, index):

"""Move the element at index down to maintain heap property."""

size = len(self.heap)

largest = index

left\_child\_index = 2 \* index + 1

right\_child\_index = 2 \* index + 2

if left\_child\_index < size and self.heap[left\_child\_index][0] > self.heap[largest][0]:

largest = left\_child\_index

if right\_child\_index < size and self.heap[right\_child\_index][0] > self.heap[largest][0]:

largest = right\_child\_index

if largest != index:

self.heap[index], self.heap[largest] = self.heap[largest], self.heap[index]

self.\_heapify\_down(largest)

def insert(self, priority, job):

"""Insert a new job with a given priority."""

self.heap.append((priority, job))

self.\_heapify\_up(len(self.heap) - 1)

def delete\_max(self):

"""Remove and return the job with the highest priority."""

if not self.heap:

return None

max\_job = self.heap[0]

self.heap[0] = self.heap.pop()

self.\_heapify\_down(0)

return max\_job

def peek(self):

"""Return the job with the highest priority without removing it."""

if self.heap:

return self.heap[0]

return None

def display\_all(self):

"""Display all jobs in heap order."""

return [(priority, job) for priority, job in self.heap]

def main():

max\_heap = MaxHeap()

while True:

print("\nJob Scheduler Menu:")

print("1. Insert a job")

print("2. Delete highest priority job")

print("3. Peek at the highest priority job")

print("4. Display all jobs in heap order")

print("5. Exit")

choice = input("Enter your choice: ")

if choice == "1":

priority = int(input("Enter job priority: "))

job = input("Enter job description: ")

max\_heap.insert(priority, job)

print(f"Job '{job}' with priority {priority} inserted.")

elif choice == "2":

max\_job = max\_heap.delete\_max()

if max\_job:

print(f"Job '{max\_job[1]}' with priority {max\_job[0]} removed.")

else:

print("No jobs to remove.")

elif choice == "3":

max\_job = max\_heap.peek()

if max\_job:

print(f"Highest priority job: '{max\_job[1]}' with priority {max\_job[0]}")

else:

print("No jobs available.")

elif choice == "4":

jobs = max\_heap.display\_all()

if jobs:

print("All jobs in heap order:")

for priority, job in jobs:

print(f"Priority: {priority}, Job: {job}")

else:

print("No jobs available.")

elif choice == "5":

print("Exiting Job Scheduler.")

break

else:

print("Invalid choice. Please try again.")

if \_\_name\_\_ == "\_\_main\_\_":

main()

OUTPUT:

Job Scheduler Menu:

1. Insert a job

2. Delete highest priority job

3. Peek at the highest priority job

4. Display all jobs in heap order

5. Exit

Enter your choice: 1

Enter job priority: 1

Enter job description: add

Job 'add' with priority 1 inserted.

Job Scheduler Menu:

1. Insert a job

2. Delete highest priority job

3. Peek at the highest priority job

4. Display all jobs in heap order

5. Exit

Enter your choice: 1

Enter job priority: 3

Enter job description: sub

Job 'sub' with priority 3 inserted.

Job Scheduler Menu:

1. Insert a job

2. Delete highest priority job

3. Peek at the highest priority job

4. Display all jobs in heap order

5. Exit

Enter your choice: 3

Highest priority job: 'sub' with priority 3

Job Scheduler Menu:

1. Insert a job

2. Delete highest priority job

3. Peek at the highest priority job

4. Display all jobs in heap order

5. Exit

Enter your choice: 4

All jobs in heap order:

Priority: 3, Job: sub

Priority: 1, Job: add

Job Scheduler Menu:

1. Insert a job

2. Delete highest priority job

3. Peek at the highest priority job

4. Display all jobs in heap order

5. Exit

Enter your choice: 2

Job 'sub' with priority 3 removed.

Job Scheduler Menu:

1. Insert a job

2. Delete highest priority job

3. Peek at the highest priority job

4. Display all jobs in heap order

5. Exit

Enter your choice: 4

All jobs in heap order:

Priority: 1, Job: add

Job Scheduler Menu:

1. Insert a job

2. Delete highest priority job

3. Peek at the highest priority job

4. Display all jobs in heap order

5. Exit

Enter your choice: 5

Exiting Job Scheduler.