class Job:

def \_\_init\_\_(self, job\_id, profit, deadline):

self.job\_id = job\_id

self.profit = profit

self.deadline = deadline

def job\_scheduling(jobs, max\_slots):

jobs.sort(key=lambda job: job.profit, reverse=True)

slots = [False] \* max\_slots

scheduled\_jobs = [None] \* max\_slots

for job in jobs:

for slot in range(min(max\_slots, job.deadline) - 1, -1, -1):

if not slots[slot]:

slots[slot] = True

scheduled\_jobs[slot] = job.job\_id

break

return [job for job in scheduled\_jobs if job is not None]

def take\_input():

num\_jobs = int(input("Enter the total number of jobs: "))

jobs = []

for i in range(num\_jobs):

print(f"\nEnter details for Job {i + 1}:")

job\_id = input("Job ID: ")

profit = int(input("Profit (integer): "))

deadline = int(input("Deadline (integer): "))

jobs.append(Job(job\_id, profit, deadline))

return jobs, max(job.deadline for job in jobs)

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

jobs, max\_slots = take\_input()

scheduled\_jobs = job\_scheduling(jobs, max\_slots)

print("\nJobs scheduled for maximum profit:", scheduled\_jobs)