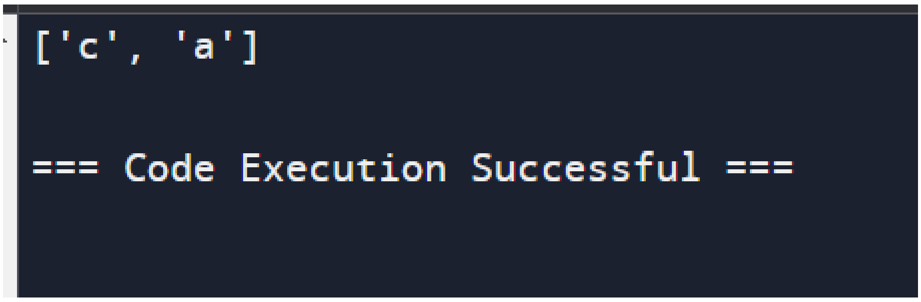
90. Job Sequencing with Deadlines

PROGRAM:-

```
class Job:
    def _init_(self, job_id, deadline, profit):
        self.job_id = job_id
        self.deadline = deadline
        self.profit = profit

def job_scheduling(jobs):
    # Sort jobs according to descending order of profit
    jobs.sort(key=lambda x: x.profit, reverse=True)

    n = len(jobs)

    # Find the maximum deadline
    max_deadline = max(job.deadline for job in jobs)

    # Create a result list to store the result (Sequence of jobs)
    result = [None] * max_deadline

    # Create a list to keep track of free time slots
    slot = [False] * max_deadline

    # Iterate through all given jobs
    for job in jobs:
        # Find a free slot for this job (Note that we start from the last possible slot)
        for j in range(min(max_deadline, job.deadline) - 1, -1, -1):
            if not slot[j]:
                slot[j] = True
                result[j] = job.job_id
                break

    # Filter out None values and return the job sequence
    return [job_id for job_id in result if job_id is not None]

# Example usage
jobs = [
    Job('a', 4, 20),
    Job('b', 1, 10),
    Job('c', 1, 40),
    Job('d', 1, 30)
]

print(job_scheduling(jobs))  # Output: ['c', 'a']
```

OUTPUT:-



```
['c', 'a']

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

TIME COMPLEXITY:-O(n logn)