**Assignment No: 2**

**Code :**

import java.util.Arrays;

import java.util.Scanner;

class Job {

    char id;

    int dead;

    int profit;

    public Job(char id, int dead, int profit) {

        this.id = id;

        this.dead = dead;

        this.profit = profit;

    }

}

public class JobScheduling {

    static class JobComparator implements java.util.Comparator<Job> {

        public int compare(Job a, Job b) {

            return b.profit - a.profit;

        }

    }

    public static void printJobScheduling(Job[] arr, int n) {

        Arrays.sort(arr, new JobComparator());

        int result[] = new int[n]; // To store result (Sequence of jobs)

        boolean slot[] = new boolean[n]; // To keep track of free time slots

        for (int i = 0; i < n; i++) {

            slot[i] = false;

        }

        for (int i = 0; i < n; i++) {

            for (int j = Math.min(n, arr[i].dead) - 1; j >= 0; j--) {

                if (!slot[j]) {

                    result[j] = i;

                    slot[j] = true;

                    break;

                }

            }

        }

        System.out.print("Following is the maximum profit sequence of jobs: ");

        for (int i = 0; i < n; i++) {

            if (slot[i]) {

                System.out.print(arr[result[i]].id + " ");

            }

        }

        System.out.println();

    }

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter the number of jobs: ");

        int n = scanner.nextInt();

        Job[] arr = new Job[n];

        for (int i = 0; i < n; i++) {

            System.out.print("Enter job ID, deadline, and profit for job " + (i + 1) + ": ");

            char id = scanner.next().charAt(0);

            int deadline = scanner.nextInt();

            int profit = scanner.nextInt();

            arr[i] = new Job(id, deadline, profit);

        }

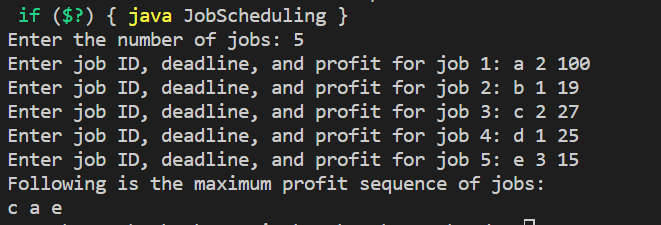
        printJobScheduling(arr, n);

        scanner.close();

    }

}

**Output:**

****