AMRITA VISHWA VIDYAPEETHAM
DEEMED TO BE UNIVERSITY UNDER SECTION 3 OF UGC ACT, 1956

SCHOOL OF COMPUTING

DESIGN AND ANALYSIS OF ALGORITHMS

LAB WORKBOOK

WEEK - 7

NAME : P JEEVAN SANDEEP

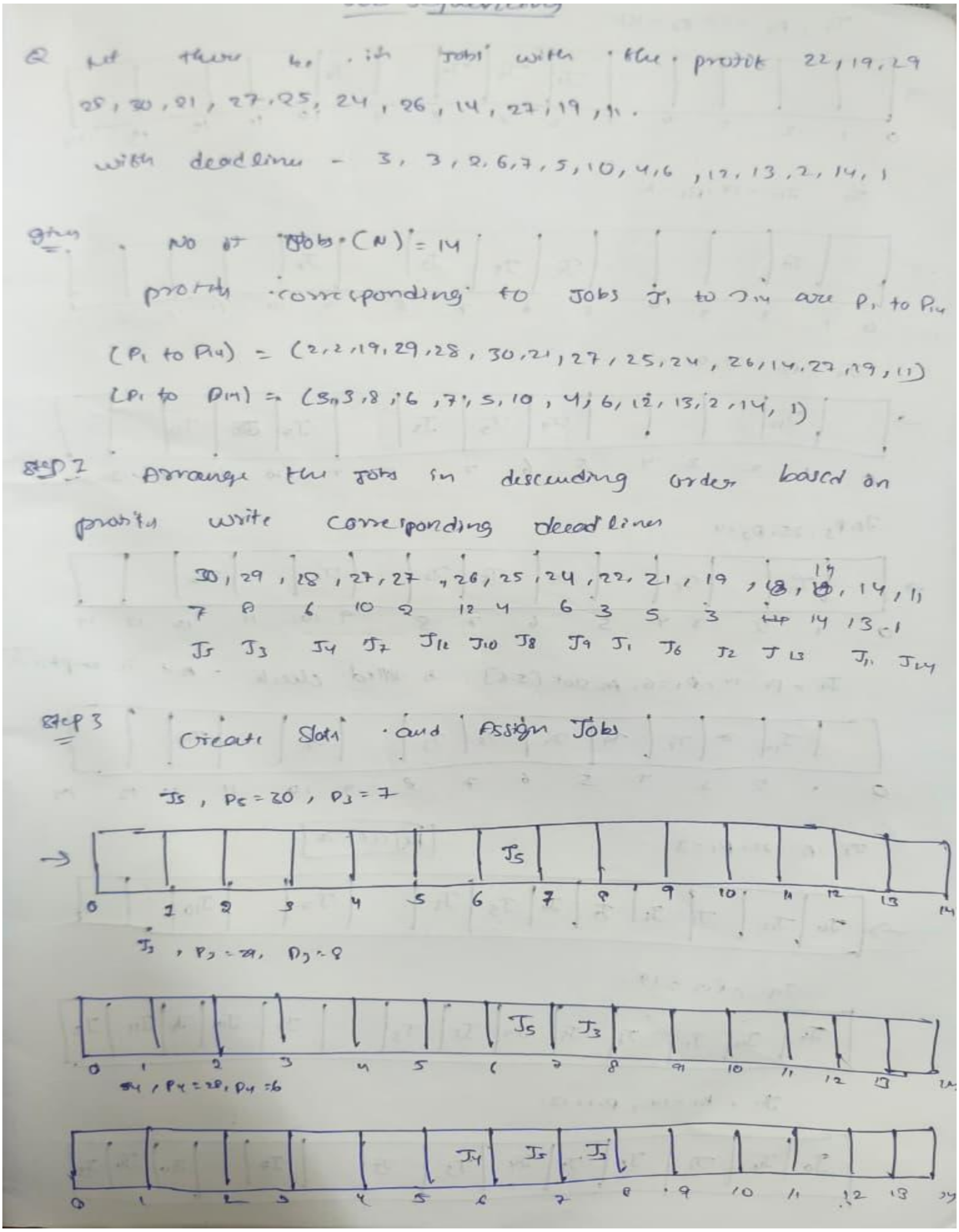
ROLL NUMBER : CH.SC.U4CSE24134

CLASS : CSE-B

Question 1: Let there be 14 jobs with the profit of 22,19,29,28,30,21,27,25,24,26,14,27,19,11 with deadlines 3,3,8,6,7,5,10,4,6,12,13,2,14,1

Implement the greedy algorithm for the Job Sequencing with Deadlines and determine the optimal sequence of jobs that maximizes total profit

WORKING:



Q let there be 14 Jobs with the profit 22, 19, 29, 28, 30, 21, 27, 25, 24, 26, 14, 27, 19, 11.

with deadline - 3, 3, 8, 6, 7, 5, 10, 4, 6, 12, 13, 2, 14, 1

given . No of Jobs (N) = 14

profits corresponding to Jobs J_1 to J_{14} are P_1 to P_{14}

$(P_1$ to $P_{14}) = (22, 19, 29, 28, 30, 21, 27, 25, 24, 26, 14, 27, 19, 11)$

$(D_1$ to $D_{14}) = (3, 3, 8, 6, 7, 5, 10, 4, 6, 12, 13, 2, 14, 1)$

Step 2 Arrange the Jobs in descending order based on profits write corresponding deadlines

30	29	28	27	27	26	25	24	22	21	19	~~18~~ 19	~~16~~ 14	11
7	8	6	10	2	12	4	6	3	5	3	14	13	1
J_5	J_3	J_4	J_7	J_{12}	J_{10}	J_8	J_9	J_1	J_6	J_2	J_{13}	J_{11}	J_{14}

Step 3 Create Slots and Assign Jobs

J_5 , $P_5 = 30$, $D_5 = 7$

J_3 , $P_3 = 29$, $D_3 = 8$

J_4 , $P_4 = 28$, $D_4 = 6$

J_7 , $P_7 = 27$, $D_7 = 10$

					J_4	J_5	J_3		J_7				

→ 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14

J_{12} $J_{12} = 27$, $D_{12} = 2$

	J_{12}				J_4	J_5	J_3		J_7				

→ 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14

J_{10} , $P_{10} = 20$, $D_{10} = 12$

	J_{12}				J_4	J_5	J_3		J_7	~~[illegible]~~	J_{10}		

→ 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14

J_8, $P_3 = 25$, $D_3 = 4$

	J_{12}		J_8		J_4	J_5	J_3		J_7		J_{10}		

→ 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14

J_9 , $P_9 = 24$, $D_9 = 6$, As slot [5-6] is filled check. As it is empty

	J_{12}	[illegible]	J_8	J_9	J_4	J_5	J_3		J_7		J_{10}		

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14

J_1 , $P_1 = 22$, $D_1 = 3$ [Reject - J_2]

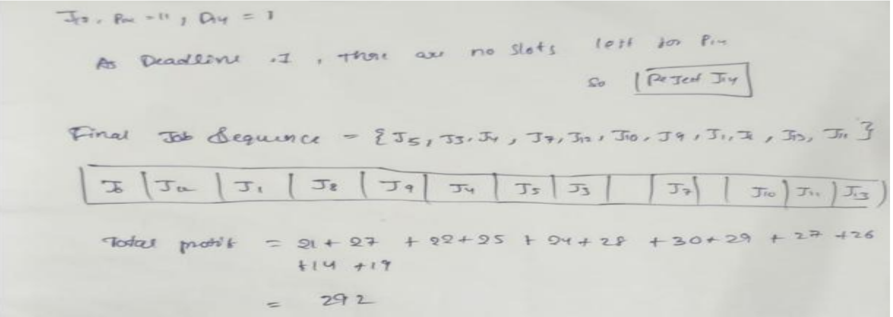
J_6	J_{12}	J_1	J_8	J_9	J_4	J_5	J_3		J_7		J_{10}		

J_{13} , $P_{13} = 19$.

J_6	J_{12}	J_1	J_8	J_9	J_4	J_5	J_3		J_7	J_{10}	J_{11}	J_{11}	J_{13}

J_{11} , $P_{11} = 14$, $D_{11} = 13$

J_0	J_{12}	J_1	J_8	J_9	J_4	J_5	J_3		J_7		J_{10}	J_{11}	J_{13}



J14 , P14 = 11 ; D14 = 1

As Deadline 1 , there are no slots left for P14

So Reject J14

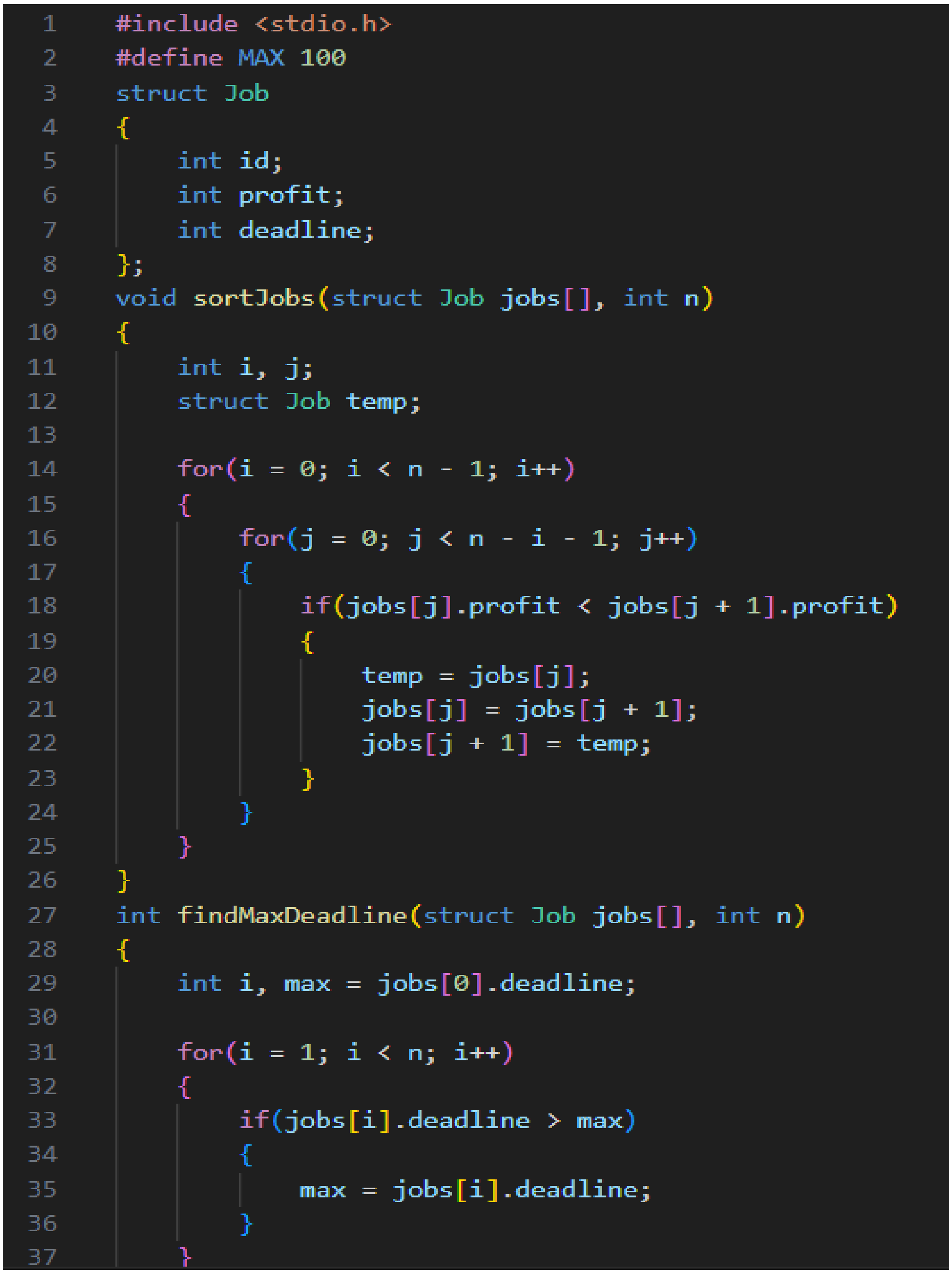
Final Job Sequence = {J5, J3, J4 , J7, J12 , J10 , J9 , J1, J2 , J13, J11 }

J6	J2	J1	J8	J9	J4	J5	J3		J7		J10	J11	J13

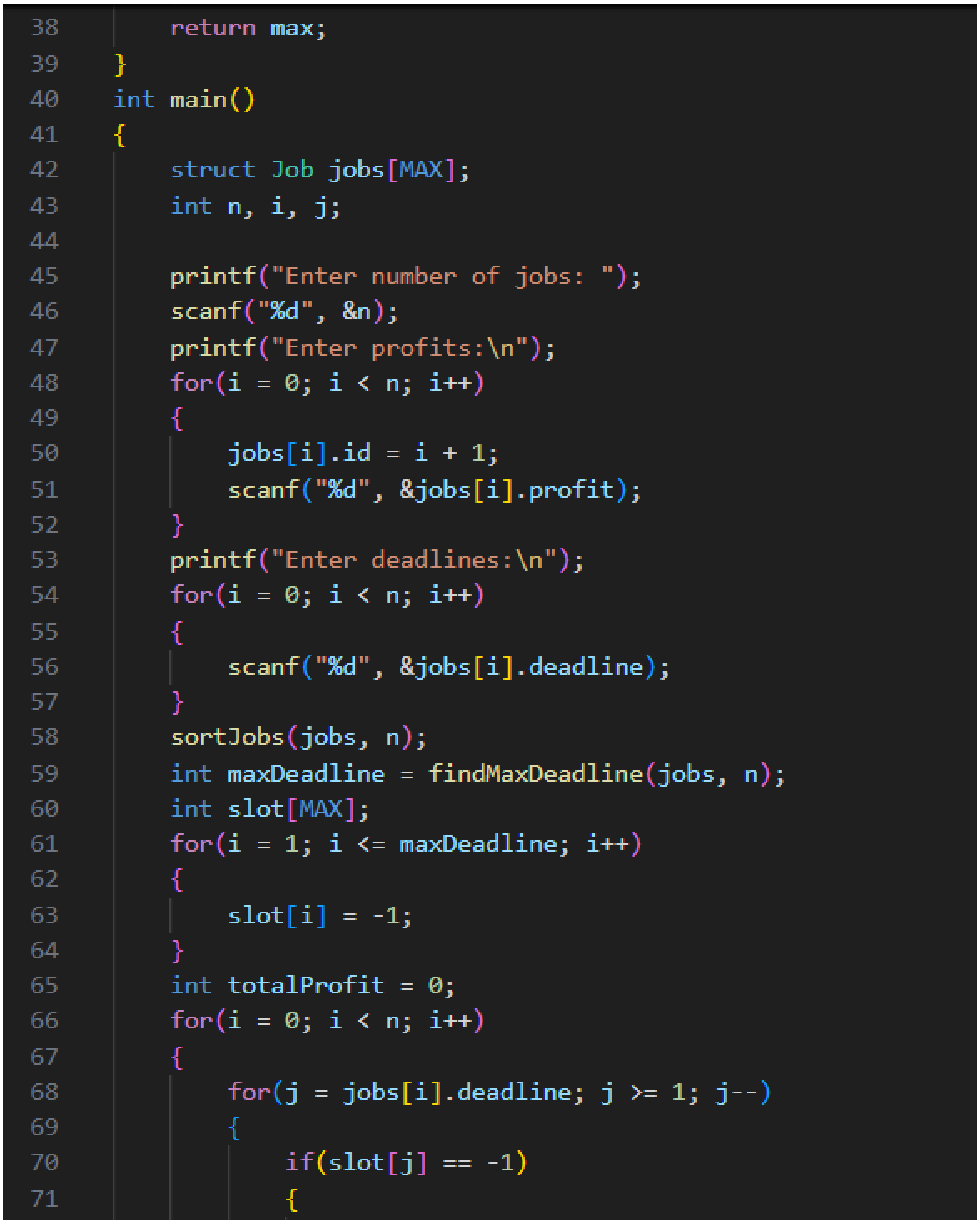
Total profit = 21 + 27 + 22+25 + 24+28 + 30+29 + 27 +26 +14 +19

= 292

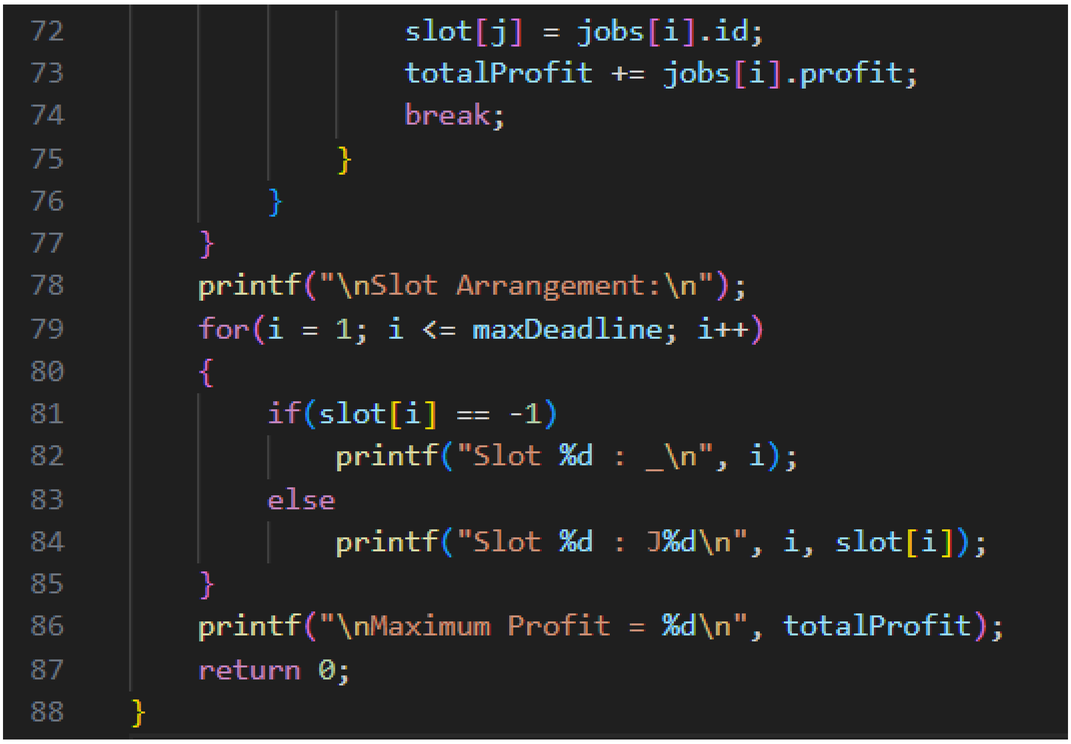
CODE:



```
#include <stdio.h>
#define MAX 100
struct Job
{
    int id;
    int profit;
    int deadline;
};
void sortJobs(struct Job jobs[], int n)
{
    int i, j;
    struct Job temp;

    for(i = 0; i < n - 1; i++)
    {
        for(j = 0; j < n - i - 1; j++)
        {
            if(jobs[j].profit < jobs[j + 1].profit)
            {
                temp = jobs[j];
                jobs[j] = jobs[j + 1];
                jobs[j + 1] = temp;
            }
        }
    }
}
int findMaxDeadline(struct Job jobs[], int n)
{
    int i, max = jobs[0].deadline;

    for(i = 1; i < n; i++)
    {
        if(jobs[i].deadline > max)
        {
            max = jobs[i].deadline;
        }
    }
```

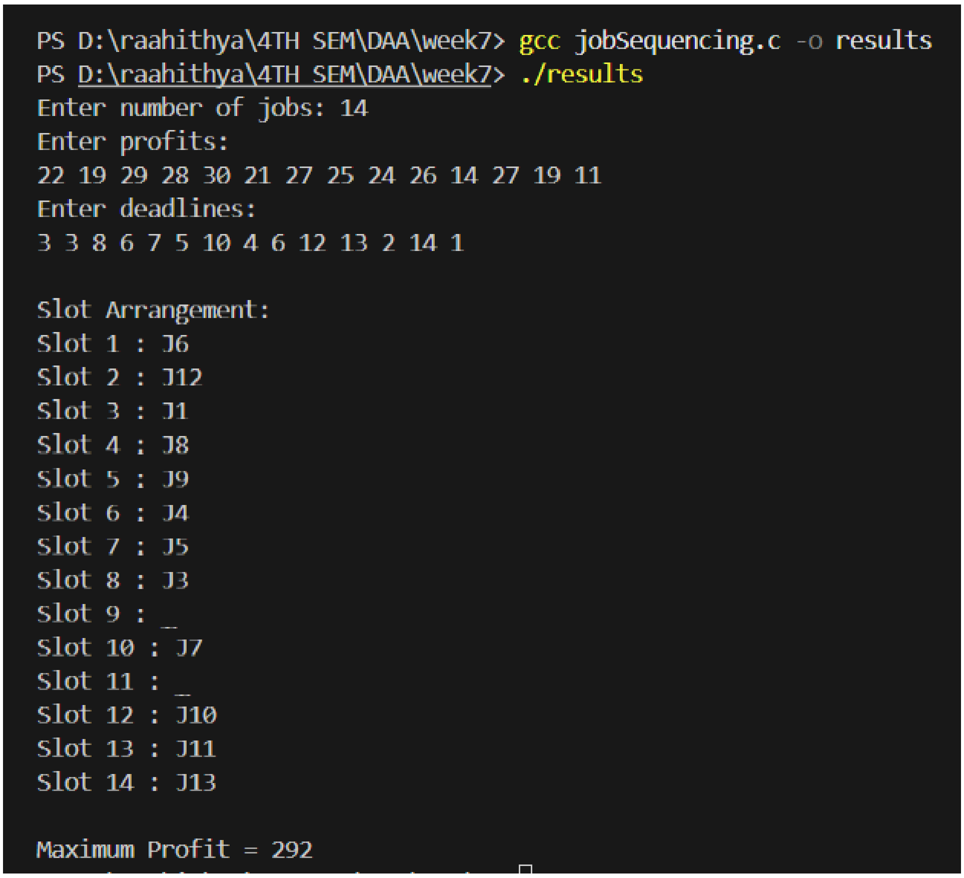


```c
    return max;
}
int main()
{
    struct Job jobs[MAX];
    int n, i, j;

    printf("Enter number of jobs: ");
    scanf("%d", &n);
    printf("Enter profits:\n");
    for(i = 0; i < n; i++)
    {
        jobs[i].id = i + 1;
        scanf("%d", &jobs[i].profit);
    }
    printf("Enter deadlines:\n");
    for(i = 0; i < n; i++)
    {
        scanf("%d", &jobs[i].deadline);
    }
    sortJobs(jobs, n);
    int maxDeadline = findMaxDeadline(jobs, n);
    int slot[MAX];
    for(i = 1; i <= maxDeadline; i++)
    {
        slot[i] = -1;
    }
    int totalProfit = 0;
    for(i = 0; i < n; i++)
    {
        for(j = jobs[i].deadline; j >= 1; j--)
        {
            if(slot[j] == -1)
            {
```



```c
                slot[j] = jobs[i].id;
                totalProfit += jobs[i].profit;
                break;
            }
        }
    }
    printf("\nSlot Arrangement:\n");
    for(i = 1; i <= maxDeadline; i++)
    {
        if(slot[i] == -1)
            printf("Slot %d : _\n", i);
        else
            printf("Slot %d : J%d\n", i, slot[i]);
    }
    printf("\nMaximum Profit = %d\n", totalProfit);
    return 0;
}
```

OUTPUT:



```
PS D:\raahithya\4TH SEM\DAA\week7> gcc jobSequencing.c -o results
PS D:\raahithya\4TH SEM\DAA\week7> ./results
Enter number of jobs: 14
Enter profits:
22 19 29 28 30 21 27 25 24 26 14 27 19 11
Enter deadlines:
3 3 8 6 7 5 10 4 6 12 13 2 14 1

Slot Arrangement:
Slot 1 : J6
Slot 2 : J12
Slot 3 : J1
Slot 4 : J8
Slot 5 : J9
Slot 6 : J4
Slot 7 : J5
Slot 8 : J3
Slot 9 : _
Slot 10 : J7
Slot 11 : _
Slot 12 : J10
Slot 13 : J11
Slot 14 : J13

Maximum Profit = 292
```

Time Complexity:

1. Sorting the jobs by profit

We used Bubble Sort in the program.

Time complexity:$O(n^2)$

2. Finding maximum deadline

We check all jobs once.

Time complexity:$O(n)$

3. Assigning jobs to slots

For each job, we may check up to d slots.$O(n^2)$

Total Time Complexity

$\boldsymbol{O(n^2) + O(n) + O(n^2)= O(n^2)}$

Space Complexity

We use:

• Job array → O(n)
• Slot array → O(d)

Total Space: O(n)