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

SHESHANK

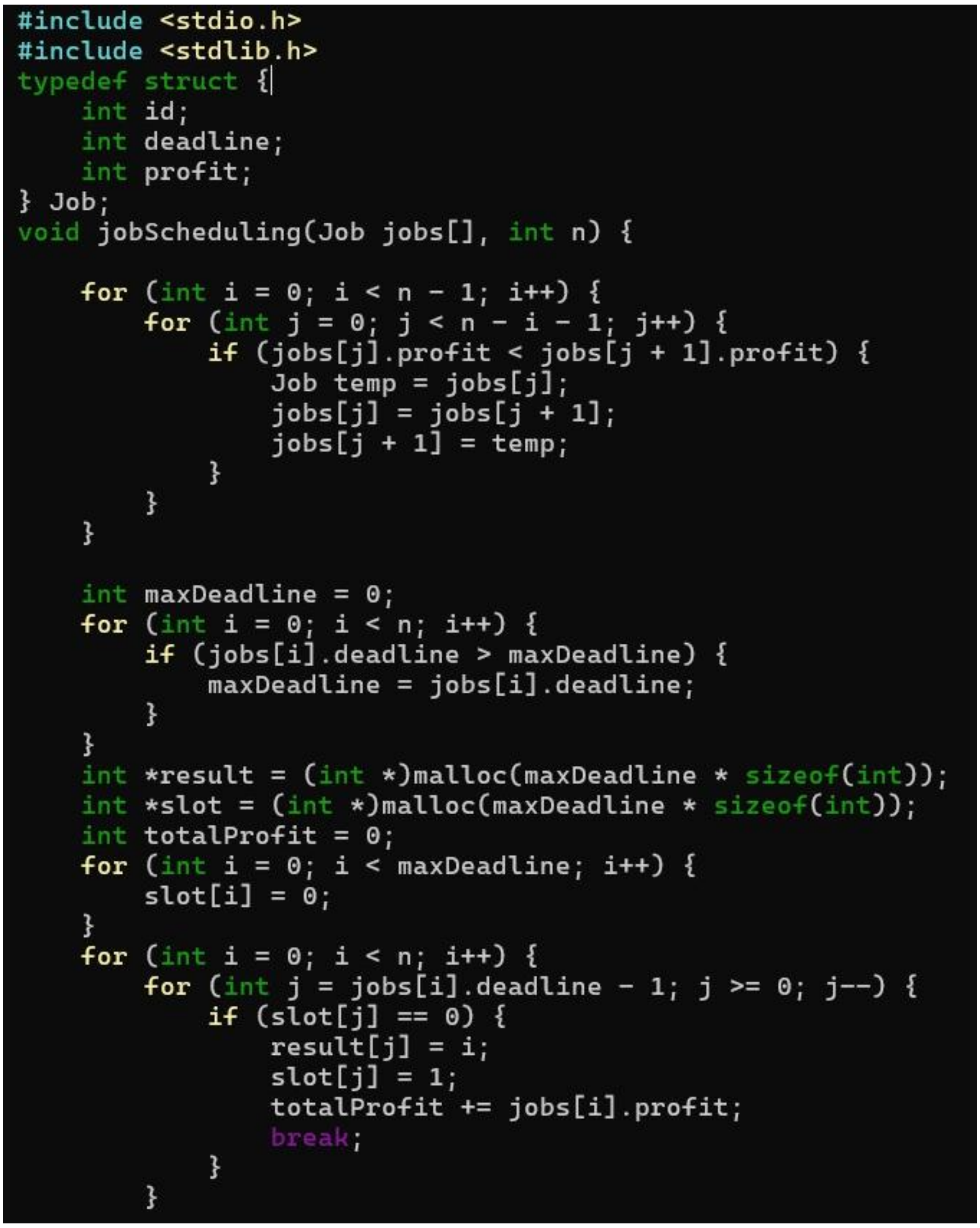
CH.SC.U4CSE24125

WEEK -6

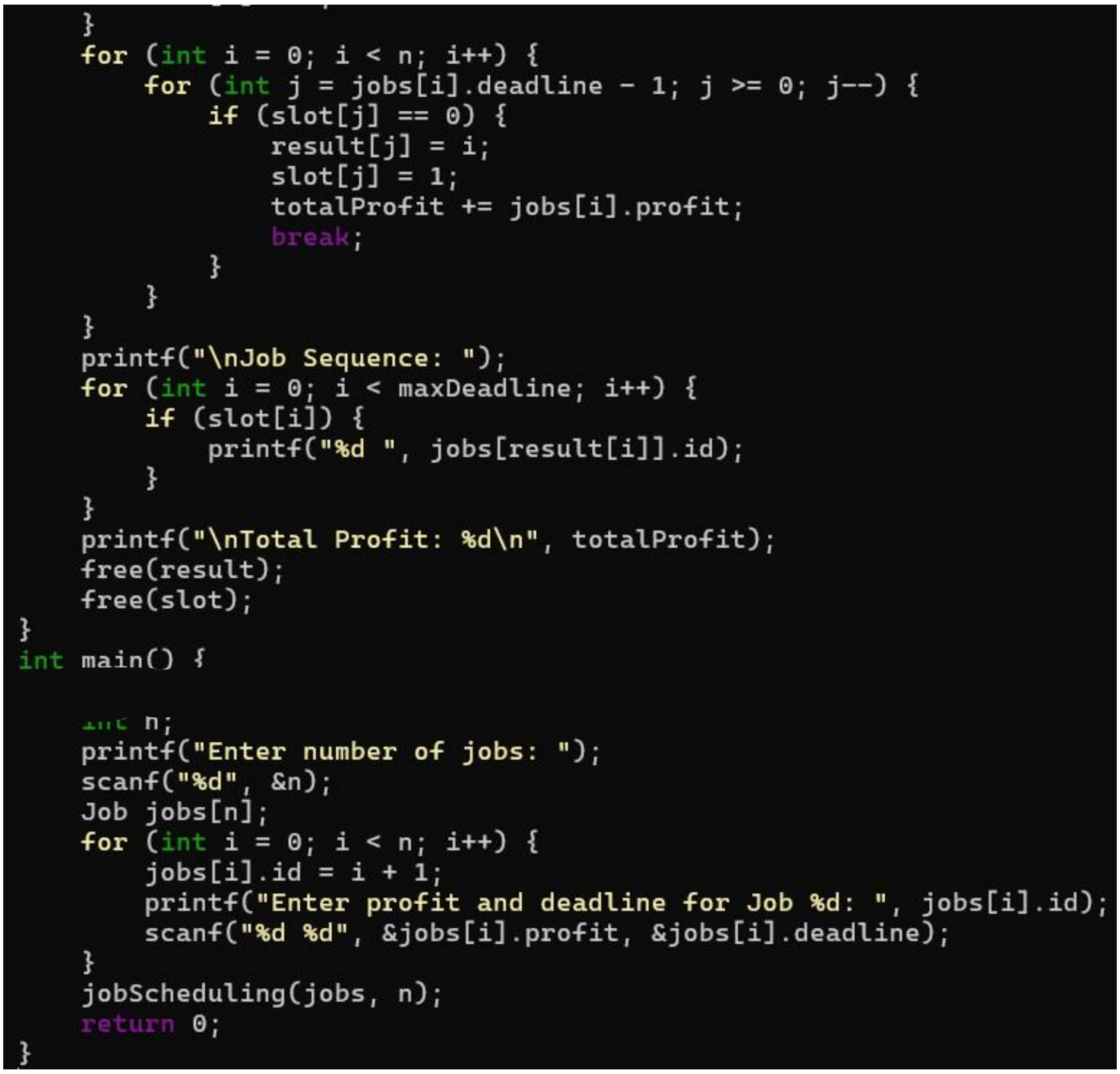
Design and Analysis of Algorithm(23CSE211)

Job -sequencing

Code:

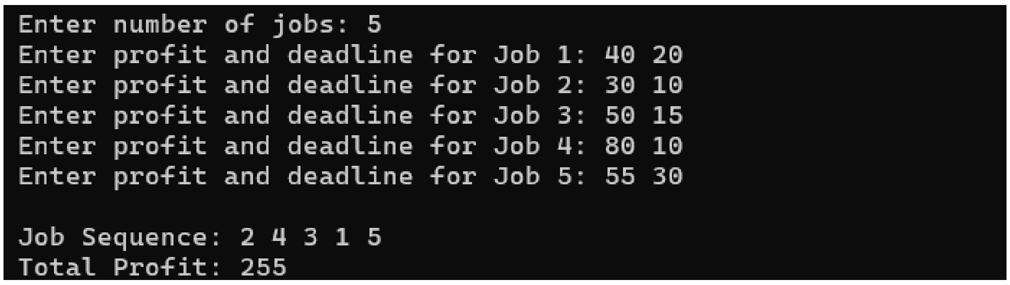


```c
#include <stdio.h>
#include <stdlib.h>
typedef struct {
    int id;
    int deadline;
    int profit;
} Job;
void jobScheduling(Job jobs[], int n) {

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

    int maxDeadline = 0;
    for (int i = 0; i < n; i++) {
        if (jobs[i].deadline > maxDeadline) {
            maxDeadline = jobs[i].deadline;
        }
    }
    int *result = (int *)malloc(maxDeadline * sizeof(int));
    int *slot = (int *)malloc(maxDeadline * sizeof(int));
    int totalProfit = 0;
    for (int i = 0; i < maxDeadline; i++) {
        slot[i] = 0;
    }
    for (int i = 0; i < n; i++) {
        for (int j = jobs[i].deadline - 1; j >= 0; j--) {
            if (slot[j] == 0) {
                result[j] = i;
                slot[j] = 1;
                totalProfit += jobs[i].profit;
                break;
            }
        }
```



```c
    }
    for (int i = 0; i < n; i++) {
        for (int j = jobs[i].deadline - 1; j >= 0; j--) {
            if (slot[j] == 0) {
                result[j] = i;
                slot[j] = 1;
                totalProfit += jobs[i].profit;
                break;
            }
        }
    }
    printf("\nJob Sequence: ");
    for (int i = 0; i < maxDeadline; i++) {
        if (slot[i]) {
            printf("%d ", jobs[result[i]].id);
        }
    }
    printf("\nTotal Profit: %d\n", totalProfit);
    free(result);
    free(slot);
}
int main() {

    int n;
    printf("Enter number of jobs: ");
    scanf("%d", &n);
    Job jobs[n];
    for (int i = 0; i < n; i++) {
        jobs[i].id = i + 1;
        printf("Enter profit and deadline for Job %d: ", jobs[i].id);
        scanf("%d %d", &jobs[i].profit, &jobs[i].deadline);
    }
    jobScheduling(jobs, n);
    return 0;
}
```

Output:



```
Enter number of jobs: 5
Enter profit and deadline for Job 1: 40 20
Enter profit and deadline for Job 2: 30 10
Enter profit and deadline for Job 3: 50 15
Enter profit and deadline for Job 4: 80 10
Enter profit and deadline for Job 5: 55 30

Job Sequence: 2 4 3 1 5
Total Profit: 255
```

Time Complexity

The time complexity of the program is $O(n^2)$. This is because the jobs are sorted using bubble sort, which takes $O(n^2)$ time in the worst case. The job scheduling step runs in $O(n \times d)$, where d is the maximum deadline, but since bubble sort dominates the execution time, the overall time complexity remains $O(n^2)$.

Space Complexity

The space complexity of the program is $O(d)$, where d is the maximum deadline. This extra space is used for the slot and result arrays, which store job assignments and occupied time slots. Apart from these arrays, only constant extra space is used for variables.