Design Analysis and Algorithm – Lab Work

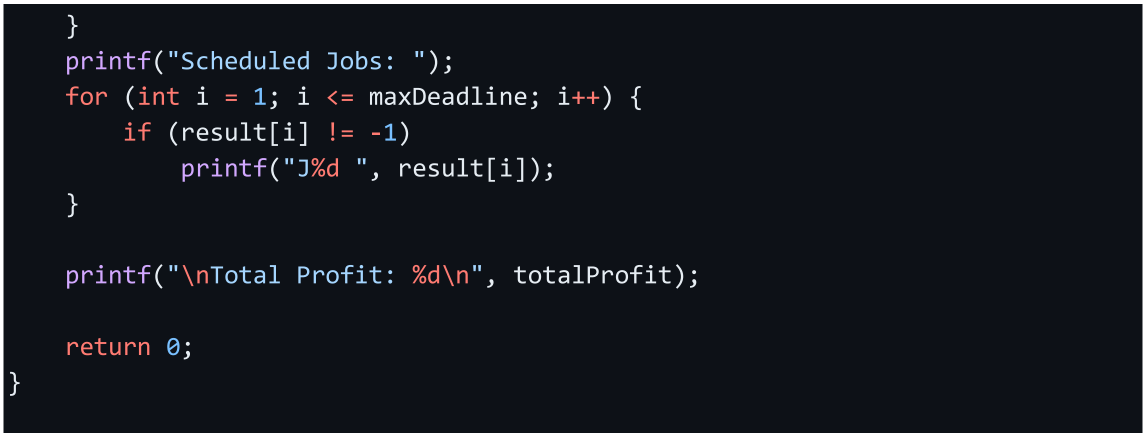
Week 6

Question 1: Write a program to implement Job Scheduling Algorithm using Greedy Aproach.

Code:

```
#include <stdio.h>
#include <stdlib.h>
typedef struct {
    int id;
    int profit;
    int deadline;
} Job;
int compare(const void *a, const void *b) {
    Job *jobA = (Job *)a;
    Job *jobB = (Job *)b;
    return jobB->profit - jobA->profit;
}
int main() {
    int n = 14;
    Job jobs[] = {
        {1, 22, 3}, {2, 19, 3}, {3, 29, 8}, {4, 28, 6},
        {5, 30, 7}, {6, 21, 5}, {7, 27, 10}, {8, 25, 4},
        {9, 24, 6}, {10, 26, 12}, {11, 14, 13}, {12, 27, 2},
        {13, 19, 14}, {14, 11, 1}
    };
    qsort(jobs, n, sizeof(Job), compare);
    int maxDeadline = 0;
    for (int i = 0; i < n; i++) {
        if (jobs[i].deadline > maxDeadline)
            maxDeadline = jobs[i].deadline;
    }
    int result[maxDeadline + 1];
    for (int i = 0; i <= maxDeadline; i++)
        result[i] = -1;

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

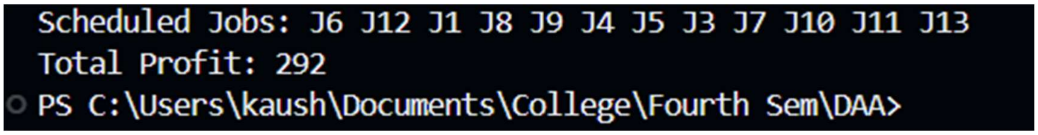


```
    }
    printf("Scheduled Jobs: ");
    for (int i = 1; i <= maxDeadline; i++) {
        if (result[i] != -1)
            printf("J%d ", result[i]);
    }

    printf("\nTotal Profit: %d\n", totalProfit);

    return 0;
}
```

Output:



```
 Scheduled Jobs: J6 J12 J1 J8 J9 J4 J5 J3 J7 J10 J11 J13
 Total Profit: 292
○ PS C:\Users\kaush\Documents\College\Fourth Sem\DAA>
```

Space Complexity: O(n^2)

Time Complexity: O(n+d) {where n = number of jobs , d = max deadline}