

Name : Deepankar Sharma  
Student ID : 233512018

### Job scheduling

```
#include <stdio.h>
#include <string.h>
#define MAX 100

typedef struct Job {
    char id[5]; int deadline; int profit;
} Job;
```

```
void jobsequencingwithdeadline(Job jobs[], int n) {
```

```
    int i, j, k, maxprofit;
    int timeslot[MAX];
    int filledtimeslot = 0;
```

```
    int dmax = 0;
```

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

```
    }
    for (i = 1; i <= dmax; i++) timeslot[i] = -1;
    printf("dmax : %d\n", dmax);
```

```
    for (i = 0+1; i <= n; i++) {
```

```
        k = minvalue(dmax, jobs[i].deadline);
```

```
        while (k >= 1) {
```

```
            if (timeslot[k] == -1) {
```

```
                timeslot[k] = i; filledtimeslot++; break;
```

```
            } k--;
```

```
        }
```

```
        if (filledtimeslot == dmax) break;
```

```
    }
    printf("Required jobs ");
```

```
    for (i = 1; i <= dmax; i++) {
```

```
        printf("%d", jobs[timeslot[i]].id);
```

```
        if (i < dmax)
            printf(" --> ");
```

```
    }
```

### Complexity Analysis

① Bubble Sort =  $O(n^2)$

② Job sequencing

$O(n^2)$

$O(n \log n)$   
(if Merge Sort)

```

// maximum profit
max_profit = 0;
for (i = 1; i <= dmax; i++) {
    max_profit += jobs[timeslot[i]].profit;
}
printf("Max Profit: %d\n", max_profit);
}

```

```

}

```

```

int main() {
    int i, j;
    Job jobs[10], temp;
    int n;
    printf("Enter the # jobs: "); scanf("%d", &n);

    for (i = 1; i <= n; i++) {
        scanf("%d %d %d", &jobs[i].id, &jobs[i].deadline,
            &jobs[i].profit);
    }
}

```

```

// descending order of profit
for (i = 1; i <= n-1; i++) {
    for (j = 1; j <= n-1; j++) {
        if (jobs[j+1].profit > jobs[j].profit) {
            temp = jobs[j+1];
            jobs[j+1] = jobs[j];
            jobs[j] = temp;
        }
    }
}
}

```

```

jobssequencingwithdeadline(jobs, n);

```

```

return 0;

```

```

}

```