Date _____ Expt. No. 5.

Expt. Name _____ Page No. 13.

AIM: Perform Job Sequencing with a deadline using the Greedy Approach using C/C++.

Algorithm:

1. Begin
2. Sort all the jobs based on profit Pi so
3. P1 > P2 > P3 ..... >= Pn
4. d = maximum deadline of job in A.
5. Create array S[1, ... , d]
6. For j=1 to n. do
7. Find the largest job x.
8. For j= i to 1.
9. If ((S[j] = 0) and (x deadline <= d))
10. Then
11. S[x] = i;
12. Break;
13. End if
14. End for
15. End for
16. End.

Program Code:

```
# include <stdio.h>
# include <stdlib.h>
# include <stdbool.h>

// Job Structure
```

Teacher's Signature: _____

Date

Expt. No.

Expt. Name

Page No. 14.

```
typedef structure {
    char id;
    int deadline;
    int profit;
} job;

// Function to compare two jobs based on their profit.
Returns true if job b's profit < job a's profit

int compareJob (const Job *a, const Job *b) {
        return b -> profit - a -> profit;
}

// Function finds the best job sequence

void bestJob (job jobs[], int sizeOfJobs) {
    // Null char array.
    char jobsToDo[5] = {'\0'};
    for (int i=0; k>=0; i<sizeOfJobs; i++) {
        k = jobs[i].deadline - 1;
    // Searching backwards the empty date nearest
    to deadline

    while (jobsToDo[k] != '\0' && k >= 0) {
        k--;
    }
    // if empty data found, set the jobs.
    if (k != -1)
        jobsToDo[k] = jobs[i].id;
    }
```

Teacher's Signature:

Date _____ Expt. No. _____

Expt. Name _____ Page No. 15.

```
// Output the final job sequence.

printf("\n Best order and Jobs to do is : ");

int idx = 0;
while(jobsToDo[idx] != '\0') {
    printf("%c ", jobsToDo[idx]);
    idx++;
}
}

// Function to display the jobs table

void display (Job jobs[], int n) {
    printf("Job Id : \t");
    for(int i=0; i<n; i++) {
        printf("%c \t", jobs[i].id);
    }
    printf("\n");

printf("Job Deadline : \t");
for(int i=0; i<n; i++) {
    printf("%d \t", jobs[i].deadline);
}

printf("\n");

printf("Job Profit \t");
for(int i=0; i<n; i++) {
    printf("%d \t", jobs[i].profit);
}

printf("\n");
}
```

Teacher's Signature: _____

Output:

Job ID :	w	v	x	y	z
Job Deadline :	1	2	2	1	3
Job Profit :	19	100	27	25	15

the Best Order and

Jobs to Do is : x v z

Date _____ Expt. No. _____

Expt. Name _____ Page No. 16.

```
int main() {

// initialize the jobs
Job jobs[] = {{'w', 1, 19}, {'v', 2, 100}, {'x', 2, 27},
              {'y', 1, 25}, {'z', 3, 15}};

// Display the jobs data
   display( jobs, 5 );

// Sorting jobs[] w.r.t their profit
   qsort (jobs, 5, sizeof (Jobs[0]), compareJob);

// Find the best job sequence
   bestJob (jobs, 5);

   return 0;

}
```

Teacher's Signature: _____