PRACTICAL: 09

AIM: Implementation of solution of Task Scheduling problem using Greedy method.

ALGORITHM: _____

1) Sort all jobs in decreasing order of profit.
2) Iterate on jobs in decreasing order of profit.For each job , do the following :
   a) Find a time slot i, such that slot is empty and i < deadline and i is greatest.Put the job in this slot and mark this slot filled.
   b) If no such i exists, then ignore the job.

CODE: _____

```
#include <algorithm>
#include <iostream>
using namespace std;

struct Job {
    char id;
    int dead;
    int profit;

}

bool comparison(Job a, Job b)
{
    return (a.profit > b.profit);
}

void printJobScheduling(Job arr[], int n)
{
    sort(arr, arr + n, comparison);

    int result[n];
    bool slot[n];

    for (int i = 0; i < n; i++)
        slot[i] = false;

    for (int i = 0; i < n; i++) {
        for (int j = min(n, arr[i].dead) - 1; j >= 0; j--) {
            if (slot[j] == false) {
```

```
                result[j] = i;
                slot[j] = true;
                break;
            }
        }
    }

    for (int i = 0; i < n; i++)
        if (slot[i])
            cout << arr[result[i]].id << " ";
}

int main()
{
    Job arr[] = { { 'a', 2, 100 },
                  { 'b', 1, 19 },
                  { 'c', 2, 27 },
                  { 'd', 1, 25 },
                  { 'e', 3, 15 } };
    int n = sizeof(arr) / sizeof(arr[0]);
    cout << "Following is maximum profit sequence of jobs "
         "\n";

    printJobScheduling(arr, n);
    return 0;
}
```

OUTPUT: _____

```
Following is maximum profit sequence of jobs
c a e
```

TIME COMPLEXITY:-

$O(n_2)$