EXPERIMENT NO:-09

Aim Of Experiment:- Implementation of solution of Task Scheduling problem using Greedy method.

ALGORITHM: _____

Job-Sequencing-With-Deadline (D, J, n, k)

1. D(0) := J(0) := 0
2. k := 1
3. Sort-Descending (p)
4. J(1) := 1
5. for i = 2 … n do
6. r := k
7. while D(J(r)) > D(i) and D(J(r)) ≠ r do
8. r := r – 1
9. if D(J(r)) ≤ D(i) and D(i) > r then
10. for l = k … r + 1 by -1 do
11. J(l + 1) := J(l)
12. J(r + 1) := i
13. k := k + 1

```
import java.util.*;
class Job {
  char id;
  int deadline, profit;
  public Job() {}
  public Job(char id, int deadline, int profit) {
    this.id = id;
    this.deadline = deadline;
    this.profit = profit;
  }
  void printJobScheduling(ArrayList<Job> arr, int t) {
    int n = arr.size();
    Collections.sort(arr, (a, b) -> b.profit - a.profit);
    boolean result[] = new boolean[t];
    char job[] = new char[t];
    for (int i = 0; i < n; i++) {
      for (int j = Math.min(t - 1, arr.get(i).deadline - 1); j >= 0; j--) {
        if (!result[j]) {
          result[j] = true;
```

```
                job[j] = arr.get(i).id;
                break;
            }
          }
        }
        for (char jb : job)
          System.out.print(jb + " ");
        System.out.println();
    }
    public static void main(String args[]) {
        ArrayList<Job> arr = new ArrayList<Job>();
        arr.add(new Job('a', 2, 100));
        arr.add(new Job('b', 1, 19));
        arr.add(new Job('c', 2, 27));
        arr.add(new Job('d', 1, 25));
        arr.add(new Job('e', 3, 15));
        System.out.println("Following is maximum profit sequence of jobs");
        Job job = new Job();
        job.printJobScheduling(arr, 3);
    }
}
```

OUTPUT:-

```
"C:\Program Files\Java\jdk-22\bin\java.exe" "-javaagent:C:\Users\Ankit raj\IntelliJ IDEA
Following is maximum profit sequence of jobs
c a e
```