**U. V. Patel College of Engineering**

**B. Tech CSE (CBA/MA/BDA) Sem-V**

**Subject: Algorithm Analysis and Design (2CSE502)**

**Practical 9**

**Implement Job Sequencing with deadlines using the Greedy technique.**

#include <stdio.h>

#define MAX 100

typedef struct Job {

char id[5];

int deadline;

int profit;

} Job;

void Demo(Job jobs[], int n);

int minValue(int x, int y) {

if(x < y) return x;

return y;

}

int main(void) {

int i, j;

Job jobs[5] = {

{"p1", 2, 100},

{"p2", 1, 19},

{"p3", 2, 27},

{"p4", 1, 25},

{"p5", 3, 15},

};

Job temp;

int n = 5;

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

for(j = 0; j < n - i; j++) {

if(jobs[j+1].profit > jobs[j].profit) {

temp = jobs[j+1];

jobs[j+1] = jobs[j];

jobs[j] = temp;

}

}

}

printf("%10s %10s %10s\n", "Job", "Deadline", "Profit");

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

printf("%10s %10i %10i\n", jobs[i].id, jobs[i].deadline, jobs[i].profit);

}

jobSequencingWithDeadline(jobs, n);

return 0;

}

void Demo(Job jobs[], int n) {

int i, j, k, maxprofit;

int timeslot[MAX];

int filledTimeSlot = 0;

int dmax = 0;

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

k = minValue(dmax, jobs[i - 1].deadline);

while(k >= 1) {

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

timeslot[k] = i-1;

filledTimeSlot++;

break;

}

k--;

}

if(filledTimeSlot == dmax) {

break;

}

}

printf("\nRequired Jobs: ");

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

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

if(i < dmax) {

printf(" --> ");

}

}

maxprofit = 0;

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

maxprofit += jobs[timeslot[i]].profit;

}

printf("\nMax Profit: %d\n", maxprofit);

}

