

CSE-2202 : Algorithm
Lab Assignment– 02

Name : Enamul Haque

ID : 201631046089

Batch 46th

Bangladesh University

Problem 1: Job sequence

```
1  #include <stdio.h>
2
3  #define MAX 100
4
5  typedef struct Job {
6      char id[5];
7      int deadline;
8      int profit;
9  } Job;
10
11 void jobSequencingWithDeadline(Job jobs[], int n);
12
13 int minValue(int x, int y) {
14     if(x < y) return x;
15     return y;
16 }
17
18 int main(void) {
19     //variables
20     int i, j;
21
22     //jobs with deadline and profit
23     Job jobs[5] = {
24         {"j1", 2, 60},
25         {"j2", 1, 100},
26         {"j3", 3, 20},
27         {"j4", 2, 40},
28         {"j5", 1, 20},
29     };
30
31     //temp
32     Job temp;
```

```

Start here X binary_searchUntitled1.c X selection_sortUntitled2.c X mergesort_descendingUntitled3.c X Untitled4.cpp X jobsequenUntitled5.c
31 //temp
32 Job temp;
33
34 //number of jobs
35 int n = 5;
36
37 //sort the jobs profit wise in descending order
38 for(i = 1; i < n; i++) {
39     for(j = 0; j < n - i; j++) {
40         if(jobs[j+1].profit > jobs[j].profit) {
41             temp = jobs[j+1];
42             jobs[j+1] = jobs[j];
43             jobs[j] = temp;
44         }
45     }
46 }
47
48 printf("%10s %10s %10s\n", "Job", "Deadline", "Profit");
49 for(i = 0; i < n; i++) {
50     printf("%10s %10i %10i\n", jobs[i].id, jobs[i].deadline, jobs[i].profit);
51 }
52
53 jobSequencingWithDeadline(jobs, n);
54
55 return 0;
56 }
57
58 void jobSequencingWithDeadline(Job jobs[], int n) {
59     //variables
60     int i, j, k, maxprofit;
61
62     //free time slots
63     int timeslot[MAX];

```

```
61
62 //free time slots
63 int timeslot[MAX];
64
65 //filled time slots
66 int filledTimeSlot = 0;
67
68 //find max deadline value
69 int dmax = 0;
70 for(i = 0; i < n; i++) {
71     if(jobs[i].deadline > dmax) {
72         dmax = jobs[i].deadline;
73     }
74 }
75
76 //free time slots initially set to -1 [-1 denotes EMPTY]
77 for(i = 1; i <= dmax; i++) {
78     timeslot[i] = -1;
79 }
80
81 printf("dmax: %d\n", dmax);
82
83 for(i = 1; i <= n; i++) {
84     k = minValue(dmax, jobs[i - 1].deadline);
85     while(k >= 1) {
86         if(timeslot[k] == -1) {
87             timeslot[k] = i-1;
88             filledTimeSlot++;
89             break;
90         }
91         k--;
92     }
93 }
```

```
Start here X binary_searchUntitled1.c X selection_sortUntitled2.c X mergesort_descendingUntitled3.c X Untitled4.cpp X jobsequenUntitled5.c X
85 while(k >= 1) {
86     if(timeslot[k] == -1) {
87         timeslot[k] = i-1;
88         filledTimeSlot++;
89         break;
90     }
91     k--;
92 }
93
94 //if all time slots are filled then stop
95 if(filledTimeSlot == dmax) {
96     break;
97 }
98 }
99
100 //required jobs
101 printf("\nRequired Jobs: ");
102 for(i = 1; i <= dmax; i++) {
103     printf("%s", jobs[timeslot[i]].id);
104 }
105 if(i < dmax) {
106     printf(" --> ");
107 }
108 }
109
110 //required profit
111 maxprofit = 0;
112 for(i = 1; i <= dmax; i++) {
113     maxprofit += jobs[timeslot[i]].profit;
114 }
115 printf("\nMax Profit: %d\n", maxprofit);
116 }
117 }
```

```
Edit View Search Project Build Debug Fortran wxSmith tools tools+ Plugins DoxyBlocks Settings Help
C:\Users\Enamul\Documents\jobsequenUntitled5.exe
Job Deadline Profit
j2 1 100
j1 2 60
j4 2 40
j3 3 20
j5 1 20
dmax: 3

Required Jobs: j2 --> j1 --> j3
Max Profit: 180

Process returned 0 (0x0) execution time : 0.038 s
Press any key to continue.
```

Problem 2: Knapsack problem

```

1
2  #include <stdio.h>
3
4  int max(int a, int b) { return (a > b) ? a : b; }
5
6  /
7  int knapSack(int W, int wt[], int val[], int n)
8  {
9      // Base Case
10     if (n == 0 || W == 0)
11         return 0;
12
13
14     if (wt[n - 1] > W)
15         return knapSack(W, wt, val, n - 1);
16
17     return max(
18         val[n - 1]
19         + knapSack(W - wt[n - 1],
20             wt, val, n - 1),
21         knapSack(W, wt, val, n - 1));
22 }
23
24 // Driver program to test above function
25 int main()
26 {
27     int val[] = { 60, 100, 120 };
28     int wt[] = { 10, 20, 30 };
29     int W = 50;
30     int n = sizeof(val) / sizeof(val[0]);
31     printf("%d", knapSack(W, wt, val, n));
32     return 0;
33 }
34

```

Symbolic Execution

C:\Users\Enamu\Documents\jobsecquen\Untitled5.exe

220
Process returned 0 (0x0) execution time : 0.037 s
Press any key to continue.

