

C 01knapsack.c X C prims.c C kruskals.c

C 01knapsack.c

```
1  /*0/1 Knapsack problem by Dynamic programming technique*/
2  #include<stdio.h>
3  #include<conio.h>
4  int max(int,int);
5  void objected_selected();
6  int m,i,j,n,p[10],w[10],v[10][10], x[10],op_soln;
7  int knapsack();
8  void main()
9  {
10     printf("Enter the number of objects\n");
11     scanf("%d", &n);
12     printf("Enter the weights of n objects\n");
13     for(i=1;i<=n;i++)
14         scanf("%d", &w[i]);
15     printf("Enter the profits of n objects\n");
16     for(i=1;i<=n;i++)
17     {
18         scanf("%d", &p[i]);
19     }
20     printf("Enter the capacity of Knapsack\n");
21     scanf("%d", &m);
22     op_soln=knapsack(n,w,m,v,p);
23     printf("The table for this problem is\n");
24     for(i=0;i<=n;i++)
25     {
26         for(j=0;j<=m;j++)
27         {
28             printf("%d\t", v[i][j]);
29         }
30         printf("\n");
31     }
32     printf("Optimal Solution=%d\n",op_soln);
33     objected_selected();
34     getch();
35 }
36
37 int max(int a, int b)
38 {
```

C 01knapsack.c X C prims.c C kruskals.c

C 01knapsack.c

```
36
37 int max(int a, int b)
38 {
39     return(a>b?a:b);
40 }
41
42 int knapsack()
43 {
44     int i,j;
45     for(i=0;i<=n;i++)
46     {
47         for(j=0;j<=m;j++)
48         {
49             if(i==0||j==0)
50             {
51                 v[i][j]=0;
52             }
53             else
54             {
55                 if(w[i]>j)
56                 {
57                     v[i][j]=v[i-1][j];
58                 }
59                 else
60                 {
61                     v[i][j]=max(v[i-1][j],v[i-1][j-w[i]]+p[i]);
62                 }
63             }
64         }
65     }
66     return v[n][m];
67 }
68 void objected_selected()
69 {
70     i=n;
71     j=m;
72     while(i!=0 && j!=0)
73     {
```

C 01knapsack.c X C prims.c C kruskals.c

C 01knapsack.c

```
63     }
64 }
65 }
66 return v[n][m];
67 }
68 void objected_selected()
69 {
70     i=n;
71     j=m;
72     while(i!=0 && j!=0)
73     {
74         if(v[i][j]!=v[i-1][j])
75         {
76             x[i]=1;
77             j=j-w[i];
78         }
79         i--;
80     }
81     printf("Objects Selected are:\n");
82     for(i=1;i<=n;i++)
83     {
84         if( x[i]==1)
85             printf("%d ",i);
86     }
87 }
```



C 01knapsack.c X C prims.c C kruskals.c

```
C 01knapsack.c
63     }
64     }
65 }
66     return v[n][m];
67 }
68 void object_selected()
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

Total Cost= 15  
PS C:\Users\muska\OneDrive\Desktop\C programs> gcc 01knapsack.c  
PS C:\Users\muska\OneDrive\Desktop\C programs> .\a.exe  
Enter the number of objects  
4  
Enter the weights of n objects  
2 1 3 2  
Enter the profits of n objects  
12 10 20 15  
Enter the capacity of Knapsack  
5  
The table for this problem is  

0	0	0	0	0	0
0	0	12	12	12	12
0	10	12	22	22	22
0	10	12	22	30	32
0	10	15	25	30	37

  
Optimal Solution=37  
Objects Selected are:  
1 2 4  
PS C:\Users\muska\OneDrive\Desktop\C programs> |