

```
1  #include<stdio.h>
2  #include<stdlib.h>
3  void knapsack();
4  int max(int,int);
5  int i,j,n,m,p[10],w[10],v[10][10];
6  int main()
7  {
8      printf("\nenter the no. of items:\t");
9      scanf("%d",&n);
10     printf("\nenter the weight of the each item:\n");
11     for(i=1;i<=n;i++)
12     {
13         scanf("%d",&w[i]);
14     }
15     printf("\nenter the profit of each item:\n");
16     for(i=1;i<=n;i++)
17     {
18         scanf("%d",&p[i]);
19     }
20     printf("\nenter the knapsack's capacity:\t");
21     scanf("%d",&m);
22     knapsack();
23 }
24 void knapsack()
25 {
26     int x[10];
27     for(i=0;i<=n;i++)
28     {
29         for(j=0;j<=m;j++)
30         {
31             if(i==0||j==0)
32             {
33                 v[i][j]=0;
34             }
35             else if(j-w[i]<0)
36             {
37                 v[i][j]=v[i-1][j];
38             }
39             else
40             {
41                 v[i][j]=max(v[i-1][j],v[i-1][j-w[i]]+p[i]);
42             }
43         }
44     }
```

```
45 printf("\nthe output is:\n");
46 for(i=0;i<=n;i++)
47 {
48     for(j=0;j<=m;j++)
49     {
50         printf("%d\t",v[i][j]);
51     }
52     printf("\n\n");
53 }
54 printf("\nthe optimal solution is %d",v[n][m]);
55 printf("\nthe solution vector is:\n");
56 for(i=n;i>=1;i--)
57 {
58     if(v[i][m]!=v[i-1][m])
59     {
60         x[i]=1;
61         m=m-w[i];
62     }
63     else
64     {
65         x[i]=0;
66     }
67 }
68 for(i=1;i<=n;i++)
69 {
70     printf("%d\t",x[i]);
71 }
72 }
73
74 int max(int x,int y)
75 {
76     if(x>y)
77     {
78         return x;
79     }
80     else
81     {
82         return y;
83     }
84 }
85
```

```
> clang-7 -pthread -lm -o main main.c  
> ./main
```

enter the no. of items: 4

enter the weight of the each item:
4 3 6 4

enter the profit of each item:
14 12 18 15

enter the knapsack's capacity: 5

the output is:

0	0	0	0	0	0
---	---	---	---	---	---

0	0	0	0	14	14
---	---	---	---	----	----

0	0	0	12	14	14
---	---	---	----	----	----

0	0	0	12	14	14
---	---	---	----	----	----

0	0	0	12	15	15
---	---	---	----	----	----

the optimal solution is 15

the solution vector is:

0	0	0	1
---	---	---	---