main.c

```c
#include<stdio.h>
#include<stdlib.h>
void knapsack();
int max(int,int);
int i,j,n,m,p[10],w[10],v[10][10];
int main()
{
printf("\nenter the no. of items:\t");
scanf("%d",&n);
printf("\nenter the weight of the each item:\n");
for(i=1;i<=n;i++)
{
scanf("%d",&w[i]);
}
printf("\nenter the profit of each item:\n");
for(i=1;i<=n;i++)
{
scanf("%d",&p[i]);
}
printf("\nenter the knapsack's capacity:\t");
scanf("%d",&m);
knapsack();
}
void knapsack()
{
int x[10];
for(i=0;i<=n;i++)
{
for(j=0;j<=m;j++)
{
if(i==0||j==0)
{
v[i][j]=0;
}
```

```
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++)
```

main.c

```
69    {
70    printf("%d/t",x[i]);
71    }
72    }
73    int max(int x,int y)
74    {
75    if(x>y)
76    {
77    return x;
78    }
79    else
80    {
81    return y;
82    }
83    }
```

```
> 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   > 
```