File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help

floyds.c X knapsack.c X

```
#include<stdio.h>
#include<conio.h>
void knapsack();
int max(int,int);
int i,j,n,m,p[10],w[10],v[10][10];
void main()
{
printf("\n enter the no. of items:\t");
scanf("%d",&n);
printf("\n enter the weight of the each item:\n");
for(i=1;i<=n;i++)
{
scanf("%d",&w[i]);
}
printf("\n enter the profit of each item:\n");
for(i=1;i<=n;i++)
{
scanf("%d",&p[i]);
}
printf("\n enter 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;
}
else if(j-w[i]<0)
{
```

File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help

floyds.c X knapsack.c X

```
37
38      v[i][j]=v[i-1][j];
39      }
40      else
41      {
42      v[i][j]=max(v[i-1][j],v[i-1][j-w[i]]+p[i]);
43      }
44      }
45      }
46      printf("\n the output is:\n");
47      for(i=0;i<=n;i++)
48      {
49      for(j=0;j<=m;j++)
50      {
51      printf("%d\t",v[i][j]);
52      }
53      printf("\n\n");
54      }
55      printf("\n the optimal solution is %d",v[n][m]);
56      printf("\n the solution vector is:\n");
57      for(i=n;i>=1;i--)
58      {
59      if(v[i][m]!=v[i-1][m])
60      {
61      x[i]=1;
62      m=m-w[i];
63      }
64      else
65      {
66      x[i]=0;
67      }
68      }
69      for(i=1;i<=n;i++)
70      {
71      printf("%d\t",x[i]);
72      }
```

File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help

floyds.c × knapsack.c ×

```
51      printf("%d\t",v[i][j]);
52      }
53      printf("\n\n");
54      }
55      printf("\n the optimal solution is %d",v[n][m]);
56      printf("\n the solution vector is:\n");
57      for(i=n;i>=1;i--)
58      {
59      if(v[i][m]!=v[i-1][m])
60      {
61      x[i]=1;
62      m=m-w[i];
63      }
64      else
65      {
66      x[i]=0;
67      }
68      }
69      for(i=1;i<=n;i++)
70      {
71      printf("%d\t",x[i]);
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
```

C:\Users\CMRS\Documents\knapsack.exe

```
 enter the weight of the each item:
2 1 3 2

 enter the profit of each item:
12 10 20 15

 enter the knapsack's capacity: 5

 the output 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


 the optimal solution is 37
 the solution vector is:
1       1       0       1
Process returned 4 (0x4)   execution time : 28.834 s
Press any key to continue.
```