Knapsack_DynamicProgramming.c

Saved

```c
#include<stdio.h>
int max(int,int);
int m,i,j,n,p[10],w[10],v[10][10],x[10],op_soln;
int knapsack( );
void objects_selected();
void main()
{
    printf("Enter the number of Objects: ");
    scanf("%d", &n);
    printf("Enter the Weights of N Objects:");
    for(i=1;i<=n;i++)
        scanf("%d", &w[i]);
    printf("Enter the Profits of N Objects:");
    for(i=1;i<=n;i++)
        scanf("%d", &p[i]);
    printf("Enter the Capacity of Knapsack:");
    scanf("%d", &m);
    op_soln=knapsack(n,w,m,v,p);
    printf("\nThe output is\n");
    for(i=0;i<=n;i++)
    {
        for(j=0;j<=m;j++)
        {
            printf("%d\t", v[i][j]);
        }
        printf("\n");
    }
    printf("\nOptimal Solution = %d",op_soln);
    objects_selected();
}
int max(int a, int b)
{
    return(a>b?a:b);
}
int knapsack()
{
    int i,j;
    for(i=0;i<=n;i++)
    {
        for(j=0;j<=m;j++)
        {
            if(i==0||j==0)
                v[i][j]=0;
            else if(w[i]>j)
                v[i][j]=v[i-1][j];
            else
                v[i][j]=max(v[i-1][j],v[i-1][j-w[i]]+p[i]);
        }
    }
    return v[n][m];
}
void objects_selected()
{
    i=n;
    j=m;
    while(i!=0 && j!=0)
    {
        if(v[i][j]!=v[i-1][j])
        {
            x[i]=1;
            j=j-w[i];
        }
        i--;
    }
    printf("\nObjects Selected = ");
    for(i=1;i<=n;i++)
    {
        if( x[i]==1)
        printf("%d,",i);
    }
}
```

```
×    Terminal

Enter the number of Objects: 5
Enter the Weights of N Objects:3 2 1 4 5
Enter the Profits of N Objects:25 20 15 40 50
Enter the Capacity of Knapsack:7

The output is
0 0 0 0 0 0 0 0
0 0 0 25 25 25 25 25
0 0 20 25 25 45 45 45
0 15 20 35 40 45 60 60
0 15 20 35 40 55 60 75
0 15 20 35 40 55 65 75

Optimal Solution = 75
Objects Selected = 2,3,4,
Process finished with exit code 5.
```