Knapsack Problem - C Program

```
#include <stdio.h>

void displayinfo(int m, int n, int w[], int p[]);
void knapsack(int m, int n, int w[], int p[], int v[][10]);
void optimal(int m, int n, int w[], int v[][10]);
int max(int i, int j);

int main()
{
    int v[10][10], w[10], p[10], i, j;

    printf("************* KNAPSACK PROBLEM ***********\n");
    printf("Enter the total number of items: ");
    int n;
    scanf("%d", &n);

    printf("Enter the weight of each item: \n");
    for (i = 1; i <= n; i++)
    {
        scanf("%d", &w[i]);
    }

    printf("Enter the profit of each item: \n");
    for (i = 1; i <= n; i++)
    {
        scanf("%d", &p[i]);
    }

    printf("Enter the knapsack capacity: ");
    int m;
    scanf("%d", &m);

    displayinfo(m, n, w, p);
    knapsack(m, n, w, p, v);

    printf("The contents of the knapsack table are:\n");
    for (i = 0; i <= n; i++)
    {
        for (j = 0; j <= m; j++)
        {
            printf("%d ", v[i][j]);
        }
        printf("\n");
    }

    optimal(m, n, w, v);
}

void displayinfo(int m, int n, int w[], int p[])
{
```

```c
    printf("Entered information about knapsack problem are:\n");
    printf("ITEM\tWEIGHT\tPROFIT\n");

    for (int i = 1; i <= n; i++)
    {
        printf("%d\t%d\t%d\n", i, w[i], p[i]);
    }

    printf("Capacity = %d\n", m);
}

void knapsack(int m, int n, int w[], int p[], int v[][10])
{
    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 (j < w[i])
            {
                v[i][j] = v[i-1][j];
            }
            else
            {
                v[i][j] = max(v[i-1][j], v[i-1][j-w[i]] + p[i]);
            }
        }
    }
}

int max(int i, int j)
{
    if (i > j)
    {
        return i;
    }
    else
    {
        return j;
    }
}

void optimal(int m, int n, int w[], int v[][10])
{
    int i = n, j = m, item = 0, x[10] = {0};
```

Knapsack Problem - C Program

```c
    printf("Optimal solution is: %d\n", v[n][m]);
    printf("Selected items are: ");

    while (i != 0 && j != 0)
    {
        if (v[i][j] != v[i-1][j])
        {
            x[i] = 1;
            j = j - w[i];
        }
        i = i - 1;
    }

    for (i = 1; i <= n; i++)
    {
        if (x[i] == 1)
        {
            printf("%d ", i);
            item = 1;
        }
    }

    if (item == 0)
    {
        printf("NIL\t Sorry! No item can be placed in Knapsack\n");
    }

    printf("\n********* *********************** *************\n");
}
```