0/1 Knapsack Problem

```
#include <stdio.h>
#include <conio.h>
int n, m, w[10], p[10], v[10][10];
int max(int a, int b)
{
   if (a > b)
      return a;
   else
      return b;
}

int knapsack()
{
   for (int i = 0; i <= n; i++)
   {
      for (int j = 0; j <= m; j++)
      {
         if (i == 0 && j == 0)
         {
            v[i][j] = 0;
         }
         else if (w[i - 1] > j)
         {
            v[i][j] = v[i - 1][j];
         }
         else
         {
            v[i][j] = max(v[i - 1][j], v[i - 1][j - w[i - 1]] + p[i - 1]);
         }
      }
   }
   return v[n][m];
}
void object_selected()
{
   int i = n, j = m;
   int x[10];
   for (int k = 1; k <= n; k++)
   {
      x[k] = 0;
   }
```

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

    for (i = 1; i <= n; i++)
    {
        if (x[i] == 1)
        {
            printf("Object %d selected \n", i);
        }
    }
}
void main()
{
    printf("Enter the number of objects\n");
    scanf("%d", &n);
    for (int i = 0; i < n; i++)
    {
        printf("Enter the weight of object  %d  ", i + 1);
        scanf("%d", &w[i]);
    }
    for (int i = 0; i < n; i++)
    {
        printf("Enter the profit of object %d  ", i + 1);
        scanf("%d", &p[i]);
    }
    printf("Enter the maximum limit\n");
    scanf("%d", &m);
    int profit = knapsack();
    object_selected();

    printf("Maximum profit %d\n", profit);
}
```

```
Enter the number of objects
4
Enter the weight of object  1  2
Enter the weight of object  2  1
Enter the weight of object  3  3
Enter the weight of object  4  2
Enter the profit of object 1  12
Enter the profit of object 2  10
Enter the profit of object 3  20
Enter the profit of object 4  15
Enter the maximum limit
5
Object 1 selected
Object 2 selected
Object 4 selected
Maximum profit 37
```