0-1 knapsheack

#include <bits/stdc++.h>

using namespace std;

int max(int a, int b)

{

return (a > b) ? a : b;

}

void knapSack(int W, int wt[], int val[], int n)

{

int i, w;

int K[n + 1][W + 1];

for (i = 0; i <= n; i++) {

for (w = 0; w <= W; w++) {

if (i == 0 || w == 0)

K[i][w] = 0;

else if (wt[i - 1] <= w)

K[i][w] = max(val[i - 1] +

K[i - 1][w - wt[i - 1]], K[i - 1][w]);

else

K[i][w] = K[i - 1][w];

}

}

int r = K[n][W];

cout<<"Max Benefit : "<<r<<endl;

w = W;

cout<<"The items are count :";

for (i = n; i > 0 && r > 0; i--) {

if (r == K[i - 1][w])

continue;

else {

cout<<wt[i - 1]<<" ";

r = r - val[i - 1];

w = w - wt[i - 1];

}

}

}

int main()

{

int n,w,i;

cout<<"Enter your total items : ";

cin>>n;

int val[n];

int wt[n];

cout<<"Enter your and value "<<endl;

for(i=0;i<n;i++)

{

cin>>wt[i]>>val[i];

}

cout<<"Enter your max weight : ";

cin>>w;

knapSack(w, wt, val, n);

return 0;

}

2:

#include <bits/stdc++.h>

using namespace std;

int max(int a, int b)

{

return (a > b) ? a : b;

}

void knapSack(int W, int wt[], int v[], int n)

{

int i, w;

int K[n + 1][W + 1];

for (i = 0; i <= n; i++) {

for (w = 0; w <= W; w++) {

if (i == 0 || w == 0)

K[i][w] = 0;

else if (wt[i - 1] <= w)

K[i][w] = max(v[i - 1] +

K[i - 1][w - wt[i - 1]], K[i - 1][w]);

else

K[i][w] = K[i - 1][w];

}

}

int r = K[n][W];

cout<<r<<endl;

w = W;

for (i = n; i > 0 && r > 0; i--)

{

if (r == K[i - 1][w])

continue;

else {

cout<< wt[i - 1]<<" ";

r = r - v[i - 1];

w = w - wt[i - 1];

}

}

}

int main()

{

int wt[] = { 0,1,2,3,4,4,6 };

int v[] = {0,3,4,5,5,6,9};

int W = 10;

int n = sizeof(v) / sizeof(v[0]);

knapSack(W, wt, v, n);

return 0;

}