**Implement 0/1 Knapsack problem using dynamic programming.**

#include<stdio.h>

#include<conio.h>

int i,j,n,c,w[10],p[10],v[10][10];

void knapsack(int n,int w[10],int p[10],int c)

{

int max(int,int);

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

{

for(j=0;j<=c;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]));

}

}

printf("\n\n Maximum Profit is : %d ",v[n][c]);

printf("\n\n\n Table : \n\n");

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

{

for(j=0;j<=c;j++)

{

printf("\t%d",v[i][j]);

}

printf("\n");

}

}

int max(int a,int b)

{

return ((a>b)?a:b);

}

void main()

{

clrscr();

printf("\n Enter the no. of objects : ");

scanf("%d",&n);

printf("\n Enter the weights : ");

for(i=1;i<=n;i++)

{

scanf("%d",&w[i]);

}

printf("\n Enter the Profits : ");

for(i=1;i<=n;i++)

{

scanf("%d",&p[i]);

}

printf("\n Enter the capacity : ");

scanf("%d",&c);

knapsack(n,w,p,c);

getch();

}

Output:

