## **DAA LAB**

## LAB 3: Implement 0/1 Knapsack Problem using Dynamic Programming.

PROGRAM:

**#include<stdio.h>**

**int w[10],p[10],v[10][10],n,i,j,cap,x[10]={0};**

**int max(int i,int j)**

**{**

**return ((i>j)?i:j);**

**}**

**int knap(int i,int j){**

**int value;**

**if(v[i][j]<0){**

**if(j<w[i])**

**value=knap(i-1,j);**

**else**

**value=max(knap(i-1,j),p[i]+knap(i-1,j-w[i]));**

**v[i][j]=value;**

**}**

**return(v[i][j]);**

**}**

**int main(){**

**int profit,count=0;**

**printf("\nEnter the number of objects ");**

**scanf("%d",&n);**

**printf("Enter the profit and weights of the elements \n ");**

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

**printf("\nEnter profit and weight For object no %d :",i);**

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

**}**

**printf("\nEnter the capacity ");**

**scanf("%d",&cap);**

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

**for(j=0;j<=cap;j++)**

**if((i==0)||(j==0))**

**v[i][j]=0;**

**else**

**v[i][j]=-1;**

**profit=knap(n,cap);**

**i=n;**

**j=cap;**

**while(j!=0&&i!=0){**

**if(v[i][j]!=v[i-1][j]){**

**x[i]=1;**

**j=j-w[i];**

**i--;**

**}**

**else**

**i--;**

**}**

**printf("object included are \n ");**

**printf("Sl.no\tweight\tprofit\n");**

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

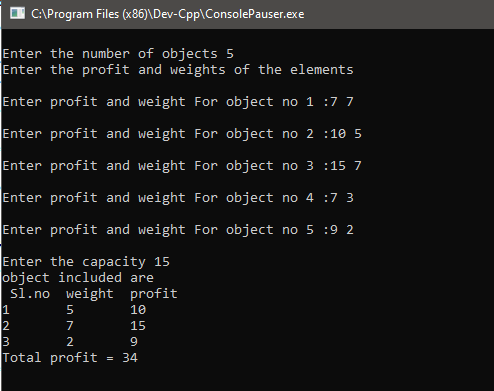
**if(x[i])**

**printf("%d\t%d\t%d\n",++count,w[i],p[i]);**

**printf("Total profit = %d\n",profit);**

**}**

OUTPUT:

****