**EXPERIMENT – 7**

**AIM**: **Implementation of a knapsack problem using greedy algorithm.**

**Input:**

#include<stdio.h>

#include<conio.h>

void knapsack(int n,float m,float w[ ],float p[ ]);

void main( )

{

int i,j,n;

float p[15],w[15],c[15],temp,m;

clrscr( );

printf("\nEnter number of objects:");

scanf("%d",&n);

printf("\nEnter weights:");

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

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

flushall( );

printf("\nEnter profits:");

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

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

printf("\nEnter knapsack size:");

scanf("%f",&m);

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

c[i]=p[i]/w[i];

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

{

for(j=0;j<n-1;j++)

{

if(c[j] < c[j+1])

{

temp=c[j];

c[j]=c[j+1];

c[j+1]=temp;

temp=w[j];

w[j]=w[j+1];

w[j+1]=temp;

temp=p[j];

p[j]=p[j+1];

p[j+1]=temp;

}

}

}

printf("\n The items are arranged as ...\n");

printf("\n\nItems\tweights \tProfits");

{

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

printf("\nx[%d]\t%.0f\t\t%.0f",i,w[i],p[i]);

}

knapsack(n,m,w,p);

getch( );

}

void knapsack(int n,float m,float w[ ],float p[ ])

{

float x[15],U,profit=0.0,weight=0.0;

int i;

U=m;

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

x[i]=0.0;

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

{

if(w[i]>U)

break;

x[i]=1.0;

U=U-w[i];

}

if(i<n)

x[i]=U/w[i];//take fractional part of item to fulfil the size

printf("\nThe solution vector is:");

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

printf("\n%d\t\%.2f",i,x[i]);

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

{

w[i]=w[i]\*x[i];

p[i]=p[i]\*x[i];

}

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

{

profit=profit+p[i];//computing total profit & wt.

weight=weight+w[i];

}

printf("\nMaximum profit is:");

printf("\n\t\t%.2f",profit);

printf("\nMaximum weight is:");

printf("\n\t\t%.2f",weight);

}