7. Design and implement C/C++ Program to solve discrete Knapsack and continuous Knapsack problems using greedy approximation method.

#include<stdio.h>

int main( )

{

float weight[50], profit[50], ratio[50], Totalvalue, temp, capacity, amount;

int n,i,j;

printf("Enter the number of items :");

scanf("%d",&n);

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

{

printf("Enter Weight and Profit for item[%d] :\n",i);

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

}

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

scanf("%f",&capacity);

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

ratio[i]=profit[i]/weight[i];

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

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

if (ratio[i] < ratio[j])

{

temp = ratio[j];

ratio[j] = ratio[i];

ratio[i] = temp;

temp = weight[j];

weight[j] = weight[i];

weight[i] = temp;

temp = profit[j];

profit[j] = profit[i];

profit[i] = temp;

}

printf("Knapsack problems using Greedy Algorithm:\n");

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

{

if (weight[i] > capacity)

break;

else

{

Totalvalue = Totalvalue + profit[i];

capacity = capacity - weight[i];

}

}

if (i < n)

Totalvalue = Totalvalue + (ratio[i]\*capacity);

printf("\nThe maximum value is :%f\n",Totalvalue);

return 0;

}

OUTPUT:

Enter the number of items :4

Enter Weight and Profit for item[0] :

2 12

Enter Weight and Profit for item[1] :

1 10

Enter Weight and Profit for item[2] :

3 20

Enter Weight and Profit for item[3] :

2 15

Enter the capacity of knapsack :

5

Knapsack problems using Greedy Algorithm:

The maximum value is : 38.333332