**Fractional Knapsack Code:**

#include <stdio.h>

int m,n;

float x[10];

struct knaps

{

int w,p,obj\_no;

float pw;

};

struct knaps arr[10];

void sortks()

{

int i, j;

struct knaps temp;

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

{

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

{

if (arr[j].pw < arr[j + 1].pw)

{

temp = arr[j];

arr[j] = arr[j + 1];

arr[j + 1] = temp;

}

}

}

}

int greedyknapsack()

{

int i,rem\_cap,profit=0;

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

x[i]=0;

rem\_cap=m;

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

{

if(arr[i].w>rem\_cap)

break;

x[arr[i].obj\_no-1]=1;

profit=profit+arr[i].p;

rem\_cap=rem\_cap-arr[i].w;

}

if(i<=n)

{

x[arr[i].obj\_no-1]=(float)(rem\_cap)/(float)(arr[i].w);

profit=profit+x[i]\*(arr[i].p);

}

return profit;

}

int main()

{

int i,value;

printf("Enter the size of knapsack: ");

scanf("%d",&m);

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

scanf("%d",&n);

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

{

printf("Enter weight of object %d: ",i+1);

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

}

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

{

printf("Enter profit of object %d: ",i+1);

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

}

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

arr[i].obj\_no=i+1;

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

{

arr[i].pw = (float)(arr[i].p)/(float)(arr[i].w);

}

printf("\nWeight\t\tProfit\t\tRatio\n");

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

{

printf("%d\t\t%d\t\t%f\n",arr[i].w,arr[i].p,arr[i].pw);

}

sortks();

printf("\nAfter sorting\n");

printf("Weight\t\tProfit\t\tRatio\n");

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

{

printf("%d\t\t%d\t\t%f\n",arr[i].w,arr[i].p,arr[i].pw);

}

value=greedyknapsack();

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

printf("%0.2f\t",x[i]);

printf("Profit= %d",value);

return 0;

}

**Output:**

Enter the size of knapsack:

60

Enter the number of objects:

5

Enter weight of object 1:

5

Enter weight of object 2:

10

Enter weight of object 3:

15

Enter weight of object 4:

20

Enter weight of object 5:

25

Enter profit of object 1:

30

Enter profit of object 2:

40

Enter profit of object 3:

45

Enter profit of object 4:

77

Enter profit of object 5:

90

Weight Profit Ratio

5 30 6.000000

10 40 4.000000

15 45 3.000000

20 77 3.850000

25 90 3.600000

After sorting

Weight Profit Ratio

5 30 6.000000

10 40 4.000000

20 77 3.850000

25 90 3.600000

15 45 3.000000

1.00 1.00 0.00 1.00 1.00 Profit= 282

\*\* Process exited - Return Code: 0 \*\*