

**Program 7:**

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

**Code:**

```
#include<stdio.h>

int n,m,p[10],w[10];

void greedy_knapsack()

{

float max, profit=0;

int k=0,i,j;

printf("item included is: ");

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

{

max=0;

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

{

if(((float)p[j])/w[j]>max)

{

k=j;

max=((float)p[j])/w[j];

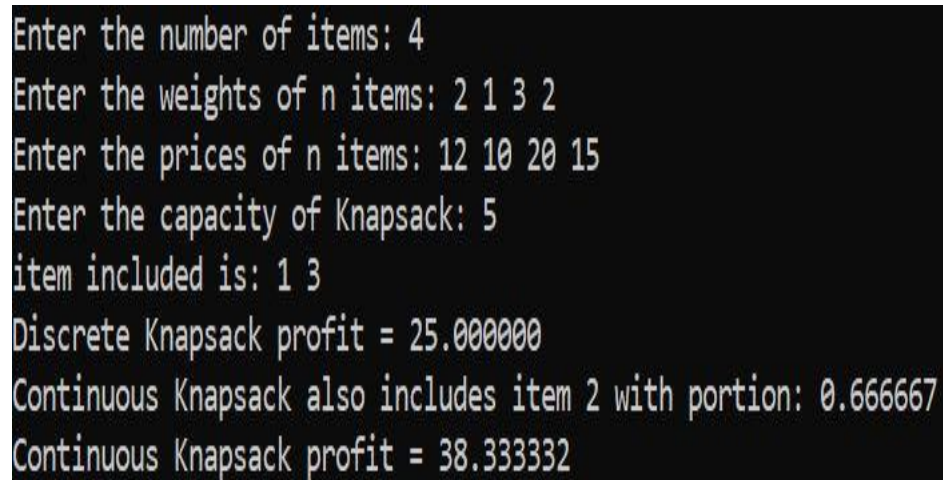
}

}

if(w[k]<=m)
```

```
{  
printf("%d ",k);  
  
m=m-w[k];  
profit=profit+p[k];  
p[k]=0;  
}  
  
else  
  
break;  
  
}  
  
printf("\nDiscrete Knapsack profit = %f\n",profit);  
  
printf("Continuous Knapsack also includes item %d with portion: %f\n",k,((float)m)/w[k]);  
profit=profit+((float)m)/w[k]*p[k];  
printf("Continuous Knapsack profit = %f\n",profit);  
  
}  
  
int main()  
  
{  
  
int i;  
  
printf("Enter the number of items: ");  
  
scanf("%d", &n);  
  
printf("Enter the weights of n items: ");  
  
for(i=0;i<n;i++)  
  
scanf("%d",&w[i]);  
  
printf("Enter the prices of n items: ");  
  
for(i=0;i<n;i++)
```

```
scanf("%d",&p[i]);  
  
printf("Enter the capacity of Knapsack: ");  
  
scanf("%d",&m);  
  
greedy_knapsack();  
  
}
```

**Output:**A screenshot of a terminal window with a black background and light green text. The output shows the program's execution flow: it prompts for the number of items (4), weights (2 1 3 2), prices (12 10 20 15), and capacity (5). It then displays the results for both discrete and continuous knapsack algorithms.

```
Enter the number of items: 4  
Enter the weights of n items: 2 1 3 2  
Enter the prices of n items: 12 10 20 15  
Enter the capacity of Knapsack: 5  
item included is: 1 3  
Discrete Knapsack profit = 25.000000  
Continuous Knapsack also includes item 2 with portion: 0.666667  
Continuous Knapsack profit = 38.333332
```