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: