P6) Aim: Implement in Java, the knapsack problem using greedy method.

```
import java.util.Scanner;
public class Knapsack{
  public static void main(String[] args){
int i, j, n;
float temp, m;
float p[] = new float[15]; //value/profit
float w[] = new float[15]; //weight
float c[] = new float[15]; //ratio
System.out.println("Enter number of objects:");
Scanner s = new Scanner(System.in);
n=s.nextInt();
System.out.println("Enter the weights:");
for(i=1; i<=n; i++)
    w[i] = s.nextFloat();
System.out.println("Enter the profits:");
for(i=1; i<=n; i++)
    p[i] = s.nextFloat();
System.out.println("Enter the knapsack capacity");
m = s.nextInt();
for(i=1; i<=n; i++)
    c[i] = p[i] / w[i];
```

```
for (i=1;i<=n;i++)
{
   for(j=1;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;
     }
   }
}
System.out.println("The items to be arranged:");
System.out.println("\n\tItem\tWeight\tProfit");
for(i=1;i<=n;i++)
System.out.println("\n"+x[i]+"\t"+w[i]+"\t"+p[i]);
 Knapsack(n,m,w,p);
}
```

```
public static void knapsack(int n, float m,
        float w[], float p[])
{
    float x[] = new float[15];
    float u, profit=0, weight=0;
    int i;
    u=m;
    for(i=1; i<=n; i++)
    x[i]=0;
    for(i=1; i<=n; i++)
    {
      if(w[i]>u)
          break;
      x[i]=1
      u=u-w[i];
    }
if(i<=n)
  x[i]=u/w[i];
  System.out.println("\n solution vector: ");
  for(i=1; i<=n; i++)
   { System.out.print(x[i]+"\t");
     w[i]=w[i]*x[i];
     p[i]=p[i]*x[i];
   }
```

```
for (i=1; i<=n; i++)
{
    profit = profit + p[i];
    weight = weight + w[i];
}
System.out.println("\n maximum profit :");
System.out.println(" it " + profit);
System.out.println("\n Total weight :" + weight);
}
}
```

Output:

```
Enter the number of Elements:
3
Enter the weights
16
14
10
enter the profit:
10
20
30
enter the knapsack capacity:
20
the items to be arranged:
```

Items	weight	Profit
x[1]	10.0	30.0
x[2]	14.0	20.0
x[3]	16.0	10.0

The solution vector is:
1.0
0.7142857
0.0
The maximum Profit is:
44.285713
The total weight is: 20.0