Design and Analysis of Algorithms – 20ISL57A

Program 9 - Implement Knapsack problem.

```
#include<stdio.h>
void knapsack(int n, float weight[], float profit[], float capacity)
{
   float x[20], tp= 0;
   int i, j, rc;
   rc=capacity;
   for(i=0;i<n;i++)
      x[i]=0.0;
   for (i=0;i<n;i++)
   {
      if(weight[i]>rc)
         break;
      else
      {
         x[i]=1.0;
         tp= tp+profit[i];
         rc=rc-weight[i];
      }
   }
   if(i<n)
      x[i]=rc/weight[i];
   tp= tp + (x[i]*profit[i]);
   printf("The result vector is:\n");
   for(i=0;i<n;i++)
      printf("%0.2f\n",x[i]);
   printf("Maximum profit is: %0.2f\n", tp);
}
int main()
{
   float weight[20], profit[20], capacity, ratio[20], temp;
   int n, i ,j;
```

```
    printf ("Enter the no. of objects:\n ");
    scanf ("%d",&n);
    printf ("Enter the weights and profits of each object:\n ");
    for (i=0;i<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;
            }
        }
    }
    knapsack(n, weight, profit, capacity);
}
```