**Fractional Knapsack:**

#include <bits/stdc++.h>

#include <algorithm>

using namespace std

bool cmp(double a, double b)

{

    return (a > b);

}

double knapsack(int cap, double wt[], double val[], int n)

{

    double x1[n], x2[n], x[n], r[n], finalval = 0.0;

    int i, currentwt = 0, j, remain = 0;

    for (i = 0; i < n; i++)

    {

        r[i] = val[i] / wt[i];

    }

    sort(r, r + n, cmp);

    for (i = 0; i < n; i++)

    {

        for (j = 0; j < n; j++)

        {

            if (r[i] == (double)val[j] / wt[j])

            {

                x1[i] = val[j];

                x2[i] = wt[j];

                val[j] = 0;

                wt[j] = 0;

            }

        }

    }

    for (i = 0; i < n; i++)

    {

        if (currentwt + x2[i] <= cap)

        {

            currentwt += x2[i];

            finalval += x1[i];

        }

        else

        {

            remain = cap - currentwt;

            finalval += (double)r[i] \* (double)remain;

            break;

        }

    }

    return finalval;

}

int main()

{

    int n, cap, i;

    cout << "Enter the capacity" << endl;

    cin >> cap;

    cout << "Enter the number of elements" << endl;

    4 cin >> n;

    double wt[n], val[n];

    for (i = 0; i < n; i++)

    {

        cout << "Enter the weight for " << i << " : " << endl;

        cin >> wt[i];

        cout << "Enter the value for " << i << " : " << endl;

        cin >> val[i];

    }

    cout << knapsack(cap, wt, val, n);

    return 0;

}

**Output:**

Enter the capacity

50

Enter the number of elements

3

Enter the weight for 1 :

10

Enter the value for 1 :

60

Enter the weight for 2 :

20

Enter the value for 2 :

100

Enter the weight for 3 :

30

Enter the value for 3 :

120

240