**Name: Surwade Trisharan Rajesh**

**Roll no.: 48**

//Write a programme to find solution of knapsack instant

#include <iostream>

using namespace std;

class Knapsack

{

    float weight[20], profit[20], capacity;

    int num;

    float ratio[20], temp;

public:

    void getData()

    {

        int i;

        cout << "Enter the no. of objects : ";

        cin >> num;

        cout << "Enter the weight & profit of each objects : ";

        for (i = 0; i < num; i++)

        {

            cin >> weight[i];

            cin >> profit[i];

        }

        cout << "Enter the capacity of each kanpsack : ";

        cin >> capacity;

        for (i = 0; i < num; i++)

        {

            cout << weight[i];

        }

        for (i = 0; i < num; i++)

        {

            ratio[i] = profit[i] / weight[i];

        }

    }

    void knapsack()

    {

        sortData();

        hknapsack(num, weight, profit, capacity);

    }

    void sortData();

    void hknapsack(int n, float weight[], float profit[], float capacity);

};

void Knapsack::sortData()

{

    int i, j;

    for (i = 0; i < num; i++)

    {

        for (j = i + 1; j < num; 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;

            }

        }

    }

}

void Knapsack::hknapsack(int n, float weight[], float profit[], float capacity)

{

    float x[20], tp = 0;

    int i, u;

    u = capacity;

    for (i = 0; i < n; i++)

        x[i] = 0.0;

    for (i = 0; i < n; i++)

    {

        if (weight[i] > u)

            break;

        else

        {

            x[i] = 1.0;

            tp = tp + profit[i];

            u = u - weight[i];

        }

    }

    if (i < n)

        x[i] = u / weight[i];

    tp = tp + (x[i] \* profit[i]);

    cout << "\n the result vector is : ";

    for (i = 0; i < n; i++)

        cout << " " << x[i];

    cout << "\n Maximum profit is : " << tp;

}

int main()

{

    Knapsack ksd;

    ksd.getData();

    ksd.knapsack();

};

Output:

Enter the no. of objects : 4

Enter the weight & profit of each objects : 12 32

11 22

45 44

12 43

Enter the capacity of each kanpsack : 20

the result vector is : 1 0.666667 0 0

Maximum profit is : 64.3333