// C++ program to solve fractional Knapsack Problem

#include <bits/stdc++.h>

using namespace std;

struct Item {

int value, weight;

Item(int value, int weight)

{

this->value = value;

this->weight = weight;

}

};

bool cmp(struct Item a, struct Item b)

{

double r1 = (double)a.value / (double)a.weight;

double r2 = (double)b.value / (double)b.weight;

return r1 > r2;

}

double fractionalKnapsack(int W, struct Item arr[], int N)

{

// Sorting Item on basis of ratio

sort(arr, arr + N, cmp);

double finalvalue = 0.0;

// Looping through all items

for (int i = 0; i < N; i++) {

if (arr[i].weight <= W) {

W -= arr[i].weight;

finalvalue += arr[i].value;

}

else {

finalvalue

+= arr[i].value

\* ((double)W / (double)arr[i].weight);

break;

}

}

return finalvalue;

}

// Driver code

int main()

{

int W = 50;

Item arr[] = { { 60, 10 }, { 100, 20 }, { 120, 30 } };

int N = sizeof(arr) / sizeof(arr[0]);

// Function call

cout << fractionalKnapsack(W, arr, N);

return 0;

}

Maximum value we can obtain = 240