****

Assignment No: 3

Title Name: Solve a fractional Knapsack problem using a greedy method.

Name: Mitali Chhipa

Class : BE **Div**: B **Batch**: B

Exam Seat No/Roll No: 405B052

****

Program:

```
/*
   DAA: Assignment No - 3
   Title : Solve a fractional Knapsack problem using a greedy method.
*/

#include <bits/stdc++.h>
using namespace std;

// Structure for an item which stores weight and corresponding value of Item
struct Item
{
   int value, weight;
   // Constructor
   Item(int value, int weight)
   {
      this->value = value;
      this->weight = weight;
   }
};

// Comparison function to sort Item according to val/weight ratio
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)
{
    sort(arr, arr + N, cmp);
    double finalvalue = 0.0; // Result (value in Knapsack)
    for (int i = 0; i < N; i++)
    {
        // If adding Item won't overflow, add it completely
        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; // Weight of knapsack
    Item arr[] = {{60, 10}, {100, 20}, {120, 30}};
    int N = sizeof(arr) / sizeof(arr[0]);

    // Function call
    cout << "Maximum value we can obtain = "
         << fractionalKnapsack(W, arr, N);
    return 0;
}
```

Output: