```
//Knapsack problem
#include <iostream>
#include <vector>
#include <algorithm>

using namespace std;

int main() {
   int n, m;
   cout<<"Enter max weight and no weights"<<endl;
   cin >> m >> n;

   vector<int> p(n); // price array
   vector<int> w(n); // weight array

   for (int i = 0; i < n; i++) {
      cout<<"Enter profit and weight for "<<i+1<<endl;
      cin >> p[i] >> w[i];
   }

   vector<pair<double, int>> pw(n);
   for (int i = 0; i < n; i++) {
      pw[i] = make_pair((double)p[i] / w[i], i); // pw ratio of that object
   }
   sort(pw.begin(), pw.end(), greater<pair<double, int>>()); // sort in descending order

   double profit = 0;
   int k = 0;
   while (m > 0 && k < n) {
      int idx = pw[k].second;
      if (w[idx] <= m) {
         m -= w[idx];
         profit += p[idx];
         cout << "Object " << idx + 1 << " added with weight " << w[idx] << " and profit " << p[idx] << endl;
      }
      else {
         profit += (pw[k].first * m); // add fractional part of pw ratio
         cout << "Object " << idx + 1 << " added with weight " << m << " and profit " << pw[k].first * m << endl;
         m = 0;
      }
      k++;
   }

   cout << "Total profit: " << profit << endl;

   return 0;
}
```

```
PS E:\GIt> cd "e:\GIt\SEM-4\AOA\" ; if ($?) { g++ Knapsack.cpp -o Knapsack } ; if ($?) { .\Knapsack }
Enter max weight and no weights
15
7
Enter profit and weight for 1
10 2
Enter profit and weight for 2
5 3
Enter profit and weight for 3
15 5
Enter profit and weight for 4
7 7
Enter profit and weight for 5
6 1
Enter profit and weight for 6
18 4
Enter profit and weight for 7
3 1
Object 5 added with weight 1 and profit 6
Object 1 added with weight 2 and profit 10
Object 6 added with weight 4 and profit 18
Object 7 added with weight 1 and profit 3
Object 3 added with weight 5 and profit 15
Object 2 added with weight 2 and profit 3.33333
Total profit: 55.3333
PS E:\GIt\SEM-4\AOA>
```