#include <iostream>

#include <vector>

#include <iomanip>

using namespace::std;

struct item {

string obj;

int value, weight;

item(string o, int v, int w) : obj(o), value(v), weight(w) {}

};

class priority\_queue {

private:

void binary\_insert(item x, int high, int low) {

int x\_weighted\_value = x.value / x.weight;

while (low <= high) {

const int mid = low + (high - low) / 2;

if (x\_weighted\_value == pq.at(mid).value / pq.at(mid).weight) {

pq.insert(pq.begin() + mid, x);

return;

}

else if (x\_weighted\_value > pq.at(mid).value / pq.at(mid).weight) {

low = mid + 1;

}

else {

high = mid - 1;

}

}

pq.insert(pq.begin() + low, x);

}

public:

vector<item> pq;

void additem(item x) {

binary\_insert(x, pq.size() - 1, 0);

}

const item get\_highest() {

item x = pq.back();

pq.pop\_back();

return x;

}

};

void print\_fp\_solution(const vector<item>& bagged\_items) {

cout << left << setw(20) << "Item Name"

<< right << setw(10) << "Weight"

<< right << setw(10) << "Value" << endl;

cout << setfill('-') << setw(40) << "" << setfill(' ') << endl;

int total\_weight = 0;

int total\_value = 0;

for (const auto& item : bagged\_items) {

cout << left << setw(20) << item.obj

<< right << setw(10) << item.weight

<< right << setw(10) << item.value << endl;

total\_weight += item.weight;

total\_value += item.value;

}

cout << setfill('-') << setw(40) << "" << setfill(' ') << endl;

cout << left << setw(20) << "Totals:"

<< right << setw(10) << total\_weight

<< right << setw(10) << total\_value << endl;

}

void farmer\_problem(const vector<item>& cave\_items, int max\_weight) {

priority\_queue cave\_items\_sort;

vector<item> bagged\_items;

int bag\_weight = 0;

for (item x : cave\_items) {

cave\_items\_sort.additem(x);

}

for (item curr : cave\_items\_sort.pq) {

item curr = cave\_items\_sort.get\_highest();

if (max\_weight > bag\_weight + curr.weight) {

bag\_weight += curr.weight;

bagged\_items.push\_back(curr);

}

}

print\_fp\_solution(bagged\_items);

}

int main()

{

vector<item> items;

items.push\_back(item("Elephant horn 1", 4000, 2));

items.push\_back(item("Elephant horn 2", 4500, 3));

items.push\_back(item("Hammer", 4500, 3));

items.push\_back(item("Scissors", 500, 1));

items.push\_back(item("Vase 1", 5000, 2));

items.push\_back(item("Vase 2", 7000, 3));

items.push\_back(item("Vase 3", 8000, 7));

items.push\_back(item("Stone neck chain", 1000, 1));

farmer\_problem(items, 13);

}

Output:

Item Name Weight Value

----------------------------------------

Vase 1 2 5000

Vase 2 3 7000

Elephant horn 1 2 4000

Elephant horn 2 3 4500

Stone neck chain 1 1000

Scissors 1 500

----------------------------------------

Totals: 12 22000

A screenshot of a computer

AI-generated content may be incorrect.