```cpp
#include<iostream>
#include<fstream>
#include"PQ.h"

int main(){

    //Open text file for reading
    std::ifstream file;

    file.open("jouleFile.txt");

    int numItems, capacity = 0;

    file >> numItems >> capacity;

    Priority_Queue pq;

    for(int i = 0; i < numItems; i++){
        item* knap_item = new item();
        file >> knap_item-> name >> knap_item-> value >> knap_item-> weight;
        knap_item-> ratio = (double) knap_item-> value / knap_item-> weight;
        pq.enqueue(knap_item);
    }

    file.close();
    //file closed

    //Used for printing out the priority queue using preorder traversal
/*
    node* head_node = new node();
    head_node = pq.getHeadNode();
    pq.printPreorder(head_node);
*/

    //Beginning the Greedy Algorithm
    Priority_Queue knapsack;
    int totalWeight = 0;
    int numSacked = 0;
    int totalValue = 0;

    for(int i = 0; i < numItems; i++){
        item* retrieved_item = new item();
        retrieved_item = pq.dequeue();
        if (totalWeight + retrieved_item-> weight < capacity){
            knapsack.enqueue(retrieved_item);
            totalValue = totalValue + retrieved_item-> value;
            totalWeight = totalWeight + retrieved_item-> weight;
            numSacked++;
        }
    }

    //Printing out the results
    std::cout << numSacked << std::endl;
    std::cout << totalWeight << std::endl;
    std::cout << totalValue << std::endl;
    for(int i = 0; i < numSacked; i++){
        item* sacked_item = new item();
        sacked_item = knapsack.dequeue();
        std::cout << sacked_item-> name << " " << sacked_item-> value << " " <<
        sacked_item-> weight << std::endl;
    }

    //Finished!
    return 0;
}
```