

Задание выполняется с использованием прилагаемых файлов.

[Скачать архив с файлами.](#)

Для перевозки партии грузов различной массы выделен грузовик, но его грузоподъёмность ограничена, поэтому перевезти сразу все грузы не удастся. Грузы массой от 210 до 220 кг грузят в первую очередь. На оставшееся после этого место стараются взять как можно больше грузов. Если это можно сделать несколькими способами, выбирают тот способ, при котором самый большой из выбранных грузов имеет наибольшую массу. Если и при этом условии возможно несколько вариантов, выбирается тот, при котором наибольшую массу имеет второй по величине груз, и т.д.

Известны количество грузов, масса каждого из них и грузоподъёмность грузовика. Необходимо определить количество и общую массу грузов, которые будут вывезены при погрузке по вышеописанным правилам.

Входные данные

Первая строка входного файла содержит два целых числа: N – общее количество грузов и M – грузоподъёмность грузовика в кг. Каждая из следующих N строк содержит одно целое число – массу груза в кг.

Пример входного файла

```
6 615
140
215
120
160
100
340
```

В данном случае сначала нужно взять груз массой 215 кг. После этого можно вывезти ещё максимум 3 груза. Это можно сделать тремя способами: $140 + 120 + 100$, $140 + 160 + 100$, $120 + 160 + 100$. Выбираем способ, при котором вывозится груз наибольшей возможной массы. Таких способов два: $140 + 160 + 100$ и $120 + 160 + 100$. Из этих способов выбираем тот, при котором больше масса второго по величине груза, то есть $140 + 160 + 100$. Всего получается 4 груза общей массой 615 кг. В ответе надо записать числа 4 и 615.

В ответе запишите два целых числа: сначала максимально возможное количество грузов, затем их общую массу.

Число

Число

sum=20 M=68

47	48
----	----

```
#include <iostream>
#include <vector>
#include <vector>
using namespace std;
void print_vec(vector<int> &v)
{
    cout<<endl;
    for (auto i = v.begin(); i != v.end(); ++i)
    {
        cout<<" " << *i << " ";
    }
    cout<<endl;
}

void print_vec2(vector<int> &v)
{
    cout<<endl;
    for (vector<int>::iterator i = v.begin(); i != v.end(); ++i)
    {
        cout<<" " << *i << " ";
    }
    cout<<endl;
}

int main()
{
    FILE *fp=fopen("23.txt","r");
    int N, M, g;
    vector<int> mass, mass_else, mass_greed;
    fscanf(fp, "%d %d", &N, &M);
    cout<<"N=" << N << endl;
    while (fscanf(fp, "%d", &g) != -1)
    {
        cout<<g<<endl;
        if (g >= 210 && g <= 220)
            mass.push_back(g);
        else
            mass_else.push_back(g);
    }

    //print_vec2(mass);
    sort(mass.begin(), mass.end());
    print_vec(mass);
    int temp=0, i=0, sum;
    while (temp+mass[i] <= M && i < mass.size())
    {
        temp+=mass[i];
        cout<<"temp=" << temp << endl;
        sort(mass_else.begin(), mass_else.end());
        reverse(mass_else.begin(), mass_else.end());
        //print_vec(mass_else);
        //temp<-M;
        {
            i=mass_else.size()-1;
            sum=temp;
            while (sum+mass_else[i] <= M && i >= 0)
            {
                mass_greed.push_back(mass_else[i]);
                sum+=mass_else[i];
                mass_else.pop_back();
                i--;
            }
        }
        sort(mass_else.begin(), mass_else.end());
        print_vec(mass_greed);
        print_vec(mass_else);
        //cout<<"sum=" << sum << endl;
        //correction of mass_greed
        sum=mass_greed[mass_greed.size()-1];
        i=0;
        while (sum+mass_else[i] <= M)
        {
            i++;
        }
        i--;
        sum+=mass_greed[mass_greed.size()-1];
        int i, flag=0;
        cout<<"max el=" << mass_else[i] << endl;
        cout<<"size=" << mass_greed.size() << endl;
        for (i >= 0; i--)
        {
            for (j=mass_greed.size()-1; j >= 0; j--)
            {
                if (sum+mass_greed[j]+mass_else[i] <= M && sum+sum-mass_greed[j]+mass_else[i])
                {
                    sum=sum-mass_greed[j]+mass_else[i];
                    i=mass_greed[j];
                    mass_greed[j]=mass_else[i];
                    mass_else[i]=i;
                    i++;
                    break;
                }
            }
        }
        cout<<"i=" << i << endl;
        if (i >= 0)
        {
            mass_greed[j]=mass_else[i];
        }
        //new method
        //i=i;
        for (mass_greed.size()-1; i >= 0; i--)
        {
            for (int u=0; u < mass_else.size()-1; u++)
            {
                if (sum-mass_greed[i]+mass_else[u] <= M && sum+sum-mass_greed[i]+mass_else[u])
                {
                    sum=sum-mass_greed[i]+mass_else[u];
                    i=mass_greed[i];
                    mass_greed[i]=mass_else[u];
                    mass_else[u]=i;
                }
            }
            if (sum==M)
            {
                break;
            }
        }
        if (sum==M)
        {
            break;
        }
    }
    //
    print_vec(mass_greed);
    cout<<"sum=" << sum << endl;
    cout<<"size=" << mass_greed.size() << endl;
    return 0;
}
```

9880-mass_greed[mass_greed.size()-1]
=9757

9757+max el(242)=9999

M-9999=1

9880-123+242=9999

9880-122+242=10000

9880-121+241=10000

615

575-140+160=595

575-120+160=615

```
first_attempt= [10, 10, 15, 15, 18, 21, 21, 22, 22, 22, 23, 24, 25, 26, 27, 27,
28, 29, 29, 31, 32, 33, 33, 33, 34, 34, 38, 38, 41, 44, 45, 48, 49, 49, 50, 51,
51, 52, 53, 56, 58, 59, 60, 62, 63, 63, 64, 65, 66, 66, 67, 67, 67, 68, 68, 68,
68, 68, 68, 69, 69, 71, 71, 71, 72, 73, 73, 74, 75, 75, 76, 79, 79, 80, 81, 81,
83, 84, 84, 85, 86, 89, 90, 91, 93, 93, 97, 99, 99, 100, 101, 101, 102, 102, 102
, 103, 104, 104, 105, 105, 108, 113, 114, 116, 118, 121, 121, 122, 123, 123]
total_count= 122 total_sum= 9880
first_attempt= [10, 10, 15, 15, 18, 21, 21, 22, 22, 22, 23, 24, 25, 26, 27, 27,
28, 29, 29, 31, 32, 33, 33, 33, 34, 34, 38, 38, 41, 44, 45, 48, 49, 49, 50, 51,
51, 52, 53, 56, 58, 59, 60, 62, 63, 63, 64, 65, 66, 66, 67, 67, 67, 68, 68, 68,
68, 68, 68, 69, 69, 71, 71, 71, 72, 73, 73, 74, 75, 75, 76, 79, 79, 80, 81, 81,
83, 84, 84, 85, 86, 89, 90, 91, 93, 93, 97, 99, 99, 100, 101, 101, 102, 102, 102
, 103, 104, 104, 105, 105, 108, 113, 114, 116, 118, 121, 121, 123, 123, 123]
7414
122 10000
```