**Luck Balance**

Lena is preparing for an important coding competition that is preceded by a number of sequential preliminary contests. Initially, her luck balance is 0. She believes in "saving luck”, and wants to check her theory. Each contest is described by two integers,

 L[i] and T[i]:

* L[i] is the amount of luck associated with a contest. If Lena *wins* the contest, her luck balance will *decrease* by L[i] ; if she *loses* it, her luck balance will *increase* by L[i] .
* T[i] denotes the contest's *importance rating*. It's equal to 1 if the contest is *important*, and it's equal to 0 if it's *unimportant*.

If Lena loses no more than  *important* contests, what is the maximum amount of luck she can have after competing in all the preliminary contests? This value *may* be negative.

For example,  k=2 and:

Contest L[i] T[i]

1 5 1

2 1 1

3 4 0

If Lena loses all of the contests, her will be 5+1+4= 10. Since she is allowed to lose 2  important contests, and there are only 2 important contests. She can lose all three contests to maximize her luck at 10. If k=1, she has to win at least 1 of the 2 important contests. She would choose to win the lowest value important contest worth 1 . Her final luck will be 5+4-1=8 .

**Function Description**

Complete the *luckBalance* function in the editor below. It should return an integer that represents the maximum luck balance achievable.

luckBalance has the following parameter(s):

* *k*: the number of important contests Lena can lose
* *contests*: a 2D array of integers where each contests[i] contains two integers that represent the luck balance and importance of the i^th  contest.

**Input Format**

The first line contains two space-separated integers n and k, the number of preliminary contests and the maximum number of important contests Lena can lose.   
Each of the next n lines contains two space-separated integers, L[i] and T[i] , the contest's luck balance and its importance rating.

**Constraints**

* 1<=n<=100
* 0<=k<=N
* I<=L[i]<=10^4
* T[i] {1.0}

**Output Format**

Print a single integer denoting the maximum amount of luck Lena can have after all the contests.

**Sample Input**

6 3

5 1

2 1

1 1

8 1

10 0

5 0

**Sample Output**

29

**Explanation**

There are n=6 contests. Of these contests,4 are important and she cannot lose more than k=3 of them. Lena maximizes her luck if she wins 3rd the important contest (where L[i]=1) and loses all of the other five contests for a total luck balance of 5=2=8=10=5-1=29.

using System.CodeDom.Compiler;

using System.Collections.Generic;

using System.Collections;

using System.ComponentModel;

using System.Diagnostics.CodeAnalysis;

using System.Globalization;

using System.IO;

using System.Linq;

using System.Reflection;

using System.Runtime.Serialization;

using System.Text.RegularExpressions;

using System.Text;

using System;

class Solution {

// Complete the luckBalance function below.

static int luckBalance(int k, int[][] contests) {

int n= contests.Length;

int[] imp = new int[n];

int l = 0,m=-1;

for(int i = 0 ; i<n;i++){

if(contests[i][1]==1){

//total include import contest

imp[++m]= contests[i][0];

}

//total contests

l+=contests[i][0];

}

//sort import contest small first

Array.Sort(imp);

//m-k must win contests should exclude from total contest

for(int z = 0 ;z<=m-k;z++){

l-=(2\*(imp[n-1-m+z]));

}

return l;

}

static void Main(string[] args) {

TextWriter textWriter = new StreamWriter(@System.Environment.GetEnvironmentVariable("OUTPUT\_PATH"), true);

string[] nk = Console.ReadLine().Split(' ');

int n = Convert.ToInt32(nk[0]);

int k = Convert.ToInt32(nk[1]);

int[][] contests = new int[n][];

for (int i = 0; i < n; i++) {

contests[i] = Array.ConvertAll(Console.ReadLine().Split(' '), contestsTemp => Convert.ToInt32(contestsTemp));

}

int result = luckBalance(k, contests);

textWriter.WriteLine(result);

textWriter.Flush();

textWriter.Close();

}

}

You have earned 20.00 points!

These points will also count towards your progress in the Problem Solving Badge.

**100%**

**Congratulations**

You solved this challenge. Would you like to challenge your friends?

[Next Challenge](https://www.hackerrank.com/challenges/reverse-shuffle-merge?h_l=interview&playlist_slugs%5B%5D=interview-preparation-kit&playlist_slugs%5B%5D=greedy-algorithms&h_r=next-challenge&h_v=zen)

* **Test case 0**
* **Test case 1**
* **Test case 2**
* **Test case 3**
* **Test case 4**
* **Test case 5**
* **Test case 6**
* **Test case 7**
* **Test case 8**
* **Test case 9**
* **Test case 10**
* **Test case 11**
* **Test case 12**

Compiler Message

**Success**

Input (stdin)

Download

* **6 3**
* **5 1**
* **2 1**
* **1 1**
* **8 1**
* **10 0**
* **5 0**

Expected Output

Download

* **29**