CodeForces

Contest 1374

Problem E1

@mrkhosravian

Problem

Summer vacation has started so Alice and Bob want to play and joy, but... Their mom doesn't think so. She says that they have to read some amount of books before all entertainments. Alice and Bob will read each book **together** to end this exercise faster.

There are nn books in the family library. The i^{th} book is described by three integers: t_i — the amount of time Alice and Bob need to spend to read it, aiai (equals 11 if Alice likes the i^{th} book and 00 if not), and b_i (equals 11 if Bob likes the i^{th} book and 00 if not).

- So they need to choose some books from the given nn books in such a way that: Alice likes **at least** kk books from the chosen set and Bob likes **at least** kk books from the chosen set;
- the total reading time of these books is **minimized** (they are children and want to play and joy as soon a possible).

The set they choose is **the same** for both Alice an Bob (it's shared between them) and they read all books **together**, so the total reading time is the sum of t_i over all books that are in the chosen set.

Your task is to help them and find any suitable set of books or determine that it is impossible to find such a set.

Input

The first line of the input contains two integers n and k (1 ≤ k ≤ n ≤ 2·1051 ≤ k ≤ n ≤ 2·105).

The next n lines contain descriptions of books, one description per line: the i^{th} line contains three integers t_i, aiai and b_i (1 ≤ ti ≤ 1041 ≤ t_i ≤ 104, 0 ≤ a_i, b_i ≤10≤ a_i ,b_i ≤ 1), where:

- t_i — the amount of time required for reading the i^{th} book;
- aiai equals 11 if Alice likes the i^{th} book and 00 otherwise;
- bibi equals 11 if Bob likes the i^{th} book and 00 otherwise.

Output

- If there is no solution, print only one integer -1. Otherwise print one integer T — the minimum total reading time of the suitable set of books.

Example 1

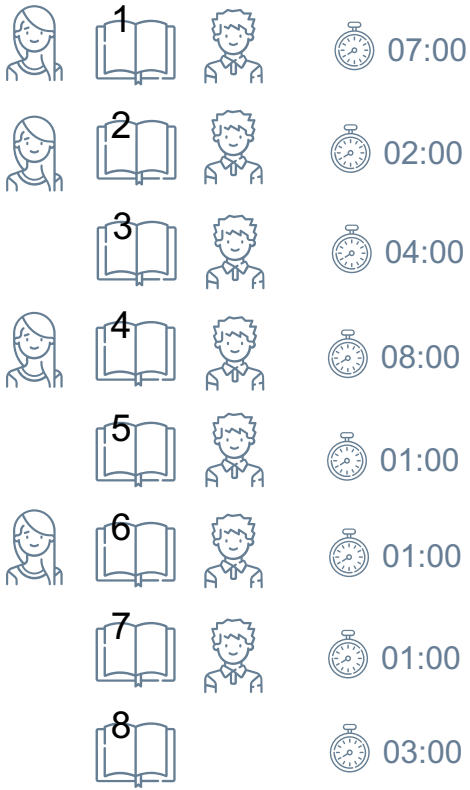
Input:

```
8 4
7 1 1
2 1 1
4 0 1
8 1 1
1 0 1
1 1 1
1 0 1
3 0 0
```

Output:

```
18
```

Solve example 1



Alice Bob

There are 8 books
Alice and Bob both likes 4 books

There are four types of book:

1.	0 0 - both Alice and Bob doesn't like these books	books { 8 }
2.	1 0 - only Alice likes these books	books { }
3.	0 1 - only Bob likes these books	books { 3, 7 }
4.	1 1 - both Alice and Bob like these books	books { 1, 2, 4, 6 }

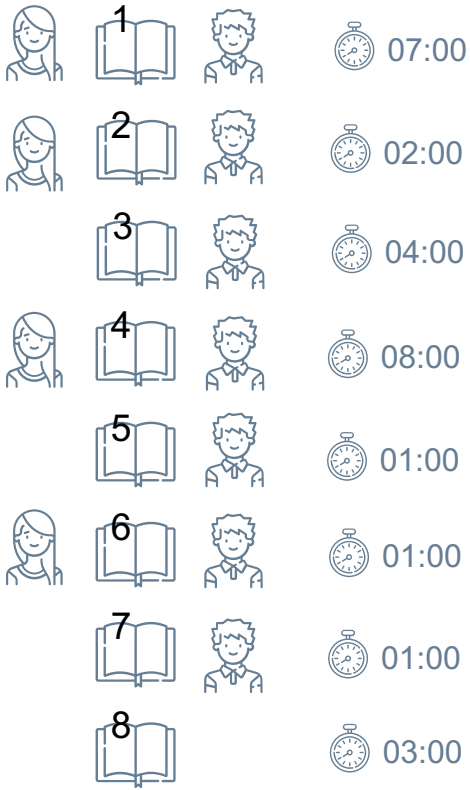
Now sort these groups by time:

1. { 03:00 }
2. { }
3. { 01:00, 04:00 }
4. { 01:00, 02:00, 07:00, 08:00 }

Make a total sum of these arrays (start from zero) :

1. { 00:00, 03:00 }
2. { 00:00 }
3. { 00:00, 01:00, 05:00 }
4. { 00:00, 01:00, 03:00, 10:00, 18:00 }

Solve example 1



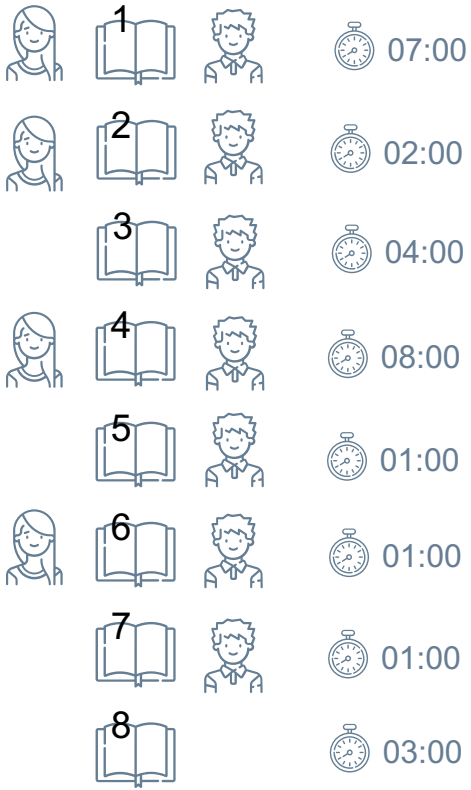
Alice Bob

There are 8 books
Alice and Bob both likes 4 books

Now we check for every possible cnt value:
Possible cnt values: 0..4 # from 0 to min(k, total books in group 4)

- If cnt = 0 then there should be 4 book that only alice likes it and 4 books that bob likes it.There no books that just alice likes it -> cnt = 0 is not a good choice
- If cnt = 1 then there should be 3 books that only alice likes it and 3 books that bob likes it. There no books that just alice likes it -> cnt = 1 is not a good choice
- If cnt = 2 then there should be 2 books that only alice likes it and 2 books that bob likes it. There no books that just alice likes it -> cnt = 2 is not a good choice
- If cnt = 3 then there should be 1 books that only alice likes it and 1 books that bob likes it. There no books that just alice likes it -> cnt = 3 is not a good choice
- If cnt = 4 -> this is the best condition because alice and bob can read all of their books together

Solve example 1



Alice Bob

There are 8 books
Alice and Bob both likes 4 books

Finally after we found the best cnt value:

Answer is:

cnt^{th} element of sums 4 + (k – cnt)th element of the sums 1 + (k – cnt)th element of sums 2

Or

4^{th} element of group 4 + 0^{th} element of group 1 + 0^{th} element of group 2

Or

18:00 + 00:00 + 00:00 = 18:00

Now go to code

Initializing

```
int n = fr.nextInt();
int k = fr.nextInt();
List<Integer>[] times = new ArrayList[3];
List<Integer>[] sums = new ArrayList[3];
for (int i = 0; i < 3; i++) {
    times[i] = new ArrayList<>();
    sums[i] = new ArrayList<>();
}
```

* fr (fast reader) is a custom scanner that uses Java BufferedReader

Create Books groups And create sum array

```
while (n-- > 0) {
    int ti = fr.nextInt();
    int a = fr.nextInt();
    int b = fr.nextInt();

    if (a == b && b == 0) continue;
    times[a * 2 + b - 1].add(ti);
}

for (int i = 0; i < 3; i++) {
    Collections.sort(times[i]);
    sums[i].add(0);
    for (int item : times[i]) {
        sums[i].add(sums[i].get(sums[i].size() - 1) + item);
    }
}
```

- What is (a * 2 + b – 1) ?

By this we can generate numbers from 0 to 2, that are groups indexes
Notice that -1 is because we don't need 0 0 group at all!

Find the best cnt value

```
int ans = Integer.MAX_VALUE;
for (int count = 0; count < Math.min(k + 1, sums[2].size()); count++) {
    if (k - count < sums[0].size() && k - count < sums[1].size()) {
        ans = Math.min(ans, sums[2].get(count) + sums[0].get(k - count) + sums[1].get(k - count));
    }
}

if (ans == Integer.MAX_VALUE) ans = -1;
System.out.println(ans);
```

- We calculate all cnt values and find the minimum answer.
- If the ans variable has changed we can assume that there is an answer, otherwise, we print -1

Fast Reader – (skip if you just use java.util.Scanner)

```
class FastReader {
    private final BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
    private StringTokenizer st;

    public String nextLine() {
        try {
            return br.readLine();
        } catch (IOException ex) {
            throw new RuntimeException(ex);
        }
    }

    public String next() {
        while (st == null || !st.hasMoreTokens()) {
            st = new StringTokenizer(nextLine());
        }
        return st.nextToken();
    }

    public int nextInt() {
        return Integer.parseInt(next());
    }
}
```

Now You can solve the problem

FINISHED

I hope u enjoyed.

@mrkhosravian