C1 coding test sample-
Best horror movie ever

Programming L+Pr. (IP-18fPROGEG) – 2022 autumn semester

There are 45 minutes available, and you have +5 minutes for the TMS hand in. For a good solution maximum 50 points could be reached. During the work you are not allowed to talk, to use cell phones or internet. Remember! Cheating is not allowed, and action will be taken.

Before you start:

- Your code should not contain syntax errors (it should be able to compile), otherwise you will get 0 points.
- If there is a runtime error for any valid input, you will get half of the points.
- You can use printed or written notes/papers.
- The first two lines of your code should contain the following:
  // 2022-10-12 C1 coding test Best horror movie ever
  // Your Full Name Neptun code
- You have to write the "reading the inputs" and the "writing the outputs" part in your code. You have to include appropriate communication with the user (messages describing what you ask for and what you print)!
- You should follow the given algorithm and use the exact same variable names as in the algorithm and the specification.
- Use the data structures described in the specification to store the data.
- Don't forget to check the preconditions (input checking)!
- When you are done, upload your Program.cs file to TMS (C1 coding test assignment). The TMS system will not test it.

Best horror movie ever

We collected some of the trending horror films from IMDb and arranged them into a list. We are searching for films to watch, check our criteria's and create a program that gives back the films we want to watch.

a) Write a program that finds the title of the highest rated film from a given year.

b) Write a program that lists the titles that has the same rating as given in the first row.

Input

The first line of the standard input contains the number of films (1≤dataLength≤50), a year when certain films premiered (1932≤limitYear≤2022), and a rating from IMDb (0≤limitRating≤10). The next N lines contain the year (1932≤limitYear≤2022) that the film premiered, the rating (0≤limitRating≤10) of the film, and the title of the movie *(1 ≤ Length(title[i]) ≤ 50)*.

Output

The first line of the standard output should contain #1, after that in the next line the name of the highest rated film from a given year. The third row should contain #2, after that the next number of rows should contain the titles of those films that has the same rating as the given rating in the first row.

Example

Input

```
10 2018 7,3
2022;6,3;Scream
2018;5,3;The Nun
2017;7,3;It
2019;6,5;It Chapter Two
2018;6,5;Halloween
2021;5,5;Halloween Kills
2022;5,0;Halloween Ends
2018;7,5;A Quiet Place
2016;7,3;Split
2021;6,3;The Conjuring: The Devil Made Me Do It
```

Output

```
#1
A Quiet Place
#2
It
Split
```

Specification

Input: $dataLength, limitYear \in \mathbb{N},\ limitRating \in \mathbb{R},$

$year[1..dataLength] \in \mathbb{N},$

$rating[1..dataLength] \in \mathbb{R},$

$title[1..dataLength] \in \mathbb{T}$

Output: $filmTiltleForMaxVal, filmTitleAll \in \mathbb{T}$

Precondition: $1 \leq dataLength \geq 50, 1932 \leq limitYear \geq 2022, 0 \leq limitRating \geq 10$

$\forall[i](1 \leq i \geq dataLength): 1932 \leq year[i] \geq 2022,$

$\forall[i](1 \leq i \geq dataLength): 0 \leq rating[i] \geq 10,$

$\forall[i](1 \leq i \geq dataLength): 1 \leq Length(title[i]) \geq 50$

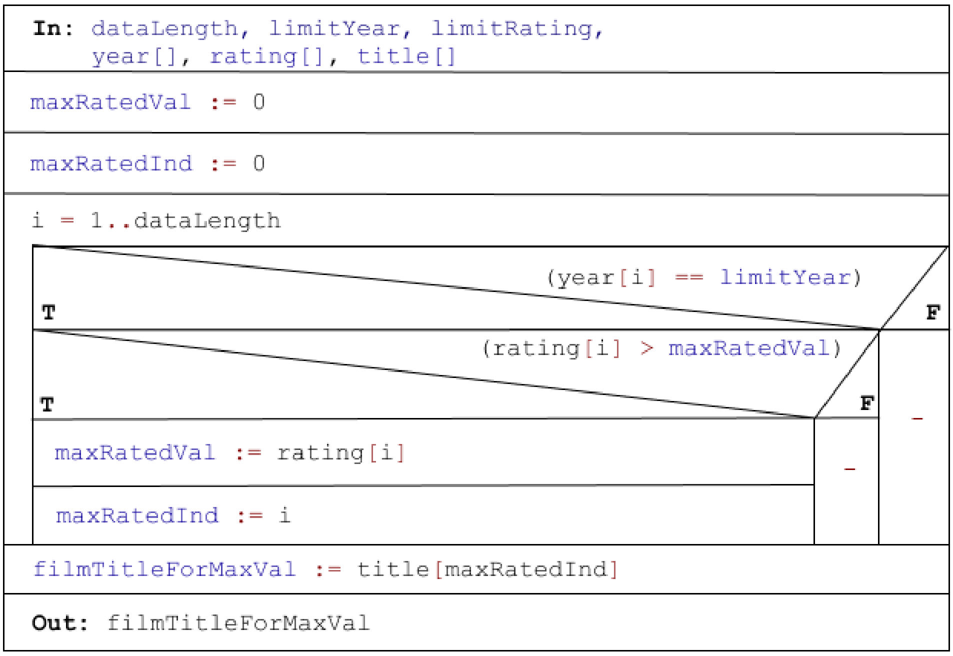
Postcondition:

a) $$filmTitleForMaxVal = \underset{\substack{i=1 \\ year[i] = limitYear}}{\overset{dataLength}{MAXVAL}} (rating[i])$$

b) $$filmTitleAll = \underset{i=1}{\overset{dataLength}{\forall}} (rating[i] = limitRating)$$

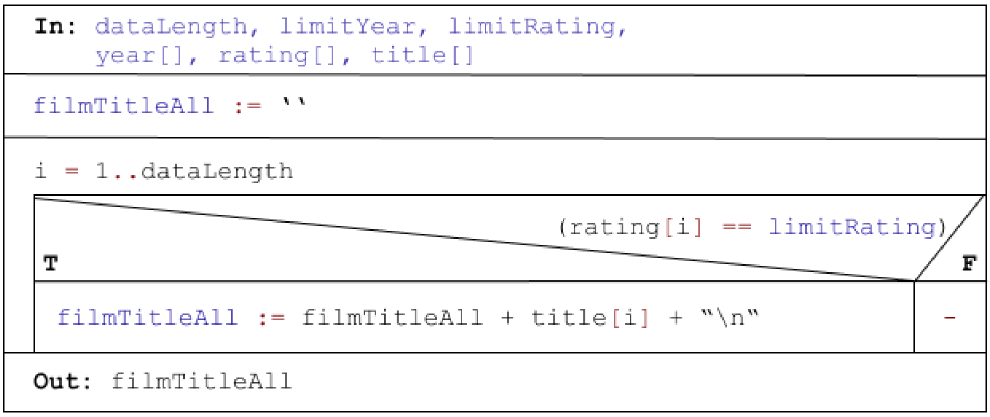
Algorithm

a)



```
In: dataLength, limitYear, limitRating,
    year[], rating[], title[]
maxRatedVal := 0
maxRatedInd := 0
i = 1..dataLength
    (year[i] == limitYear)
    T
        (rating[i] > maxRatedVal)
        T
            maxRatedVal := rating[i]
            maxRatedInd := i
        F
            -
    F
        -
filmTitleForMaxVal := title[maxRatedInd]
Out: filmTitleForMaxVal
```

b)



```
In: dataLength, limitYear, limitRating,
    year[], rating[], title[]
filmTitleAll := ''
i = 1..dataLength
    (rating[i] == limitRating)
    T
        filmTitleAll := filmTitleAll + title[i] + “\n”
    F
        -
Out: filmTitleAll
```