1   Introduction

In this project, you are given a programming question, and you will develop an efficient algorithm and implement it. You can use any of the programming languages you prefer, there is no limitation. However, usage of any libraries except the standard language library are forbidden. Please request permission if you plan to use any of these (if you are using Python, NumPy is allowed).

Please open the Google Sheet Link, and enter your group number, names of the members and the emails of the members to the cell, where your project is listed. Please keep in mind that the sheet document is divided into different sub-sheets. Thus, select the correct sub-sheet with respect to your group number. All of the test cases for each of the projects can be viewed and downloaded from this Google Drive Link.

Please open a GitHub public repository, include all of your members as contributors and add the repository link to the given Google Sheet document. This step is quite important for us to see your progress and has to be done quickly. Therefore, in the README file please keep a list of completed steps, a TO DO list and the results retrieved if there are any. You will also prepare a presentation and present to the TAs. Therefore, you should also create a Google Slides presentation and include its link to the Google Sheet document.

Please email to *comp305staff-group@ku.edu.tr*, if there is any problem in viewing the drive folder or modifying the document or if you have some troubles with any of the test cases.

2   Presentation Details

Each of the presentations should take   ~10 minutes and there will be a 5-minute Q&A session afterwards. If a presentation lasts longer than 10 minutes, then it will be interrupted. During the presentation each of the groups should explain and report:

- The algorithm you designed to solve the problem, the choices of the data structures you used and your reasoning.

- The time complexity of your algorithm (and the space complexity if applicable).

- Your run times for each of the test cases.

- Further improvements that can be done as future works.

This project does not expect from you to come up with just one solution and then test only that solution. For each of the problems you can start with some baseline approaches with more complexity and improve the baseline algorithm step by step. Be as creative as possible. Report different approaches you tested and why did you decide on the final algorithm you present. Your grading will be based on your creativity, your cumulative progress and how well did you present your approach. Additionally, there will be more test cases which won't be provided to you. Therefore, to be sure about the correctness of your algorithm, you should create extra test cases. Finally, having an efficient and fast algorithm is also as important as having correct algorithm. Do your best to find the most efficient and fastest solution. Your effort is really important for grading.

3 Deadlines

You can work on your project until the end of **23th May, 2021**. The project presentations will be held between **24th-28th of May, 2021**.

In the following pages, you can see the available project(s):

Baby Competition

One of the cutest things in the world is babies for the most of the people and they are fun to watch. To socialize babies, their families decided on to create a new game called BABY-GOAL. In that game, there are several goal post are distributed on the pitch and when the babies reach to the goal post they are given the toy that they liked the most. Babies can try to go any goal post they want and also they can try to go to other ones. Babies are move in turns. In other words, first baby starts moving when he/she stops, the next baby starts to move.(Assume that the babies are stays still in their place when they stop). Scores of the babies are decided on the each of the goal post which is number of turns that they spent on that goal-post. For instance if baby reach to one of the post in 3 turn, that baby's score on that goal post is 3. There may stubborn babies who takes their turn but do not go to the goal post. To handle that situation, there is upper bound l is decided and if baby takes more than l turn but do not reach to the post, baby's score on that goal post will be l. One thing to noticed that all families and all babies unaware about the upper bound and plays until to reach the goal posts. Scores of the babies are calculated as a sum of their scores on each goal post. For instance, if baby reach to the goal post A in 5 move and goal post B in 6 move and 7 and 8 move for C and D goal post and 8 move for goal post D and if l is bigger than the all number of moves (5,6,7,8,8) their overall score is sum of 5,6,7,8,8 which is 34.

To the illustrate better the problem, In the figure below there are 5 babies in the pitch and there are 5 goal post, and in turn each baby moves towards to goal until baby stops and other baby takes turn. As a result of this settings, some of the babies reach to the goal post in a shorter time if they move faster in their turn.
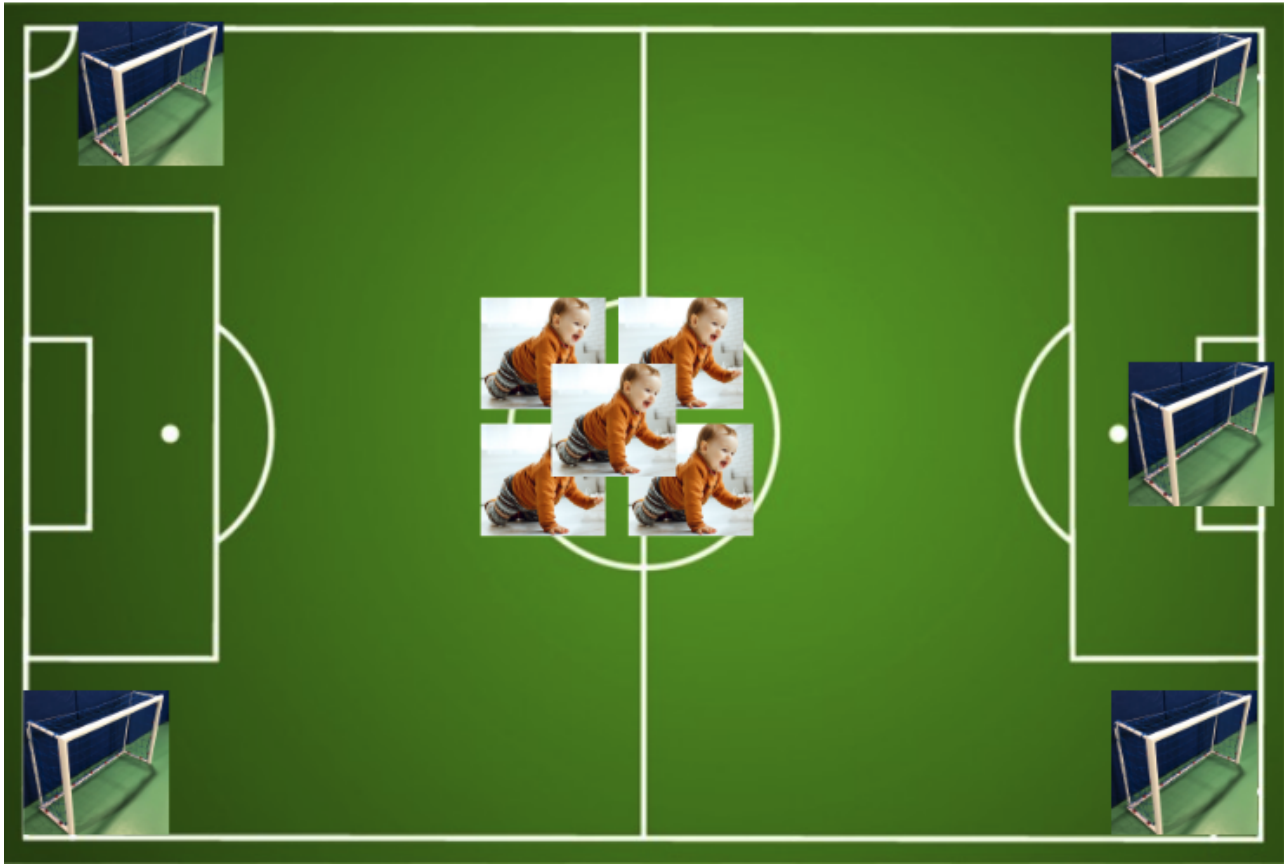

Figure 1: Babies and goal posts

After game ends, scores are recalculated based on the l. Competition rankings are calculated based on the recalculated scores. If 20 25 25 22 and 20 are the final scores of the 5 babies, their final place will be 2nd 5th 5th 3rd and 2nd in order with the given scores.

4.1 Inputs

The first line in the input is two integers. The first one is the number of babies that play the game and second one is the number of goal posts.

4.2 Output

Final place of the each baby that play the game. Each line corresponds to one baby and they are in the same order with the given input.

4.3 Sample Input Output

Sample Input 1	Sample Output 1
3 3	1
2 2 2	2
4 2 1	2
4 4 1	

Sample Input 2	Sample Output 2
6 4	1
3 1 2 2	2
4 3 2 2	5
6 6 3 2	5
7 3 4 3	4
3 4 2 4	3
2 3 3 5	

Figure 2: Sample Input and Output File.

Question According to this settings, you are given the scores of the babies in each goal post and you are asked to return the smallest possible final place for each babies. You should provide the most efficient solution, since your codes will be tested with big input size.