

## Contents

How to install and execute the program:.....	2
File Structure:.....	3
Game Guideline .....	4
Basic Requirements.....	5
Advanced Requirements .....	17

Group Member:

**Fung King Shun (56659723)**

How to install and execute the program:

Here I provided 2 methods for execute the program:

**Method 1 (directly run the .exe file):**

I have deployed the .py file which is an easy way to run the programs without the requirement of installing any libraries, packages or downloading python on your computer. You can find the .exe file in src/main.exe.

**Method 2 (install corresponding library and python version 2.7):**

Step1: Download and unzip the file

Step2: Download and install Anaconda for the website <https://www.anaconda.com/>

Step3: Create a virtual environment of python 2.7 in Anaconda

Step4: Open created virtual environment in Anaconda and run the command "cd [file download path + \src]"

Step5: Run the command "pip install -r requirements.txt"

Step6: Run the command "python main.py"

## File Structure:

For the new program, I have completed all basic and advanced requirements. Based on each feature, I will describe it in the next section. First of all, I want to show you the structure of the file. Compared to the original program, here are some changes of the file:

### 1. Added a gameRecord in the top folder

I added a new file called gameRecord file in the top folder to store game ranking records using CSV files. These data will be used as ranking records in the game and showing the top 10 results.

### 2. Added some new model .py in src file

#### ➤ star.py

This python file will be called in main.py and create the star object.

#### ➤ RankingTable.py

This python file will be called in main.py and shows the ranking table at the end of the game.

#### ➤ GameSetting.py

This python file will be called in main.py and used to change the game resolution settings.

## Game Control:






1. Changing game screen size
2. Moving the jeep using keyboard
3. Changing different setting using menu and mouse

For more details, you can push the keyboard 'h' in the game for game guidelines.

## Game Guideline

Game Rule: **“Harvest the most stars with the least amount of time”**

Game Object Description:

	jeep
	star
	cone
	accelerating ribbon
	sun

Game Flow:

1. Execute the program
2. Configure window resolution and input player name
3. Configure camera angle, viewpoint, lighting and material color
4. Push the keyboard 's' to start
5. Use the keyboard 'left' or 'right' button to avoid the cone to get a star or passing the accelerating ribbon to speed up the jeep
6. If the jeep touches the cone, runs off the road or jeep reaches the end point of the road, then it will end game
7. Record the game and display the ranking table

## Basic Requirements

### 1. Creating new objects

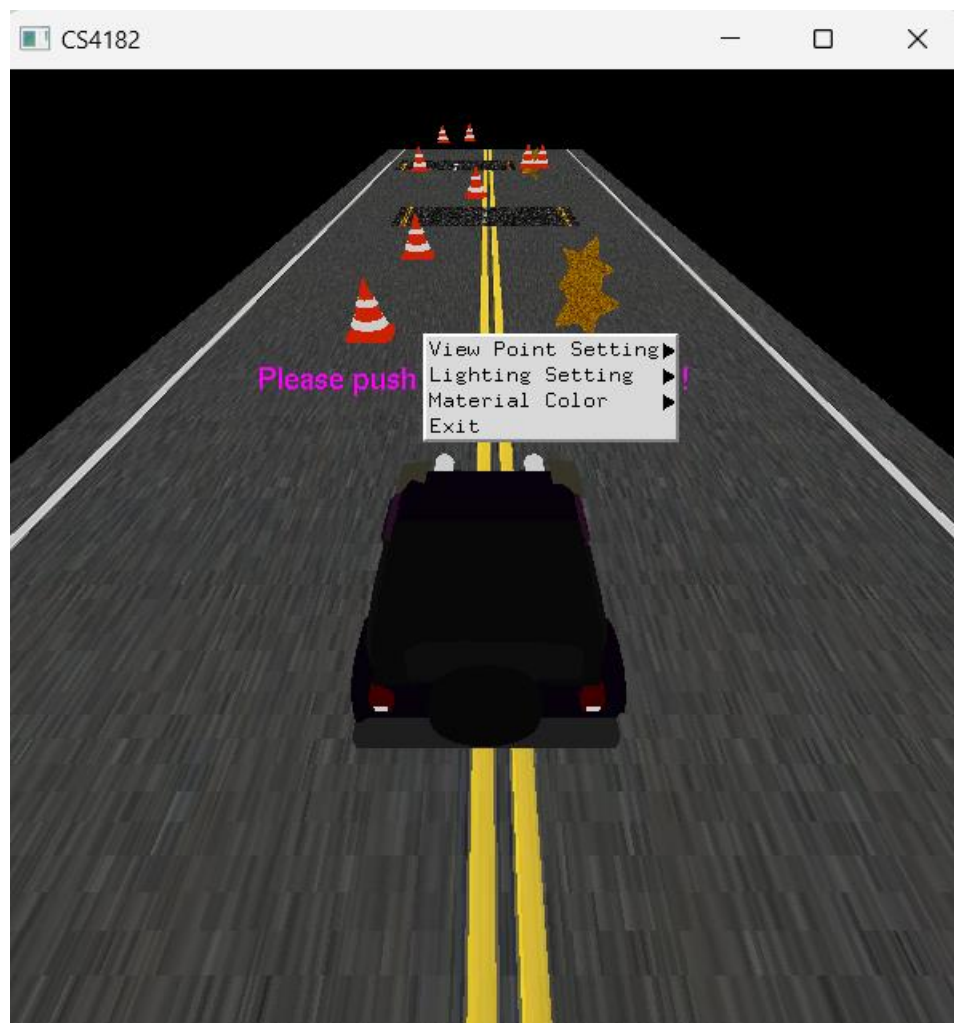
To create a new object, I selected to create a star with textures. First, I create a new module called `star.py` in `scr` file, this file will load the textures file `"/img/ starSparkle.jpg"` and object file `"/objects/star"` and it will save the information of each star object like position and size. Next, create 5 stars in `main.py` and position them at random positions. Considering that the stars should not overlap the cone object, the creation will check the star position and created corresponding star objects will not overlap with the cone. In the coding, you can find the function in `main.py`: `main()` and `star.py`. The following is the game screen:



Star with textures and creation will not overlap with cone object

## 2. Menu and Lighting

For the menu, I directly added new code in main.py using the OpenGL function `glutCreateMenu()`, `glutAddMenuEntry()`, `glutAttachMenu()`, `glutAddSubMenu()`. To open the menu, it can be done by right click the mouse and it will open the top menu. Then, it will have 4 items of selection 3 items for open submenu and 1 item for exit the program. In coding, you can find the function in main.py: `createMenu()`, `lightingColorMenuList(value)`, `lightingMenuList(value)`, `viewPointMenuList(value)`, `mainMenuList(value)`. The following is the game screen:



Mouse right click open Top-menu



Viewpoint sub-menu

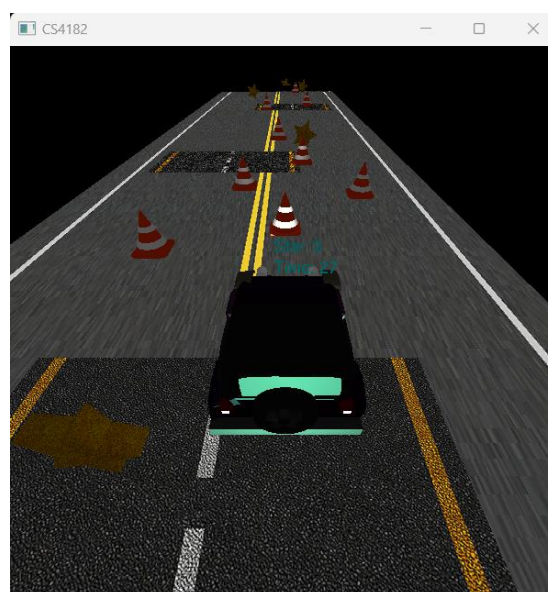


Lighting sub-menu



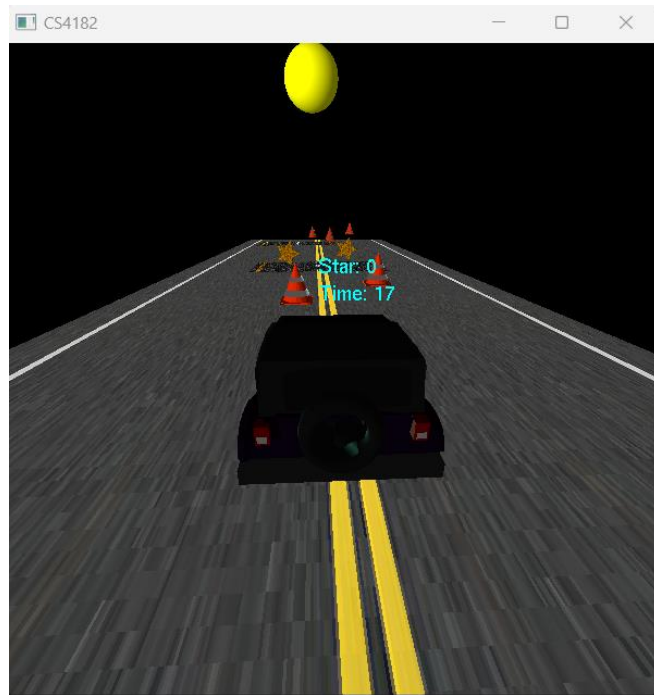
Material sub-menu

In lighting, it has 3 kinds of lighting: GL\_LIGHT0 as the spotlight, GL\_LIGHT1 as directional light, GL\_LIGHT1 and GL\_LIGHT2 as a point light. Based on each lighting, has a different setting. For the spotlight, use the OpenGL function: GL\_SPOT\_CUTOFF, GL\_SPOT\_EXPONENT. Set some angle of the light, so it can look like a spotlight. In directional light, I set the light only to come from the front direction. For the point light, I set it to come from the front direction and back direction. In coding, you can find the function in main.py: display(). The following is the game screen:



Spotlight only lighting within the angle object





Directional light from the front direction

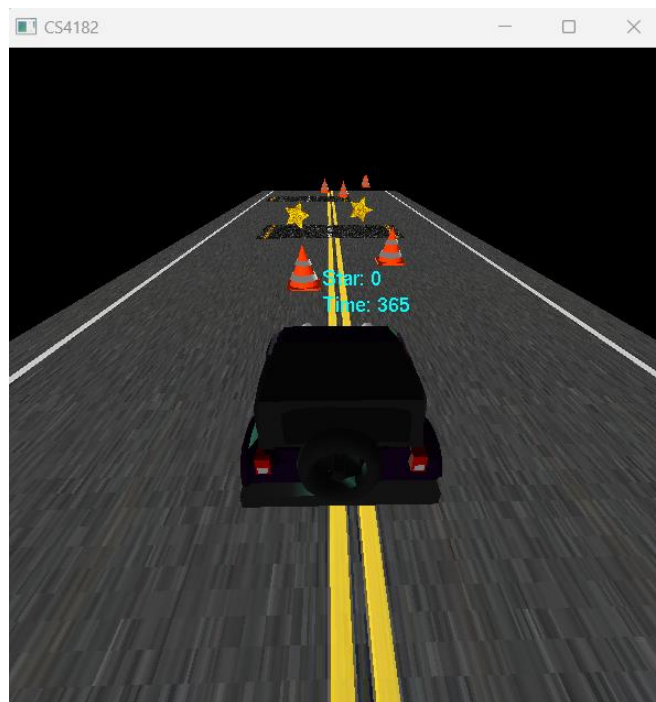


point light from the front direction and back direction

In the ambient, diffuse, specular setting, it has 3 kinds of material color. In coding, you can find the function in main.py: `lightingColorMenuList(value)`. The following is the game screen:



Blue light



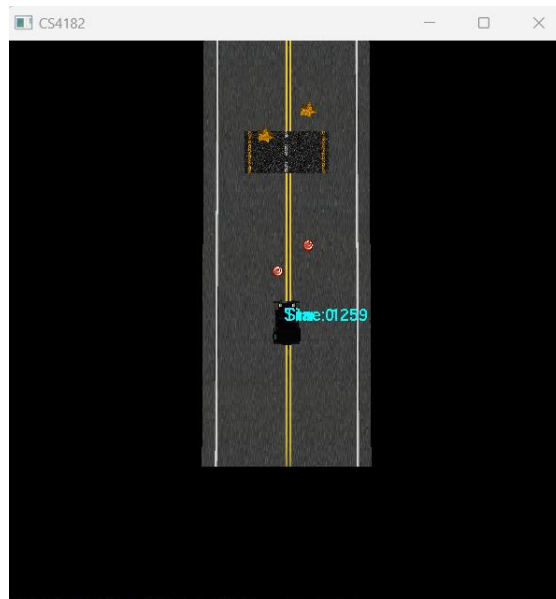
White light



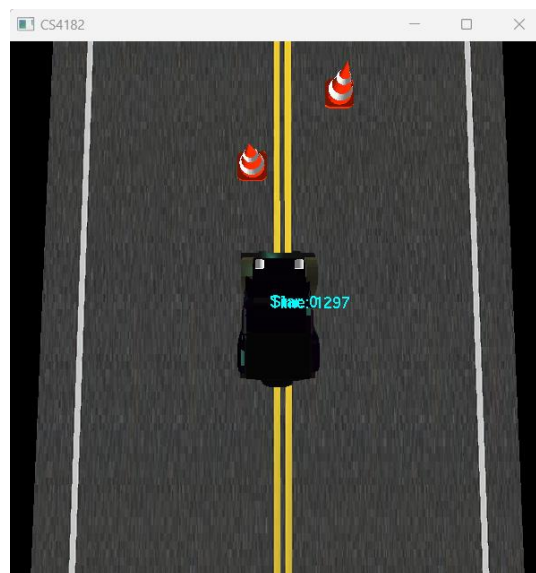
Yellow light

### 3. Manipulation

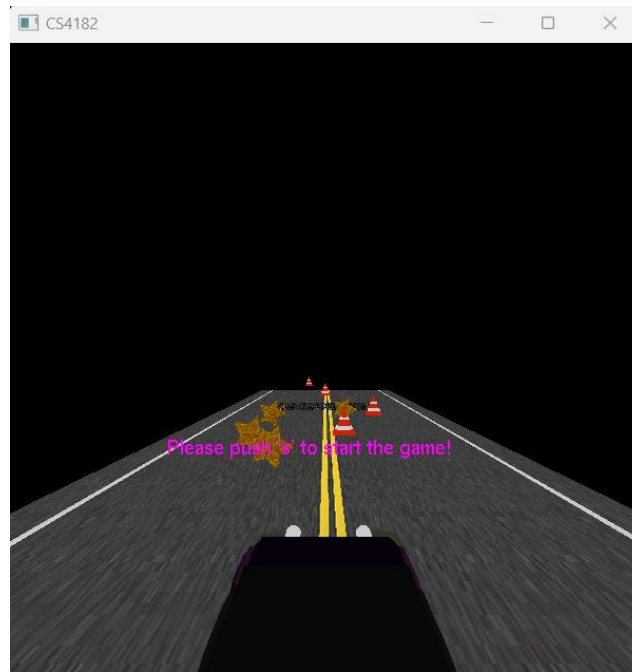
Users can use the keyboard 'left' and 'right' to move the car. Also, can select different viewpoints (top view, behind view, camera view) in the menu. If the user selected camera view in the viewpoint setting, then the user can hold the mouse and right-click moving the viewpoint in a 3D way. In addition, it's able to use the mouse scroll to zoom in or zoom out. In coding, you can find the function in main.py: `viewPointMenuList(value)`, `specialKeys(keypress, mX, mY)`, `mouseScroll(button, scroll, x, y)`, `motionHandle(x,y)`. The following is the game screen:



Top view



Top view with zoom in



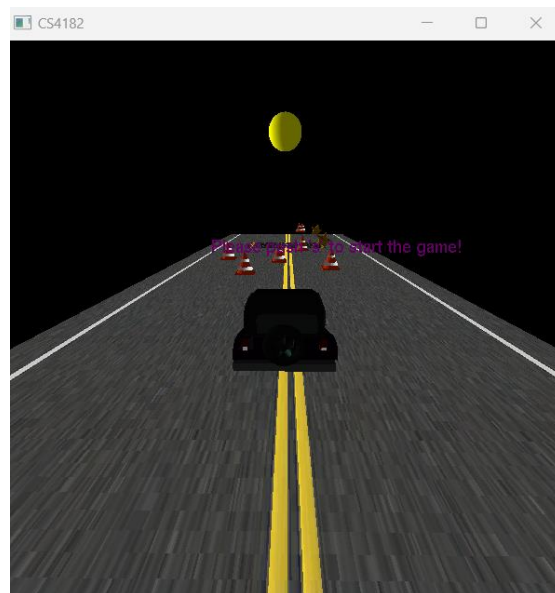
behind view



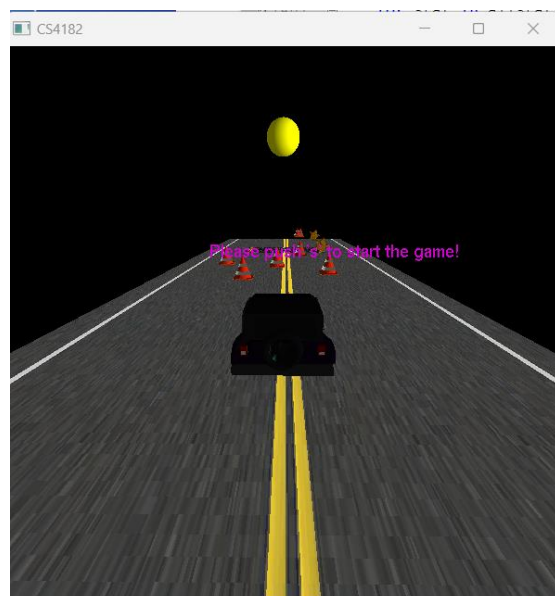
Camera view with zoom out

#### 4. Adding autonomous objects

In the program, I added a sun object. If you selected directional light in the lighting menu, you will see a sun object at the top. And this sun object self-rotates by the y-axis, move around automatically and reacts to the environment. In coding, you can find the function in main.py: display(), idle(). The following is the game screen:



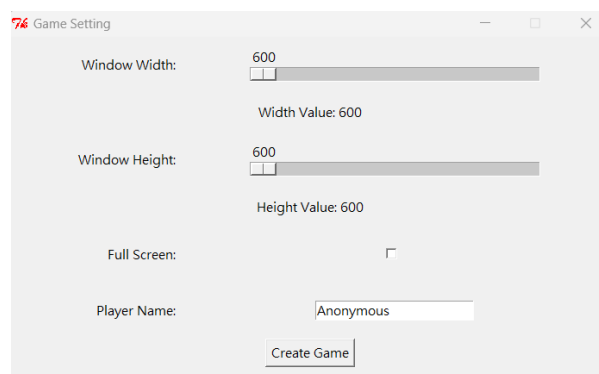
Sunlight goes back



Sunlight goes the front

## 5. Window resolution

Once run the program, the system will open a new window for setting resolution. I use an extra library called tkinter to support my design. The pop-up box required the user to use the slider to select the width and height. And the slider set the max value not larger than the current window size. Also, it has a checkbox the track is full screen and a text box input for the player's name as a ranking record. In coding, you can find the function in main.py: gameConfig() and GameSetting.py. The following is the game screen:



pop-up box



873x600 resolution

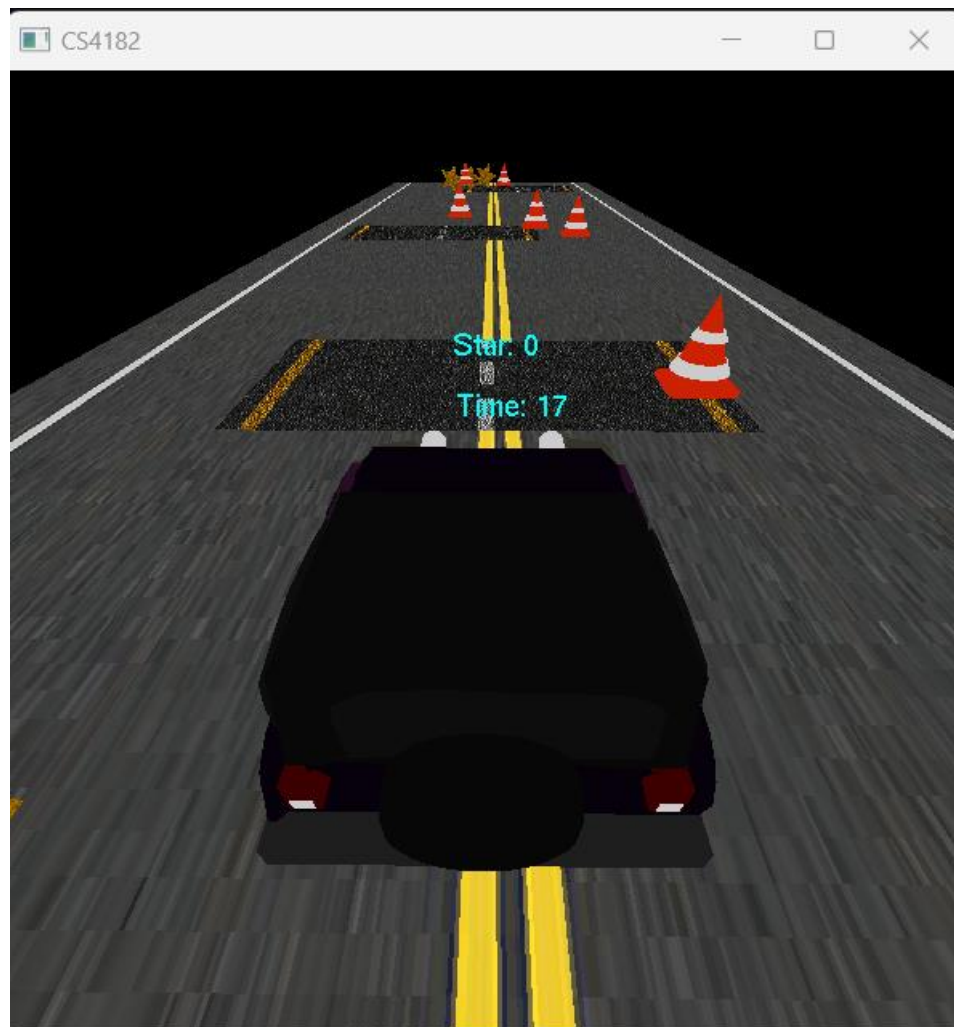


Full Screen



## 6. Accelerating ribbon

The system will set 4 accelerating ribbons on the road at a fixed location, and the jeep can be accelerated after passing the ribbon. And display it with loaded texture `'/img/road.png'`. When the jeep passing, the jeep speed will  $+0.2$  speed. In coding, you can find the function in `main.py`: `collisionCheck()` and `drawAcceleratingRibbon(self)`. The following is the game screen:



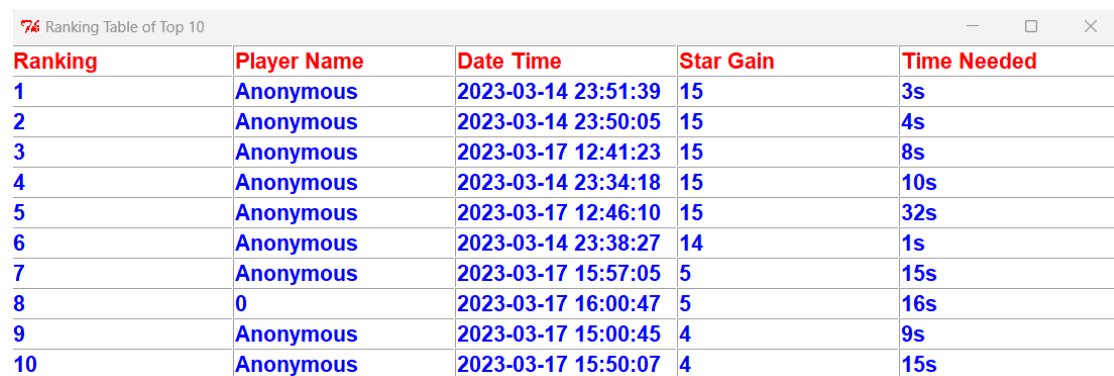
accelerating ribbon



## Advanced Requirements

### 1. Scouring and ranking system

The system will be counting players use how many time to finish the game and get how many stars. When the game end, if the player can get the better result of the current top 10 ranking it will record the result. The ranking rule is first to sort star gain, then sort the time needed. In coding, you can find the function in main.py: recordGame() displayRanking() and RankingTable.py. The following is the game screen:



Ranking	Player Name	Date Time	Star Gain	Time Needed
1	Anonymous	2023-03-14 23:51:39	15	3s
2	Anonymous	2023-03-14 23:50:05	15	4s
3	Anonymous	2023-03-17 12:41:23	15	8s
4	Anonymous	2023-03-14 23:34:18	15	10s
5	Anonymous	2023-03-17 12:46:10	15	32s
6	Anonymous	2023-03-14 23:38:27	14	1s
7	Anonymous	2023-03-17 15:57:05	5	15s
8	0	2023-03-17 16:00:47	5	16s
9	Anonymous	2023-03-17 15:00:45	4	9s
10	Anonymous	2023-03-17 15:50:07	4	15s

Ranking Table of Top 10 result