Soal
Case

Important Notes :

1. **Please use Struct to store the data in this case.**
2. **Logo, table format, or all things for decoration purposes are not graded.**
3. **Please do the main features first (Don’t be stuck doing the validations and not finishing other features).**
4. **Feel free to asks us (AW/LI) if there are any questions.**
5. **Good Luck, Have Fun 😊.**

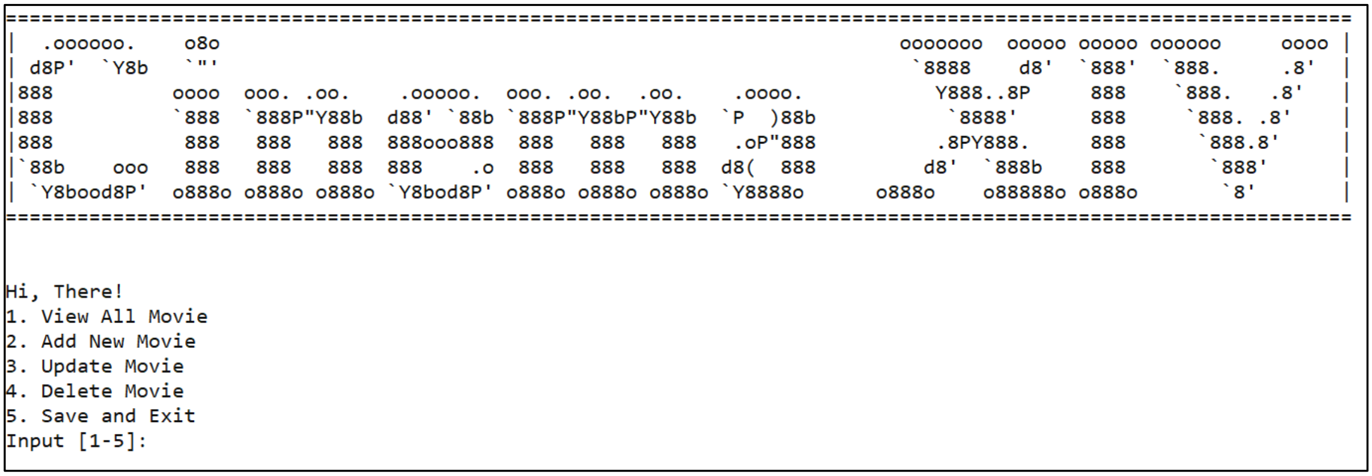
Cinema XIV

Cinema XIV is one of the busiest movie theater in Jakarta since it was opened in 2020. In order to expand the business, **Cinema XIV** has hired you to create a program to manage the movies that are going to be played every day.

- When the program is started, the program will read previously saved data from **cinema.csv** and add the movie data into the **array.** The data inside is sorted and the file will be in the following format:

  [Movie ID],[Movie Name],[Movie Start Time],[Movie End Time],[Movie Studio]

  - The file is a **Comma-Separated Values** file, which means that the data of each movie will be separated by a **comma**.
  - For more details, please open **cinema.csv** provided with this file.
- The program consists of **5 menus**:
  1. View All Movie
  2. Add New Movie
  3. Update Movie
  4. Delete Movie
  5. Save and Exit
- The program will display the main menu and ask the user to choose between the options, which must be **between 1** and **5**.

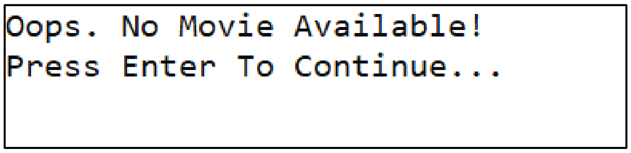


```
=================================================================================================================
|  .oooooo.    o8o                                                         ooooooo  ooooo ooooo oooooo     oooo |
| d8P'  `Y8b   `"'                                                          `8888    d8'  `888'  `888.     .8'  |
|888          oooo  ooo. .oo.    .ooooo.  ooo. .oo.  .oo.    .oooo.           Y888..8P     888    `888.   .8'   |
|888          `888  `888P"Y88b  d88' `88b `888P"Y88bP"Y88b  `P  )88b           `8888'      888     `888. .8'    |
|888           888   888   888  888ooo888  888   888   888   .oP"888          .8PY888.     888      `888.8'     |
|`88b    ooo   888   888   888  888    .o  888   888   888  d8(  888         d8'  `888b    888       `888'      |
| `Y8bood8P'  o888o o888o o888o `Y8bod8P' o888o o888o o888o `Y8888o      o888o    o88888o o888o       `8'       |
=================================================================================================================


Hi, There!
1. View All Movie
2. Add New Movie
3. Update Movie
4. Delete Movie
5. Save and Exit
Input [1-5]:
```

Figure 1. Main menu.

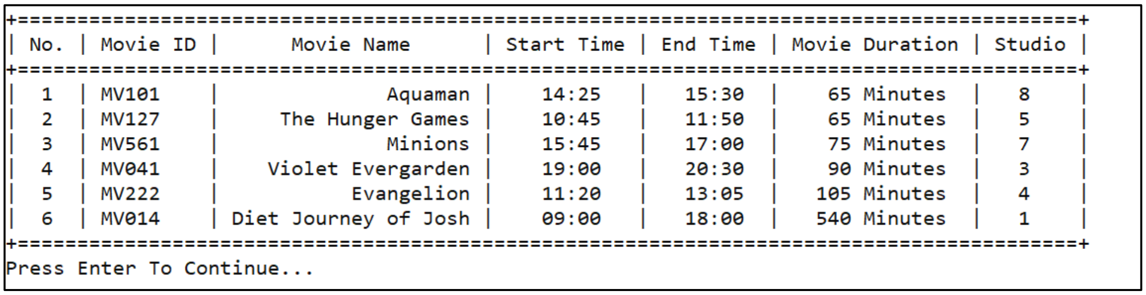
- If the user chooses **menu 1** (‘**View All Movie**’), then:
  - Check whether there is any movie in the **array**. If there is **none**, display an **error message**.



```
Oops. No Movie Available!
Press Enter To Continue...
```

Figure 2. No Movie Error Message

  - Else, display all movie(s), **sorted** using either **Merge Sort** or **Quicksort** in **Ascending** Order. The movie(s) should be sorted by the **movie duration**.
  - The **Movie Duration** is the result of the calculation of duration from **Movie Start Time** to **Movie End Time** in **minutes**. For example : 19.00 – 21.00 = 120 Minutes.



```
+=========================================================================================+
| No. | Movie ID |      Movie Name      | Start Time | End Time | Movie Duration | Studio |
+=========================================================================================+
|  1  | MV101    |              Aquaman |    14:25   |   15:30  |     65 Minutes  |   8    |
|  2  | MV127    |     The Hunger Games |    10:45   |   11:50  |     65 Minutes  |   5    |
|  3  | MV561    |              Minions |    15:45   |   17:00  |     75 Minutes  |   7    |
|  4  | MV041    |    Violet Evergarden |    19:00   |   20:30  |     90 Minutes  |   3    |
|  5  | MV222    |           Evangelion |    11:20   |   13:05  |    105 Minutes  |   4    |
|  6  | MV014    | Diet Journey of Josh |    09:00   |   18:00  |    540 Minutes  |   1    |
+=========================================================================================+
Press Enter To Continue...
```

Figure 3. Display All Movies

- If the user chooses **menu 2** (‘**Add New Movie**’), then:
  - Ask the user to input the following :

- Movie Name

  Movie Name should be between **1-20 Characters.**

- Movie Start Time

  Movie Start Time must be validated according to the following time format.

  HH:MM

  HH = Two Digits of Number Representing the hour of time in 24-Hour Format (00-23) e.g. 21
  MM = Two Digits of Number Representing the minutes of time (00-59) e.g. 47
  Example : 19:14

- Movie End Time

  Movie End Time must be validated according to the following time format.

  HH:MM

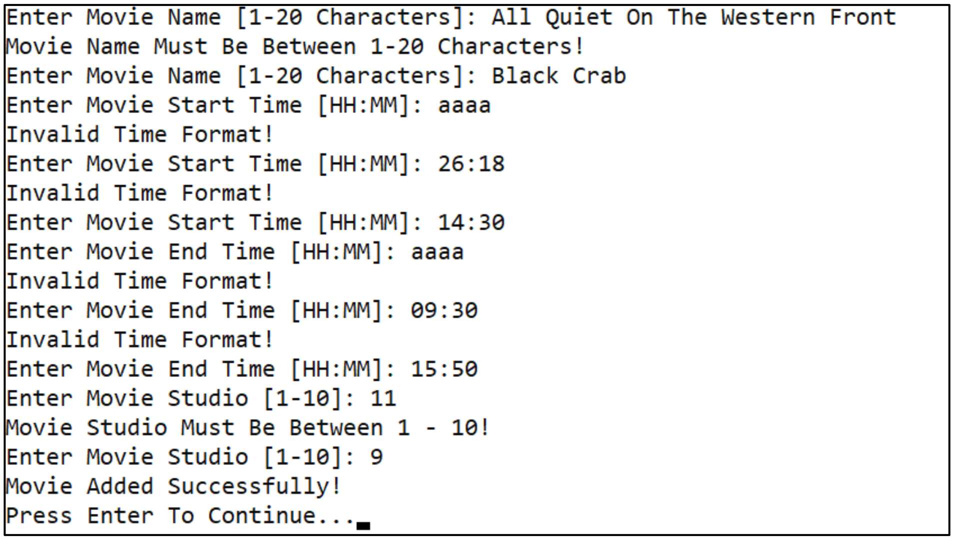
  HH = Two Digits of Number Representing the hour of time in 24-Hour Format (00-23) e.g. 21
  MM = Two Digits of Number Representing the minutes of time (00-59) e.g. 47
  Example : 21:14

  Movie End Time should also be **after the Start Time** and must be **in the same day**.

- Movie Studio

  Movie Studio must be between **1-10.**

- Each **invalid input** should be responded with **error message related to the inputs.**



```
Enter Movie Name [1-20 Characters]: All Quiet On The Western Front
Movie Name Must Be Between 1-20 Characters!
Enter Movie Name [1-20 Characters]: Black Crab
Enter Movie Start Time [HH:MM]: aaaa
Invalid Time Format!
Enter Movie Start Time [HH:MM]: 26:18
Invalid Time Format!
Enter Movie Start Time [HH:MM]: 14:30
Enter Movie End Time [HH:MM]: aaaa
Invalid Time Format!
Enter Movie End Time [HH:MM]: 09:30
Invalid Time Format!
Enter Movie End Time [HH:MM]: 15:50
Enter Movie Studio [1-10]: 11
Movie Studio Must Be Between 1 - 10!
Enter Movie Studio [1-10]: 9
Movie Added Successfully!
Press Enter To Continue...
```

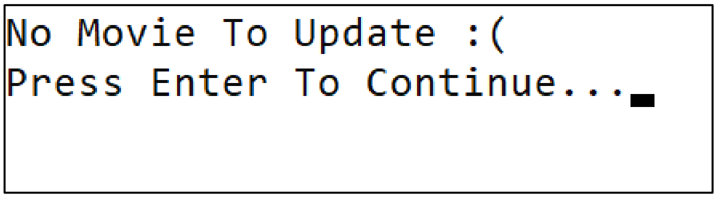
Figure 4. Add Movie Inputs

  - Then, **generate an ID** for the movie with the following format:

[MVXXX]
X : Random Number Between 0-9

  - Upon successful inputs, **add the new movie with all its details to the array**.

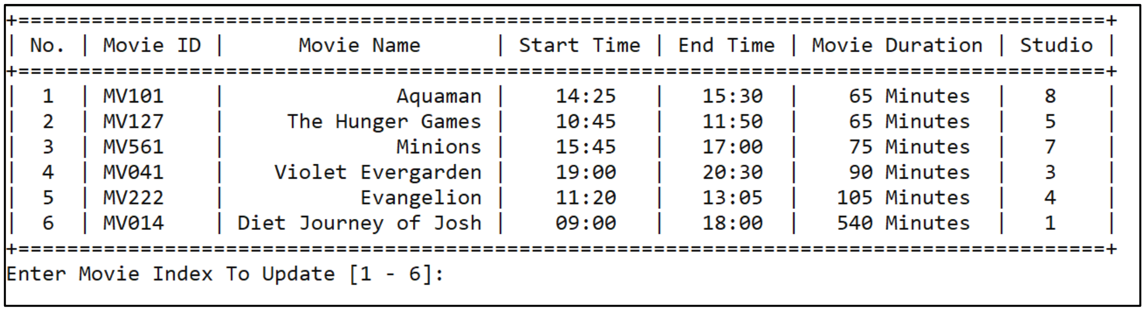
- If the user chooses **menu 3** (‘**Update Movie**’), then:
  - Check whether there is any movie in the **array.** If there is none, display an **error message.**



```
No Movie To Update :(
Press Enter To Continue...
```

Figure 5. Empty Error Message on Update

  - Else, display all movie(s) **sorted** using **Merge Sort** or **Quicksort** in **Descending** Order. The movie must be sorted by the **Movie Duration**.
  - Then, asks the user to input the **movie index** to update. Validate the index must **exists**.



```
+=========================================================================================+
| No. | Movie ID |      Movie Name      | Start Time | End Time | Movie Duration | Studio |
+=========================================================================================+
|  1  | MV101    |              Aquaman |    14:25   |   15:30  |    65 Minutes  |    8   |
|  2  | MV127    |     The Hunger Games |    10:45   |   11:50  |    65 Minutes  |    5   |
|  3  | MV561    |              Minions |    15:45   |   17:00  |    75 Minutes  |    7   |
|  4  | MV041    |    Violet Evergarden |    19:00   |   20:30  |    90 Minutes  |    3   |
|  5  | MV222    |           Evangelion |    11:20   |   13:05  |   105 Minutes  |    4   |
|  6  | MV014    | Diet Journey of Josh |    09:00   |   18:00  |   540 Minutes  |    1   |
+=========================================================================================+
Enter Movie Index To Update [1 - 6]:
```

Figure 6. Update Movie (Movie Exists)

- Next, asks the user to input **new details** of the movie, such as:
  - Movie Name

    Movie Name should be between **1-20 Characters.**
  - Movie Start Time

    Movie Start Time must be validated according to the following time format.

    HH:MM

    HH = Two Digits of Number Representing the hour of time in 24-Hour Format (00-23) e.g. 21
    MM = Two Digits of Number Representing the minutes of time (00-59) e.g. 47
    Example : 19:14
  - Movie End Time

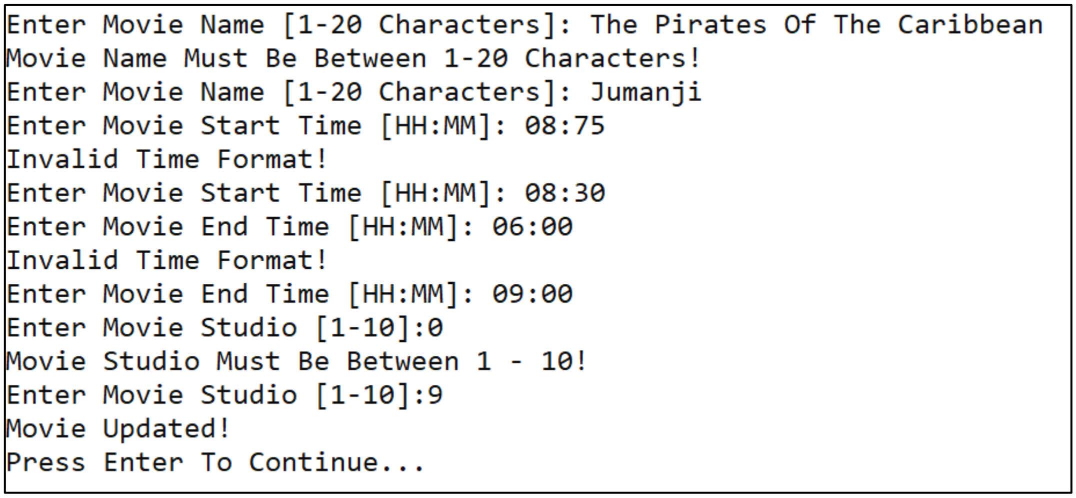
    Movie End Time must be validated according to the following time format.

    HH:MM

    HH = Two Digits of Number Representing the hour of time in 24-Hour Format (00-23) e.g. 21
    MM = Two Digits of Number Representing the minutes of time (00-59) e.g. 47
    Example : 21:14

    Movie End Time should also be **after the Start Time** and must be **in the same day**.
  - Movie Studio

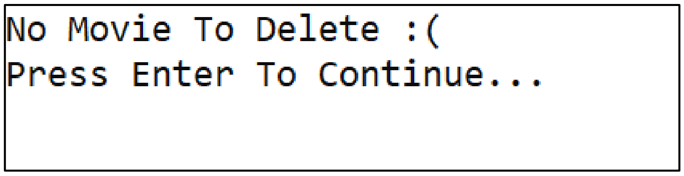
    Movie Studio must be between **1-10.**
- Each **invalid input** should be responded with **error message related to the input.**
- Last, **update the selected movie** with the new details.



```
Enter Movie Name [1-20 Characters]: The Pirates Of The Caribbean
Movie Name Must Be Between 1-20 Characters!
Enter Movie Name [1-20 Characters]: Jumanji
Enter Movie Start Time [HH:MM]: 08:75
Invalid Time Format!
Enter Movie Start Time [HH:MM]: 08:30
Enter Movie End Time [HH:MM]: 06:00
Invalid Time Format!
Enter Movie End Time [HH:MM]: 09:00
Enter Movie Studio [1-10]:0
Movie Studio Must Be Between 1 - 10!
Enter Movie Studio [1-10]:9
Movie Updated!
Press Enter To Continue...
```

Figure 7. Update Movie Input

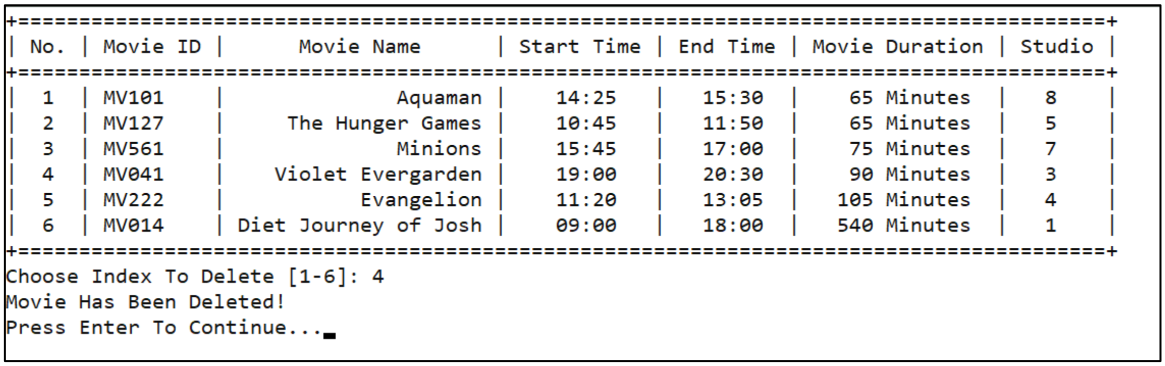
- If the user chooses **menu 4** (‘**Delete Movie**), then:
  - Check whether there is any movie in the **array.** If there is none, display an **error message.**



```
No Movie To Delete :(
Press Enter To Continue...
```

Figure 8. Error Message on Delete

  - Else, display all movie(s) **sorted** using **Merge Sort** or **Quicksort** in **Descending** Order. The movie must be sorted by the **Movie Duration**.
  - Then, asks the user to input the **movie index** to update. Validate the index must **exists**.
  - Last, remove the selected movie from the **array.**



```
+=========================================================================================+
| No. | Movie ID |      Movie Name      | Start Time | End Time | Movie Duration | Studio |
+=========================================================================================+
|  1  | MV101    |              Aquaman |    14:25   |   15:30  |    65 Minutes  |   8    |
|  2  | MV127    |     The Hunger Games |    10:45   |   11:50  |    65 Minutes  |   5    |
|  3  | MV561    |              Minions |    15:45   |   17:00  |    75 Minutes  |   7    |
|  4  | MV041    |    Violet Evergarden |    19:00   |   20:30  |    90 Minutes  |   3    |
|  5  | MV222    |           Evangelion |    11:20   |   13:05  |   105 Minutes  |   4    |
|  6  | MV014    | Diet Journey of Josh |    09:00   |   18:00  |   540 Minutes  |   1    |
+=========================================================================================+
Choose Index To Delete [1-6]: 4
Movie Has Been Deleted!
Press Enter To Continue...
```

Figure 9. Delete Movie

- If the user chooses **menu 5** (‘**Save & Exit**’), then:

- Check whether there are any data on the **array.** If there is any data, save all **sorted data** to a file called **cinema.csv** using the following format:

  [Movie ID],[Movie Name],[Movie Start Time],[Movie End Time],[Movie Studio]

- Last, **terminate** the program.

Run the program for more details!