Project

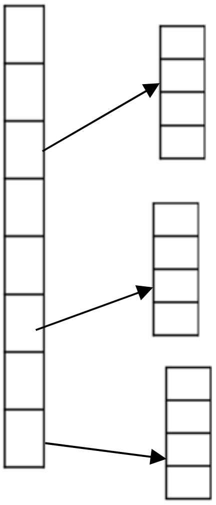
Instructions:

1. This is a teamwork project, each team should be maximum **four** students.

- The work should be your own. All submissions will be checked for originality using a software for that. **High similarity of your code with other submissions will be considered a plagiarism attempt, you will lose all marks of this assignment and the case will be reported to the department, so do not share your work with others.**

2. No grade without presenting your work.
3. Only one student should submit the work, put team names as comments in the beginning of the class that has the main method.
4. The deadline is 2/12/2023. You need to submit your code as exported project on Blackboard, emails will not be accepted.

As a data structures expert, you are required to design a new data structure (as ADT) that **combines two data structures** in order to optimize the a web browser's history mechanism. The figure below illustrates the new data structure, let's call it **"StackHashTable"**. It is composed of a **hash table** of **stacks**, where the record in hash table includes date of visit, and a reference to a stack. The stack composed of records, where each recod holds the data of a specific website (url, website title, visit time)



In the context of a web browser's history mechanism, the hash table is primarily used to store key-value pairs, where the date visit of a web page is the key, and the associated web page data is the value. The information to be saved in the hash table typically includes:

1. Visit Date (Key): The date when the page was visited is used as the key in the hash table. This key is essential for efficient and quick retrieval of the corresponding web page data. Assume the date is in the format "dd-mm-yyyy".

2. Web Page Data (Value): The value associated with each URL key is the web page data. This data may include:

- URL: The URL of the web page.
- website title: the title of the webpage.

- Visit Time: The time when the page was visited, which helps in maintaining the chronological order and tracking the pages visits. The time is in the format “HH:MM:SS”.

To access a specific website, you need to use a hash function that receives date of visit. You need to develop your hash function, you might use Java hashcode method to convert the date (string) to integer, then you can use % to map that to an integer within the table range.

Your data structure should have the following method:

- **addWebsite(String date, String url, String title, String time):** to add a new website’s record.
- **deleteWebsite(String date, String url):** to delete that website on the specific date. If that website not found in that date, it displays an error message.
- **searchWebsite(String url):** to search for the website in all dates. If it is found, it displays the date and time of each visit for that website. If no visit to that site, it displays “This website has been never visited”.
- **displayAllWebsites(String date):** it displays all websites visited on that date (url, title, time).

In the test class, you need to display the following options:
Enter your choice:

1- Add Website.
2- Delete Website.
3- Search Website
4- Print Websites.
5- Exit

- When the user selects 1, your application should ask for date, url, titel,, and time. It will call method addWebsite.
- When the user selects 2, your application should ask for date, and url and call method deleteWebsite.
- When the user selects 3, your application should ask for url and call method searchWebsite
- When the user selects 4, your application should ask for visit date, and call method displayAllWebsite.
- When the user selects 5, your application should save the data structure object to a file (Object serialization)

Note: when your application starts, it should search for the file (if it exists) that contains the data structure object and upload it to memory. There will be no file when we execute the application for the first time.

Grading:

- Combined Data Structure Implementation: **8 pts**.
- Tester Application: **2 pts.**
- No grade without presentation.
- The instructor can give the student ZERO in the project if they cannot answer any question in the discussion, which means they did not contribute to the project.