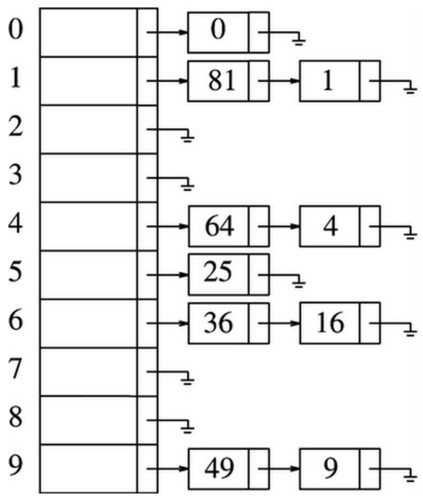
EEM 480 HW4

In this project you are required to realize a hash table using chaining as seen in figure below.



- This hash structure will be used to trace a text file. The program will get a path of a text file written in English. The program will trace each word and keep track the positions of them using the hash tables and their related links. All punctuation marks will be removed. (Ex. The boy, who has green hair is walking down the street. "boy" and "street" has to be isolated from comma or dot)
- Try hash table size of ~500, ~1000, ~10000 Check the number of occurrences of collusion. (I will definitely check)
- After the table is organized, the program ask a user a word and this word is searched through the hash table. If it is found in the text, the program inform user it's place in the text. (If it is more than once and informed its correct positions, you will get extra 5 mark) and its repetations.
- Explain how you obtain key from word.
- Explain your Hash function.

Here the Interface for your HW is given as :

```
public interface HW4_Interface {
    public Integer GetHash(String mystring);
    public void ReadFileandGenerateHash(String filename, int size);
    public void DisplayResult(String Outputfile);
    public void DisplayResult();
    public void DisplayResultOrdered(String Outputfile);
    public int showFrequency(String myword);
    public String showMaxRepeatedWord();
    public int checkWord(String myword);
    public float TestEfficiency();
    public int NumberOfCollusion();

}
```

You have to design and implement **`HW4_Hash`** class which implements **`HW4_Interface`**.

Here the functions and their explanations :

```
    Integer GetHash(String mystring); // generate an integer
value (hash index) related to the input word. If collusion
occurs the collusion has to be solved by chaining.
    void ReadFileandGenerateHash(String filename, int size); //
Create the changing hash where the size given by the user. The file
which contains a very long text will be parsed and during the parsing
hash table must be modified by the words.
    void DisplayResult(String Outputfile); // All the words in the
text and their frequency has to be displayed in a text file.
    void DisplayResultOrdered(String Outputfile); // All the words
and in the text and their frequency has to be displayed in a text file
in an ordered fashion. The most repeated words will be listed at the
beginning and the least repeated words at the end
    void DisplayResult(); // All the words in the text and their
frequency has to be displayed on the screen.
    int showFrequency(String myword); The frequency of myword in the
text file will be given. If there is no myword in the text -1 must be
returned.
    String showMaxRepeatedWord(); // The most repeated word has to
be returned.
    int checkWord(String myword); // Checks whether myword is found
in the text. If found, display its position in text.
    Integer TestEfficiency(); // Returns the efficiency of your hash
method with size of hash.
    int NumberOfCollusion();// Returns the number of collusions
during parsing the file.
```

The important thing about your assignment is the implementation of your hash function. Main duty of the hash function is generating a digestion respect to the input data. In this project, the digestion becomes an index value for your table and the input data is the word that you are indexing. If your hash function is a loose one, the overlapping ratio of the words increase. Then it causes a bad distribution of the indexed words and increases the search time. So we would like to have a hash function such that distributes the input words in a finely manner as much as possible. Briefly, you had better to make a little research on hash functions in order to implement your's a good one.

Example (In text file)

Outside, even through the shut window-pane, the world looked cold. Down in the street little eddies of wind were whirling dust and torn paper into spirals, and though the sun was shining and the sky a harsh blue, there seemed to be no colour in anything, except the posters that were plastered everywhere. The blackmoustachio'd face gazed down from every commanding corner. There was one on the house-front immediately opposite. BIG BROTHER IS WATCHING YOU, the caption said, while the dark eyes looked deep into Winston's own. Down at streetlevel another poster, torn at one corner, flapped fitfully in the wind, alternately covering and uncovering the single word INGSOC. In the far distance a helicopter skimmed down between the roofs, hovered for an instant like a bluebottle, and darted away again with a curving flight. It was the police patrol, snooping into people's windows. The patrols did not matter, however. Only the Thought Police mattered.

Behind Winston's back the voice from the telescreen was still babbling away about pig-iron and the overfulfilment of the Ninth Three-Year Plan. The telescreen received and transmitted simultaneously. Any sound that Winston made, above the level of a very low whisper, would be picked up by it, moreover, so long as he remained within the field of vision which the metal plaque commanded, he could be seen as well as heard. There was of course no way of knowing whether you were being watched at any given moment. How often, or on what system, the Thought Police plugged in on any individual wire was guesswork. It was even conceivable that they watched everybody all the time. But at any rate they could plug in your wire whenever they wanted to. You had to live -- did live, from habit that became instinct -- in the assumption that every sound you made was overheard, and, except in darkness, every movement scrutinized.

Winston kept his back turned to the telescreen. It was safer, though, as he well knew, even a back can be revealing. A kilometre away the Ministry of Truth, his place of work, towered vast and white above the grimy landscape. This, he thought with a sort of vague distaste -- this was London, chief city of Airstrip One, itself the third most populous of the provinces of Oceania. He tried to squeeze out some childhood memory that should tell him whether London had always been quite like this. Were there always these vistas of rotting nineteenth-century houses, their sides shored up with baulks of timber, their windows patched with cardboard and their roofs with corrugated iron, their crazy garden walls sagging in all directions? And the bombed sites where the plaster dust swirled in the air and the willow-herb straggled over the heaps of rubble; and the places where the bombs had cleared a larger patch and there had sprung up sordid colonies of wooden dwellings like chicken-houses? But it was no use, he could not remember: nothing remained of his childhood except a series of bright-lit tableaux occurring against no background and mostly unintelligible.

```
Lutfullah is not found in the text
voice is found in location 5 and number of occurrences is 1
There are 17 collusions occurred.
```

Rules for HW Submission

. You must write your HW in NetBeans environment.
. You must write a report with name "**Report_HW4.pdf**" explaining your HW (purpose, how did you solve it, algorithm etc.) and what you the environment you used (NetBeans, for example). The person who read your report can easily use the class you have written.
. Discuss the result you have obtained.
. Submission should be in the form of a zip/rar. When extracted, the result should be a single folder with the name "HW4".
. Do not forget to put your report into the zip/rar file.
. The name of your project will be "**Name_Surame_HW4**. e.g. *Lutfullah_Arici_HW4*. If you do not obey the rule I will not grade your homework.

- You must bundle your whole project folder into your HW4.zip file.
- If I extract your project file, then import to my environment and if it doesn't work, you will be graded on 30 not 85. (Double check. It saves life)
- Do HW by yourself. Be honest.
- **Don't even think cheating, you get full mark but minus!**