MIDDLE EAST TECHNICAL UNIVERSITY, NORTHERN CYPRUS CAMPUS
CNG315 Algorithms

Assignment 4: Another Simple Scanpath Analyser

This assignment aims to help you practice ***string processing*** and ***string matching.*** Your main task in this assignment is to develop another simple scanpath analyser using **C programming language**.

Overview

Autism Spectrum Disorder (ASD) is a neurodevelopmental disorder characterised by differences in communication and social interaction [1, 2]. Recent research studies have analysed the eye movements of people with and without autism recorded while interacting with web pages, and have found that people with autism tend to have different processing strategies in comparison with people without autism when they interact with web pages [1, 2].

Eye tracking is the process of recording the eye movements of people to understand where they look at, called fixations, and how long these fixations last. The series of their fixations represent their scanpaths. Scanpaths are usually analysed based on the areas of interest (AOIs) of visual stimuli, especially which AOIs are frequently used and in which order. Figure 1 shows an example of a scanpath of a particular person on one web page of Wordpress.com where the red rectangles show the AOIs of the web page, and the yellow circles show the person's fixations - the radius of each circle is directly proportional to the duration of its corresponding fixation.

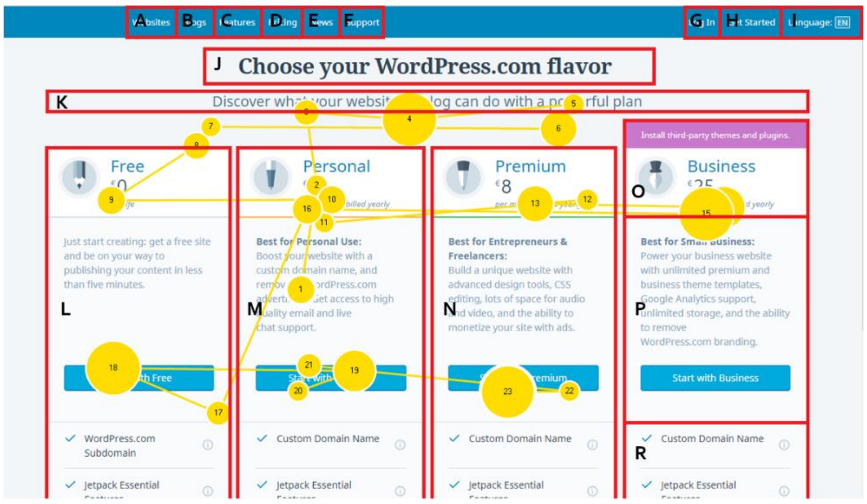

Figure 1. A scanpath of a particular user on one web page of Wordpress.com

To analyse scanpaths based on AOIs, they are first represented as string sequences. For example, if a person looks at the AOIs L, M, P, and R respectively, his/her scanpath is represented as LMPR. In this assignment, you will need to develop another simple scanpath analyser to check whether a given pattern is available in a set of scanpaths of people with and without autism.

This application will start by taking the name of the two files from the user where the first file includes a number of scanpaths for people with autism and another one includes a number of scanpaths for people without autism, which are represented as string sequences based on ten AOIs named with digits from zero to 9. These files may include a different number of scanpaths, and each line in these files represent a different scanpath. You can assume that the maximum length of a scanpath is 100. An example content of a file is given below:

```
6547382712384758123
3547712364687
```

```
847364537292
9374638473682
8162537252
```

This application will then ask the user to enter the pattern to be detected (such as, 123). It will then show the scanpaths that include the pattern for people with and without autism by highlighting the patterns with square brackets. It will also show how many pattern matches are found in how many scanpaths and also the most frequent AOI before the pattern for each group of people. If there are multiple of them, all of them should be printed. A sample run is provided below:

```
Enter the file name for people with autism: peoplewithautism.txt
Enter the file name for people without autism: peoplewithoutautism.txt
Enter the pattern: 123

Detected patterns for people with autism:
1. 65473827[123]84758[123]
2. 35477[123]64687
3 patterns detected in 2 scanpaths
The most frequent AOI before the pattern is/are 7

Detected patterns for people without autism:
None
```

Implementation Requirements

This application will have one linked-list for people with autism and another one for people without autism. In each node of the linked lists, the string representation of a scanpath should be kept. You will need to implement the following functions and call them appropriately in the main function. You can also implement some helper functions.

- **createScanpathList**: This function takes the name of the file containing a number of scanpaths which are represented as string sequences and creates a linked-list where each node contains a string representation of a single scanpath. It will return the created list.
- **searchPattern:** This function takes the pattern and the linked lists of people with and without autism to check whether the pattern is available in the scanpaths of people with and without autism by using the **Rabin-Karp Algorithm** where d=10 (since there are 10 digits) and q=11. If the pattern is detected in a scanpath, it will be highlighted with square brackets as shown above. The general pseudo-code of the **Rabin-Karp Algorithm** is given below.

```
Rabin-Karp(T, P, d, q)
      n := length[T];
      m := length[P];
      h := $d^{m-1}$ mod q;
      p := 0;
      $t_0$ := 0;
      for i := 1 to m do
            p := (dp + P[i]) mod q
            $t_0$ := ($dt_0$ + T[i]) mod q
      for s := 0 to n – m do
            if p = $t_s$ then
                  if P[1..m] = T[s+1..s+m] then
                        print "pattern occurs with shift s"
            if s < n-m then
                  $t_{s+1}$ := (d($t_s$ – T[s+1]h) + T[s+m+1]) mod q
```

As mentioned above, this function will also show how many pattern matches are found in how many scanpaths and also the most frequent AOI before the pattern for each group of people. If there are multiple of them, all of them should be printed.

Incremental and Modular Development

You are expected to follow an incremental development approach. Each time you complete a function or module, you are expected to test it to make sure that it works properly. You are also expected to follow modular programming which means that you are expected to divide your program into a set of meaningful functions.

Professionalism and Ethics

You are expected to complete the assignments on your own. Sharing your work with others, uploading the assignment to online websites to seek solutions, and/or presenting someone else's work as your own work will be considered as cheating.

Rules

- You need to write your program by using **C programming language.**
- You need to create your own data files for testing your program.
- You need to name your file with **your student id**, e.g. 1234567.c, and submit it to ODTUCLASS.
- **Code quality, modularity, efficiency, maintainability, and appropriate comments** will be part of the grading.

Grading Policy

The assignment will be graded as follows:

Grading Item	**Mark (out of 100)**
Data Representation	5
Main Function – coordinate the inputs and function calls	15
createScanpathList	20
searchPattern – Use of Rabin-Karp Algorithm	25
searchPattern – Print scanpaths that includes the pattern	15
searchPattern – Print statistics (the number of pattern matches, the number of scanpaths include the pattern, and the most frequent AOI before the pattern for each group of people)	20

References:

1. Sukru Eraslan, Yeliz Yesilada, Victoria Yaneva and Le An Ha. 2020. "Keep it Simple!" An Eye-tracking Study for Exploring Complexity and Distinguishability of Web Pages for People with Autism. Universal Access in the Information Society (SCI-E, SSCI).
2. Sukru Eraslan, Victoria Yaneva, Yeliz Yesilada and Simon Harper. 2019. Web users with autism: eye tracking evidence for differences, Behaviour & Information Technology, (SCI-E, SSCI) 38, 7, 678-700.