Homework 2: Color Image Retrieval and Downloading/Parsing Web Pages

Check lcm.liacs.nl for the duedate. Each student must submit their own unique solution. Students may ask each other for assistance

We assume that students are capable of writing source code in C and C++. In particular, you should make sure you are knowledgeable on the usage of:

```
stdlib.h
malloc.h
string.h
char
unsigned char
malloc
strcpy
strstr
strcat
'\0'
pointers - e.g. char *Myfunction(char *p, char *q)
```

and using a debugger in Linux, especially for finding wild pointer errors.

All given source code was in general written by the student assistant often using OpenCV source code as standalone projects.

IMAGE RETRIEVAL

(1) Color based Similar Image Retrieval (this is an intro, easy difficulty level)

Go into the **imageretrieval** directory on a linux machine:
Go into the **Debug** directory under imageretrieval
Type the command:

```
make clean
```

Type the command:

```
make
```

Type the command:

```
./imageretrieval ./myimages/index1.jpeg ./myimages/ ranklist.html
```

(a) Open the retrieval result ranklist.html in the Debug folder using a web browser and take a screenshot.

(b) Read the file, maintest.cpp. Download some jpeg images (preferably small to medium size - less than 800x800 pixels) and try out some more color queries. Evaluate scientifically - describe when it works and does not?

(2) Downloading and Parsing Weblinks (low-medium difficulty level)

Building a web search engine fundamentally involves downloading and parsing the webpage for weblinks (e.g. <a href="http://www.liacs.nl">LIACS</a>). In principle, one can then use depth-first search or breadth-first search to crawl the web network using the weblinks. This assignment introduces several *basic components* (which are intended to be improved later on) in the C programming language.

In this homework we give you C source code based on well known libraries which

(i) Downloads a web page
and
(ii) Extracts weblinks from the web page

The source code has been tested on Linux on the computers in room 302/303 and comes with a demo program which does (i) and (ii) for a URL supplied at the command line.

You will be making the functions a bit more general. Please read this file: *Tutorial.haut.www.part1.pdf* which contains an overview and compile commands.

Note that there are two websearch directories: websearch.classic and websearch.new

In the directory **websearch.new** is new parsing software called **Haut HTML-parser,** which was developed by a LIACS student, Micky Faas. *For problems (2) and (3) you should use Haut HTML-parser.*

In the directory, **websearch.classic**, is the old parsing software from Google called **htmlstreamparser** that had been used years ago - it is not necessary to use it.

(a) The goal is to write two functions based on the tutorial C code (you are supposed to modify the supplied source code) which takes as input a web URL, downloads the webpage, and outputs the list of weblinks. The functions should be as follows:

char *GetWebPage(char *myurl)
- this returns a string containing the html from the remote webpage at myurl
- if unsuccessful, it will return NULL

char *GetLinksFromWebPage(char *myhtmlpage, char *myurl)
- this extracts the weblinks from the string myhtmlpage (i.e. http://www.liacs.nl)
- in this case we pass myurl only for the original webpage name for use in creating the absolute links below
- please return the absolute links like http://www.liacs.nl/edu instead of just /edu
- The returned string should have each weblink separated by the newline characters: '\n'
- If it does not find any weblinks then it should return NULL

You should also write a command line program which uses the above functions and prints the links.
It is important to print the absolute links like "http://www.liacs.nl/edu" instead of just "/edu" from myurl :

Tip: If the relative link has a slash ('/') then its relative to the root directory (e.g. https://www.example.com), otherwise its relative to the current directory (e.g. https://www.example.com/example/dir/)
There is also the option of '../' which as expected takes you to the parent directory.

```
> showlinks http://www.liacs.nl
```

In the "main" function, it should simply pass the url from the command line to the functions you wrote and print the weblinks.

(b) Parsing Image Links (low-medium difficulty level)

Write another function which only extracts the image links that start with <img> called

char *GetImageLinksFromWebPage (char *myhtmlpage, char *myurl)

(3) HTML Parser Shootout

(a) Download using either your own source code or a browser these html webpages:
- https://liacs.leidenuniv.nl/
- https://www.universiteitleiden.nl/
- https://news.google.com/
- https://www.imdb.com/
- https://io9.gizmodo.com/
- https://www.tomshardware.com/
- https://www.amazon.com/
- https://www.msn.com/nl-nl/
- https://www.cnet.com/
- https://en.wikipedia.org/wiki/Main_Page

(b) Write code to load the webpages into RAM memory and parse each of them X times. So, if X=1000 then 10,000 webpages will have been parsed. The general idea is that you want to set X to be a number that allows you to easily estimate the amount of time that it takes on your computer. In C, you can use the *clock()* command for timing.

(c) For a different parser (you can use the one in websearch.classic if you wish but it would be better to use one from your favorite language such as Python or Java or ...), do the same process as in **(b)** with the same X and compare the speeds. *The general intention is to do a fair comparison between the two parsers. Comment on the* *strengths and weaknesses of each parser. For example, how many links does each parser find and how much memory does each take?*

Submission - please see LCM.liacs.nl for the duedate

For problems (1), (2) and (3), you should turn in a zip file on the LML course manager containing

- Your C source code which should compile on machines 302/303.
- a summary file called: **Journal.pdf**

The file "**Journal.pdf**" should include and address

(i) Your name and student ID
(ii) Which problems did you get working?
(iii) Did the functions work as expected?

and your answers to the problems and screenshots for

1(a) at least provide screenshot
1(b) give your comments here
2(a) at least provide screenshot
2(b) at least provide screenshot
3(c) describe what you did, which parsers were used and your comments and the results here