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

Check tron.liacs.nl for the duedate. Every student must submit their own solution.

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?

WEB SEARCH ENGINES

Building a web search engine fundamentally involves downloading and parsing the webpage for weblinks. 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.
Go into the **websearch** directory and please refer to the document: "Tutorial.www.part1.pdf"

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

(a) The goal is to write two functions based on the tutorial C 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
Please print the absolute links like "http://www.liacs.nl/edu" instead of just "/edu" from myurl :

> 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)

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** which contains your answers to the problems and screenshots for
  1(a)
  1(b)
  2(a)
  2(b)

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

(i) Your answers to problems 1 and 2
(ii) Which problems did you get working?
(iii) Did the functions work as expected?
(iv) Screen shot(s) showing your program working for each problem.