McMaster University
Department of Computing and Software
Dr. W. Kahl

SFWR ENG 2S03
Exercise Sheet 10

SFWR ENG 2S03 — Principles of Programming

15 November 2006

Exercise 10.1 — Association Lists — Web Spider (55% of Final, 2005)

The following C type definitions will be used to define "**URL sets**" implemented as **ordered linked lists**:

```
typedef struct UrlListNodeStruct { const char * url;
                                   struct UrlListNodeStruct * next; } UrlListNode;

typedef UrlListNode * UrlSet;
```

Building on this, the following C type definitions will be used to record the link structure of set of web pages, again implemented as an **ordered linked list**, ordered define "**URL sets**" represented as ordered lists:

```
typedef struct PageListNodeStruct { const char * pageURL;
                                    UrlSet       links;
                                    struct PageListNodeStruct * next; } PageListNode;
typedef PageListNode * PageSet;
```

The considered price lists will satisfy the following **invariants**:

(1) None of the char * fields is *NULL*.

(2) Each char * field points to a zero-terminated string.

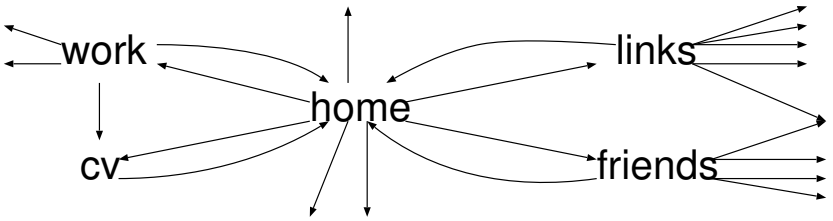
(3) Each char * field has been obtained by *malloc*().

(4) All lists are **ordered** with respect to their char * fields and the ordering implemented by the following library function:

> int *strcmp*(const char *s1, const char *s2);
>
> The *strcmp*() function compares the two strings *s1* and *s2*. It returns an integer less than, equal to, or greater than zero if *s1* is found, respectively, to be less than, to match, or be greater than *s2*.

(5) In every list, no two of the char * fields contain matching (i.e., equal) strings.

A typical personal web site might have the following link structure, where arrows without named targets are **external** links:



(The solutions to the items (a,b) and (c) are ***independent of each other!***)

(a) ≈11% Assuming — for simplicity — that each URL has 30 characters, we are interested in the memory demands for a 5-page *PageSet* representing the personal web site drawn above, on a 32-bit machine. (I.e., how many bytes will the whole *PageSet*, including all list nodes and strings, occupy?)

Different representations with different memory requirements are possible; identify and explain two different possibilities, and calculate the memory demands for each.

Clearly state your assumptions and document your reasoning.

(You do not have to calculate the results of multiplications if these results are greater than 400.)

(b) ≈6% What are the consequences of the two alternatives you identified in (a) above for programming, for example an *insertPage* or a *deletePage* function?

(Do not program these functions here; just explain what differences between the two possibilities would be.)

(c) ≈38% **Design and implement** a C function with the following prototype:

UrlSet findExternal(*PageSet*);

Passed a *PageSet* as argument, this shoud create a *UrlSet* containing exaclty the external links referenced as *url* inside the argument *PageSet*, but not as *pageURL*. (Note (5) above!)

Decompose into sub-functions as appropriate, and describe the interface and specification of each sub-function.