CS 60
Programming Assignment #4
Due Date: April 30, 2010 (11:00 AM). 80 Points.
Remember: Points will be deducted for homework turned in after the Due Date.
Deadline: May 3, 2010 (11:00 AM)
Remember: No homework will be accepted after the Deadline
You must work on this assignment independently.

For this assignment you will write a C program to manipulate a (mini Google) multilinked structure. The problem is to keep a set of web pages each of which contains a set of index terms. To facilitate the assignment each web page will be represented by a positive integer (`int`). Also each index term will be a positive integer (`int`). Each web page contains exactly two index terms.

Each web page will be represented by an instance of the following `struct webpage`. The web pages will be in increasing order (with respect to the web page number) in a singly-linked list using the pointer `nextwp` in `struct webpage`.

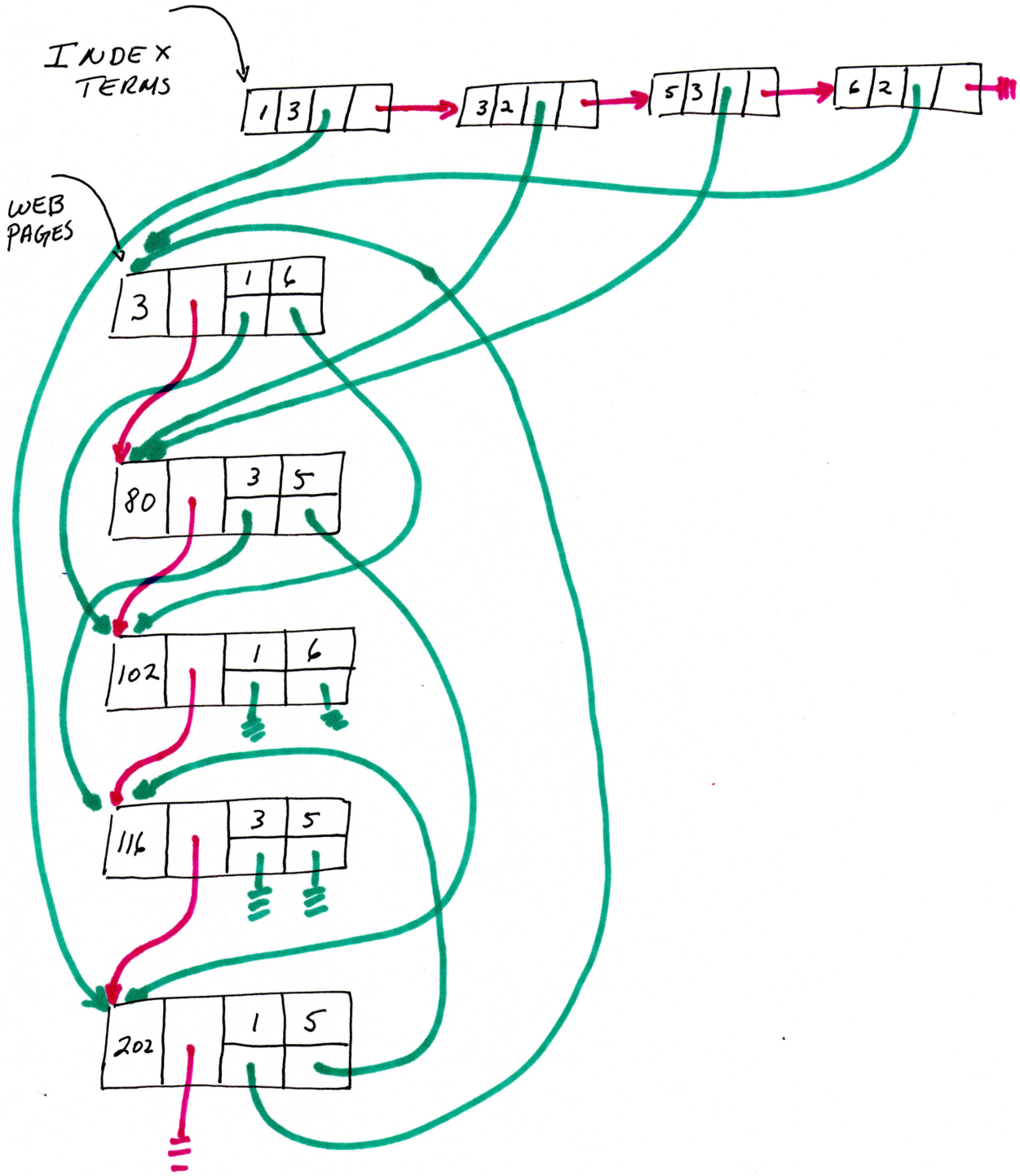
```
typedef struct webpage {
    int              wpnumber;   // Web page id
    struct webpage*  nextwp;     // Pointer to web page
    struct itarray        {
          int             indexterm;  // Indexterm
          struct webpage* wplink;     // Pointer to web page
      } itl[2];
  } Webpage;
```

Each index term will be represented by an instance of the following `struct indexterm`. We will also maintain this in increasing order (with respect to the index term number) in a singly-linked list using the pointer `nextit` in `struct indexterm`.

```
typedef struct indexterm {
    int                itnumber;       //Index term id
    int                numberoftimes;  //# of web pages that include itnumber
    Webpage*           link2webpage;   //Pointer to web page
    struct indexterm* nextit;          //Pointer to index term
  } Indexterms;
```

In addition there will be one singly linked list for each index term. Consider index term 150. The singly linked list for index term 150 starts at the instance of the `struct indexterm` that represents index term 150 and it uses the pointer `link2webpage`. This takes us to a web page (lets say it is web page 200) that uses index term 150. In the instance of `struct webpage` for web page 200, one of the two entries in the `itl` array will have the `indexterm` equal to 150. Then in the corresponding entry `wplink` will either be Null (which means the list ends there) or it will point to an instance of a `struct webpage` that represents another web page. The remaining web pages that use the index term 150 are found by continuing the same search process just described.

Below you will find an instance of our data structure.


INDEX TERMS
1 3
3 2
5 3
6 2
WEB PAGES
3
1 6
80
3 5
102
1 6
116
3 5
202
1 5

In addition your program will have two variables, one containing the number of web pages in the system and the other the number of index terms in the system. The code given in the `map.html` web page will help you get started. You may use any part(s) of that code in your homework.

Your main program will read in a set of commands to manipulate the set of web pages. In the map.html web page you will find a skeleton code that shows how to read in the data (you may use this code, but note that you need to add a few things to make it work). You will also find in map.html page a sample input and output file. **YOUR PROGRAM MUST MIMIC THE OUTPUT LINES IN THE SAMPLE OUTPUT FILE. YOU MAY LOSE POINTS IF YOU DO NOT FOLLOW THE PRINTING PATTERN IN THE SAMPLE OUTPUT FILE.** You should write a function that will initialize the lists to empty. The command lines are given below. You must print every command line just after you read it. By `m, i1,` or `i2` below we mean that you should expect a positive integer.

- `i m i1 i2`

  You may assume that `i1` is different from `i2`. The value of `i1` may be larger or smaller than `i2`. This indicates that there is a new web page `m` with two index terms denoted by `i1` and `i2`. If the web page already exists, then the operation will do nothing. Otherwise a new instance of `struct webpage` must be created and it should be added to three lists (the list of web pages and two lists of index terms). Note that if there was not a list for either or both of the index terms, they should be created at this point too.

- `d m`

  This indicates that you must delete the web page `m`. If `m` does not exist, then your program should not make any changes and nothing should be printed. When the page exists, the instance of `struct webpage` representing webpage `m` should be deleted from the list of web pages and from the two lists of index terms used by the web page. If either or both index term lists become empty (i.e., the index term(s) is not used in any remaining web page), they should also be deleted. When one deletes or removes an object from the data structure, one must also free the space it uses.

- `t m`

  This indicates that you must print *true* if you already have the web page `m` in the system, and *false* otherwise.

- `c i1`

  If the index term `i1` does not exist, then nothing should be printed. Otherwise, you must print the number of web pages that include the index term `i1`. Note that there must be at least one such web page if the index term is represented by your program.

- `p i1`

  Same is `c i1`, but instead of printing the number of web pages, you should print the id of each web page that contains index term `i1`. **Note that in the printout the web pages do not need to in order.**

- `x i1`

  If the index term `i1` is not in the system, the operation should do nothing. Otherwise, all the web pages that include index term i1 should be deleted and index term `i1` should also be removed. Note that as a result of this operation other index terms may need to be removed. When one deletes or removes an object from the data structure, one must also free the space it uses.

- `y i1 i2`

  You may assume that `i1` is different from `i2`. The value of `i1` may be larger or smaller than `i2`. If either index term `i1` or index term `i2` are not in the system the operation will do nothing. Otherwise, all the web pages containing both index term `i1` and index term `i2` should be removed from the system. This may involve deleting some index terms. When one deletes or removes an object from the data structure, one must also free the space it uses.

- `l`

  Print the number of web pages currently in the system.

- `r`

  Print the id of all the web pages (in increasing order) currently in the system.

- `n`

  Print the number of index terms currently in the system.

- `s`

  Print the id of all the index terms (in increasing order) currently in the system.

- `q`

  Your program must delete all the elements from all the lists and end. You must free all the space that was allocated dynamically.

```
You may assume that each input line is of the form indicated above.
You may also assume that there is a \n after the last character in each line, and
there are no other blanks (or symbols) in the input.  See the sample
input file in the class web page.
```

You must have a different function for each of the commands and a pointer to the first object in the lists (used by the functions) should be passed to functions, i.e., you must pass by value a pointer to the first object in the list. There may be other functions that perform common operations as well as basic ones. For some functions you may need to return a pointer back to the calling function or pass to the function the address to a pointer.

You must have a `.h` file and several `.c` files for your functions and the main code. You must use a `makefile`. The executable produced when we type `make` should be the file called `executeit`.

Turnin electronically to hw04@cs60. You must include all your .h, .c, makefile and the student.id files.

Example: The resulting state after of the list after each command is given below. We also indicate what the command will output (if nothing is printed then it is left blank). Note that the "state of the system" should NOT be printed. It is included below to illustrate the information your program should be maintaining. The initial condition is that the web page list and the list of index terms must be empty.

```
INPUT COMMAND    Resulting State
-----------      ----------------------
                    State:
                     Web Pages
                     Index terms
i 40 5 9         Output: i 40 5 9
                    State:
                     Web Pages
                      40:  5  9
                     Index terms
                      5 9
i 30 3 9         Output: i 30 3 9
                    State:
                     Web Pages
                      30:  3  9
                      40:  5  9
                     Index terms
                      3  5  9
i 52 4 7         Output: i 52 4 7
```

```
                    State:
                     Web Pages
                      30:  3  9
                      40:  5  9
                      52:  4  7
                     Index terms
                      3  4  5  7  9
i 55 3 9          Output: i 55 3 9
                    State:
                     Web Pages
                      30:  3  9
                      40:  5  9
                      52:  4  7
                      55:  3  9
                     Index terms
                      3  4  5  7  9
i 30 1 12         Output: i 30 1 12
                    State:
                     Web Pages
                      30:  3  9
                      40:  5  9
                      52:  4  7
                      55:  3  9
                     Index terms
                      3  4  5  7  9
d 52              Output: d 52
                    State:
                     Web Pages
                      30:  3  9
                      40:  5  9
                      55:  3  9
                     Index terms
                      3  5  9
i 52 3 4          Output: i 52 3 4
                    State:
                     Web Pages
                      30:  3  9
                      40:  5  9
                      52:  3  4
                      55:  3  9
                     Index terms
                      3  4  5  9
t 30              Output: t 30
                          true
                    State:
                     Web Pages
                      30:  3  9
                      40:  5  9
                      52:  3  4
                      55:  3  9
                     Index terms
                      3  4  5  9
t 45              Output: t 45
                          false
```

```
                    State:
                     Web Pages
                      30:  3  9
                      40:  5  9
                      52:  3  4
                      55:  3  9
                     Index terms
                      3  4  5  9
c 9                 Output: c 9
                            3
                    State:
                     Web Pages
                      30:  3  9
                      40:  5  9
                      52:  3  4
                      55:  3  9
                     Index terms
                      3  4  5  9
c 30                Output: c 30
                    State:
                     Web Pages
                      30:  3  9
                      40:  5  9
                      52:  3  4
                      55:  3  9
                     Index terms
                      3  4  5  9
p 3                 Output: p 3
                            30
                            52
                            55
                    State:
                     Web Pages
                      30:  3  9
                      40:  5  9
                      52:  3  4
                      55:  3  9
                     Index terms
                      3  4  5  9
P 8                 Output: p 8
                    State:
                     Web Pages
                      30:  3  9
                      40:  5  9
                      52:  3  4
                      55:  3  9
                     Index terms
                      3  4  5  9
x 3                 Output: x 3
                    State:
                     Web Pages
                      40:  5  9
                     Index terms
                      5  9
```

```
i 22 2 3            Output: i 22 2 3
                    State:
                     Web Pages
                      22:  2  3
                      40:  5  9
                     Index terms
                      2  3  5  9
i 25 5 7            Output: i 25 5 7
                    State:
                     Web Pages
                      22:  2  3
                      25:  5  7
                      40:  5  9
                     Index terms
                      2  3  5  7  9
i 35 4 6            Output: i 35 4 6
                    State:
                     Web Pages
                      22:  2  3
                      25:  5  7
                      35:  4  6
                      40:  5  9
                     Index terms
                      2  3  4  5  6  7  9
y 2 4               Output: y 2 4
                    State:
                     Web Pages
                      22:  2  3
                      25:  5  7
                      35:  4  6
                      40:  5  9
                     Index terms
                      2  3  4  5  6  7  9
y 4 6               Output: y 4 6
                    State:
                     Web Pages
                      22:  2  3
                      25:  5  7
                      40:  5  9
                     Index terms
                      2  3  5  7  9
y 5 7               Output: y 5 7
                    State:
                     Web Pages
                      22:  2  3
                      40:  5  9
                     Index terms
                      2  3  5  9
l                   Output: l
                            2
                    State:
                     Web Pages
                      22:  2  3
                      40:  5  9
```

```
                    Index terms
                     2  3  5  9
r                   Output: r
                            22
                            40
                    State:
                     Web Pages
                      22:  2  3
                      40:  5  9
                     Index terms
                      2  3  5  9
n                   Output: n
                            4
                    State:
                     Web Pages
                      22:  2  3
                      40:  5  9
                     Index terms
                      2  3  5  9
s                   Output: s
                            2
                            3
                            5
                            9
                    State:
                     Web Pages
                      22:  2  3
                      40:  5  9
                     Index terms
                      2  3  5  9
q                   Output: q
                    State:
                     Web Pages
                     Index terms
```