CP2 Exercise 4 **13/11/2006**

1. Change last week's exercise 2 so that instead of a temporary file it uses a dynamically allocated block of memory to store the original data from the log file while the newest entry is added at the beginning of the file (*before writing back the old logs*).

2. Expand exercise 2 from the 30th of October to behave like follows:

a) change the program to accept the month to be written from the command-line instead of explicitly asking for it.

b) compare each date written into the HTML file with the date returned by the system time. If the current date matches that of a date written into the HTML file then highlight it using either a different colour or font style (*bold, italic etc.*).

c) change the program to also open a text file "**dates.txt**" in which special dates have been saved in the following format:

```
0101:New Year's Day
0214:Valentine's Day
1031:Hallowe'en
1105:Guy Fawkes Day
1224:Christmas Eve
1225:Christmas Day
1226:Boxing Day
1231:New Year's Eve
```

Each date is saved in a separate line in the format ***MMDD:string*** where *MM* is the month, *DD* is the day,':' is a delimiter and *string* is the description of the day.

Your program should first use the tokenisation function written for last week (*or alternatively* ***strtok***) to break up each of the lines read from the "**dates.txt**" file into its date and text parts. Using an array of 366 strings the descriptions of dates should then be copied into the apropriate entry (*index of the day*) of the array. All other dates in the array should be set to empty strings. If you use your own tokenisation function with dynamic memory allocation then do not forget to free tokens after using them.

While writing the calendar file your program should then print the special dates found in the array into the HTML table entry for matching days.

For an example HTML file please see:
http://programming.swordfighter.co.uk/cp2ex0405/ex0104table.html