

CS13002 Programming and Data Structures, Spring 2006

Lab test 1

Total points: 20

February 07, 2006

Time: 13:30–16:00

For students with even PC numbers

Time is specified in the standard format **dd-mm-yyyy hh:mm:ss**. Here **dd** is a 2-digit representation of the date. A one-digit date (1 through 9) needs to be preceded by a 0. Similarly, for the month **mm** (an integer in the range 01–12), the hour **hh** (00–23), the minute **mm** (00–59) and the second **ss** (00–59). Some valid examples are: **07-02-2006 13:39:02** and **29-02-2000 00:59:59**. Some invalid examples are: **31-04-2007 00:59:59** (invalid date), **07-02-2006 25-59-73** (invalid time).

In computers, time is often represented more compactly. For example, the Unix™ operating system represents system time by a 32-bit *signed* integer. It stores the number of seconds from the *epoch* which is **01-01-1970 00:00:00** (12 midnight of Jan 01, 1970). Positive values imply times after the epoch and negative values stand for times before the epoch. For this test, you are asked to investigate positive times only (though you represent time by a 32-bit *signed* integer).

Let us now see how we can convert a time in the prescribed format (**dd-mm-yyyy hh:mm:ss**) to the number of seconds after the epoch. First compute the number of days elapsed after the epoch. This number equals $(yyyy - 1970) \times 365 + l + d$, where l is the number of leap years after the epoch and before the supplied time (the year 2000 was a leap year), and d is the contribution due to the date and month fields. Each day has 86400 seconds. Convert the number of days to number of seconds. Finally, add the contribution of the day's time **hh:mm:ss**.

Write a program that does the following. It reads an integer value n in the range $0 \leq n \leq 2^{31} - 1$ and converts that integer n to the equivalent time **dd-mm-yyyy hh:mm:ss**. You should reverse the method described in the previous paragraph. First determine the year, then the month and the date. Subsequently, decide the hour, minute and second of the day. Finally, print the time in the prescribed format.

Sample output

```
Enter system time in seconds : 2000000000
Time (dd-mm-yyyy hh:mm:ss) is 18-05-2033 03:33:20

Enter system time in seconds : 123456789
Time (dd-mm-yyyy hh:mm:ss) is 29-11-1973 21:33:09
```

Test input

Submit the output of your program for the following ten values of system time:

```
0 /* The epoch */
97531
246813579
730989296
946684799
946684800
1000000000
1330559999
1330560000
2147483647 /* This is the largest representable time*/
```