



Chapitre 1 – Structure pour la gestion de temps
Struktur zur Behandlung von Zeit

Exercice Algo 1.7

Dans cet exercice nous voulons réaliser une petite application qui permet d'utiliser une structure de temps qui est définie dans le fichier time.h:

In dieser Übung wollen wir eine kleine Applikation realisieren, welche es erlaubt eine Zeitstruktur zu verwenden welche in der Datei time.h definiert ist:

```
struct tm
{
    int tm_sec; /* Seconds: 0-59 (K&R says 0-61?) */
    int tm_min; /* Minutes: 0-59 */
    int tm_hour; /* Hours since midnight: 0-23 */
    int tm_mday; /* Day of the month: 1-31 */
    int tm_mon; /* Months *since* january: 0-11 */
    int tm_year; /* Years since 1900 */
    int tm_wday; /* Days since Sunday (0-6) */
    int tm_yday; /* Days since Jan. 1: 0-365 */
    int tm_isdst; /* +1 Daylight Savings Time, 0 No DST, * -1 don't know */
};
```

```
workspace_Algo_VSC> .\Algo-Ex1.7_sol\main.exe
Programm "Algo-Ex_1_7-Structure"
a) Current date and time is      : 15.10.2020 09:21:25
b) Overflow will occur 1s after: 19.01.2038 04:14:07
c) Current date and time is      : 15.10.2020 09:21:25
   Overflow will occur 1s after: 19.01.2038 04:14:07
d) Current week day is          : "Thursday"
```

Notes/Bemerkungen:

- Infos: https://www.tutorialspoint.com/c_standard_library/c_function_time.htm
 - Le temps actuel peut être obtenue en utilisant la fonction suivante, cependant time_t est normalement définie comme variable 32bit signé:
Die aktuelle Zeit kann mit folgender Funktion ermittelt werden, wobei time_t normalerweise als vorzeichenbehaftete 32bit Variable definiert ist:

```
time_t time(time_t *t)
```

- La conversion du temps actuel dans le format de la structure tm peut être réalisé en utilisant la fonction suivante:
Die Umwandlung der nun ermittelten Zeit in das Format der Struktur tm kann nun mittels Aufruf der folgenden Funktion vorgenommen werden:

```
struct tm *localtime(const time_t *timer)
```

- a) Implémentez dans **main()** l'affichage de l'information du **temps actuel** en utilisant **printf** avec le format indiqué sur la première page.
Implementieren Sie in **main()** die Anzeige zur Information der **aktuellen Zeit** gemäß dem auf der ersten Seite angegebenen Formats.

```
#include <stdio.h>
#include <time.h>

int main()
{
    printf("Programm \"Algo-Ex_1_7-Structure\"\n\n");

    time_t const cTimeInSec = time(NULL); // get actual time in seconds

    struct tm* const pCurrTime
        = localtime(&cTimeInSec); // convert seconds in time structure

    printf(" a) Current date and time is      : %02d.%02d.%4d %02d:%02d:%02d",
        pCurrTime->tm_mday, pCurrTime->tm_mon+1, 1900+pCurrTime->tm_year,
        pCurrTime->tm_hour, pCurrTime->tm_min, pCurrTime->tm_sec);

    return 0;
}
```

- b) Ajoutez dans **main()** la fonctionnalité qui permet d'afficher quand exactement le problème de **dépassement** de la variable 32bit signé du type **time_t** va apparaître. Pour ceci, il est possible d'utiliser des constantes définies dans le fichier **limits.h**. Fügen Sie in **main()** die Funktionalität ein, welche es ermöglicht genau anzugeben, wann die vorzeichenbehaftete 32bit Variable vom Typ **time_t** **überlaufen** wird. Hierzu können Sie auch die Konstanten verwenden, welche in der Datei **limits.h** definiert sind.

```
#define DATE_TIME_FORMAT "%02d.%02d.%4d %02d:%02d:%02d"

time_t const cTimeMax = INT_MAX;
struct tm* pMaxTime
    = localtime(&cTimeMax); // convert seconds in time structure

printf("\n\n b) Overflow will occur 1s after: " DATE_TIME_FORMAT,
    pMaxTime->tm_mday, pMaxTime->tm_mon+1, 1900+pMaxTime->tm_year,
    pMaxTime->tm_hour, pMaxTime->tm_min, pMaxTime->tm_sec);
```

- c) Implémentez une **fonction** qui prend comme paramètre le texte et l'information du temps à afficher et adaptez main() pour appeler cette fonction au lieu de printf dans les exercices précédentes .
Implementieren Sie eine **Funktion** welche als Parameter den anzuzeigenden Text und die Zeit übernimmt und passen Sie main() an, so dass diese Funktion in den vorherigen Aufgaben statt printf aufgerufen wird.

```
void showTime(char* pText, struct tm const * const pTime)
{
    assert(pText!=0);
    assert(pTime!=0);
    printf("%s: " DATE_TIME_FORMAT, pText,
           pTime->tm_mday, pTime->tm_mon+1, 1900+pTime->tm_year,
           pTime->tm_hour, pTime->tm_min, pTime->tm_sec);
}
```

```
int main()
{
    // ...
    struct tm* const pCurrTime2 = localtime(&cTimeInSec); // convert again...
    showTime("\n\n c) Current date and time is      ", pCurrTime2);

    struct tm* pMaxTime2 = localtime(&cTimeMax); // convert again
    showTime("\n      Overflow will occur 1s after", pMaxTime2);
}
```

- d) Implémentez une **fonction** qui prend comme paramètre l'information du jour de la semaine et détermine son nom du jour correspondant. Affichez dans main le nom du jour en utilisant printf qui appelle cette fonction.
Implementieren Sie eine **Funktion** welche als Parameter den Wochentag übernimmt und diesen in den entsprechenden Namen des Wochentages umwandelt. Zeigen Sie in main nun den Wochentag an, indem Sie diese Funktion verwenden und in printf aufrufen.

```
char const * getWeekDayAsText(int dayOfWeek)
{
    #define NbrOfWeekDays (7)
    char const * const pDayOfWeekName[NbrOfWeekDays] =
        {"Sunday", "Monday", "Tuesday", "Wednesday", "Thursday",
         "Friday", "Saturday"};

    assert(dayOfWeek < NbrOfWeekDays);
    return pDayOfWeekName[dayOfWeek];
}
```

```
int main()
{
    // ...
    printf("\n\n d) Current week day is          : \"%s\"\n",
           getWeekDayAsText(pCurrTime->tm_wday));

    return 0;
}
```