

Microcontroller / C-Programmierung

Selbststudium Semesterwoche 4

1. Aufgabe 5-3 im Buch Seite 103: Schreiben Sie die Funktion `strcat` mit Zeiger, welche eine Zeichenkette `t` an das Ende der Zeichenkette `s` kopiert. Diese Aufgabe lehnt an das Beispiel `strcat(s,t)` aus Kapitel 2 im Buch an.

```
#include <stdio.h>
void strcat(char*, char*);

void main(void)
{
    // test string: "te" and "st"
    char s[] = "te";
    char t[] = "st";

    // addresses as parameters
    strcat(s, t);

    // finally print the concatenated string to console
    printf("%s", s);

    getchar();
}
/* This function concatenates string t to end of string s; s must be big enough
 * @param char *s string value that will be extended with characters of *t
 * @param char *t string value that will be copied to the end of *s
 */
void strcat(char *s, char *t)
{
    while (*s != '\0') // find the end of char array *s
    {
        s++;           // increment address of s
    }

    while ((*s++) = *(t++)) != '\0' // copy value by value from t to s
    ;
}

```

2. Aufgabe 5-8 (Buch Seite 108): Die Funktionen `day_of_year` und `month_of_year` sollen mit Fehlerprüfung versehen werden.

```
#include <stdio.h>

// macro that determines the correct ending of numbers (e.g. 1st, 2nd, 3rd,...)
#define numEnd(val) (val==1 ?"st":(val==2 ?"nd":(val==3 ?"rd":"st")))

int day_of_year(int, int, int);
void month_day(int, int, int*, int*);

static char daytab[2][13] = {
    {0, 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31},
    {0, 31, 29, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31}
};

void main(void)
{
    int y, m, d;

    y = 1900, m = 0, d = 56;
    printf("Day %i of the Year %i\n", d, y);
    month_day(y, d, &m, &d);
    if(m>0 || d>0)

```

```
{
    printf("was the %i%s Day in the %i%s Month", d, numEnd(d), m, numEnd(m));
}
else
{
    printf("Input Error: Verify your input data!");
}

printf("\n\n");

y = 1900, m = 2, d = 25;
printf("Day %i of Month %i in the Year %i\n", d, m, y);
d = day_of_year(y, m, d);
if(d>0)
{
    printf("was the %i%s Day of the Year", d, numEnd(d));
}
else
{
    printf("Input Error: Verify your input data!");
}

getchar();
}

/* day_of_year: set day of year from month & day */
int day_of_year(int year, int month, int day)
{
    int i, leap;
    leap = year%4 == 0 && year%100 != 0 || year%400 == 0;

    // error checking
    if((year < 0) || (month < 1 || month > 12) || (day < 1 || day > daytab[leap][month]))
    {
        return -1;
    }

    for (i = 1; i < month; i++)
    {
        day += daytab[leap][i];
    }
    return day;
}

/* month_day: set month, day from day of year */
void month_day(int year, int yearday, int *pmonth, int *pday)
{
    int i, leap;

    // error checking
    if(year < 0 || yearday < 1 || yearday > 365)
    {
        *pmonth = -1;
        *pday = -1;
        return;
    }

    leap = year%4 == 0 && year%100 != 0 || year%400 == 0;

    // i is the index of the month-of-the-year
    for (i = 1; yearday > daytab[leap][i]; i++)
    {
        yearday -= daytab[leap][i];
    }
    *pmonth = i;
    *pday = yearday;
}
```

3. Aufgabe 5-9 (Buch Seite 110): Die Funktionen `day_of_year` und `month_of_year` soll mit Zeiger statt Indexes implementiert werden.

```
/* day_of_year: set day of year from month & day */
int day_of_year(int year, int month, int day)
{
    int i = 0, leap = 0;
    char *p;
    leap = year%4 == 0 && year%100 != 0 || year%400 == 0;

    // error checking
    if((year < 0) || (month < 1 || month > 12) || (day < 1 || day > daytab[leap][month]))
    {
        return -1;
    }

    p = daytab[leap]; // point p to address of daytab[leap][0]
    while(month-- != 0)
    {
        day += *p++; // increment pointer p to the next address
    }
    return day;
}

/* month_day: set month, day from day of year */
void month_day(int year, int yearday, int *pmonth, int *pday)
{
    int i = 0, leap = 0;
    char *p;

    // error checking
    if(year < 0 || yearday < 1 || yearday > 365)
    {
        *pmonth = -1;
        *pday = -1;
        return;
    }

    leap = year%4 == 0 && year%100 != 0 || year%400 == 0;

    p = daytab[leap]; // point p to address of daytab[leap][0]
    while(yearday > *p)
    {
        yearday -= *p++;
        i++;
    }

    *pmonth = i;
    *pday = yearday;
}
```

4. Implementieren Sie die Beispiel-Funktionen `alloc` und `afree` von Seite 98 im Buch. Testen Sie diese auf eine korrekte Funktion.