Calendars

Implement a series of functions for calendat date calculations.

Introduction

In this assignment, we consider all dates starting with Oct 16th 1582 and ending with Dec 31st 2199 as valid.

Use the provided main function to test your own functions.

Assignment

Implement the following functions:

Basics

- `int isLeap(int year)` returns 1 if year is a leap year, -1 if invalid and 0 otherwise
- `int daysInMonth(int year, int month)` returns the number of days in month of year, or -1 if date is invalid.
- `int checkDate1(int year)`
  returns whether the given year is valid, i.e. returns 1 if date is valid, and 0 otherwise.
- `int checkDate(int year, int month)`
  returns whether the given month in the give year is valid, i.e. returns 1 if date is valid, and 0 otherwise.
- `int checkDate(int year, int month, int day)` returns whether the given date is valid, i.e. returns 1 if date is valid, and 0 otherwise.

Calculations

- `int ymd2w(int year, int month, int day)` calculates the weekday of a given date. The date is passed to the function as three integers. Returns the weekday according to Table 1 in Weekdays exercise or -1, if date is invalid.
- `int dayNumber(int year, int month, int day)` returns the day count, or -1 if the given date is invalid. The day count is the number of days passed since the start of the year plus one. E.g.: Jan, 1 st is day 1 in any year; Dec. 31 st is day 365 in a non-leap year and day 366 in a leap year.
- `int weekNumber(int year, int month, int day)` returns the week number of a given date, or -1 if date is invalid. Week 1 in any year is the week of Jan, 1 st .

Output

- `void printDayName(int day)` prints the name of day. The name of a day is given in Table 1 in Exercise Weekdays. prints `invalid day(<day>)` if day is not in the range 0-6. The output produced contains no whitespace or other additional characters before or after the name. The names are the full English names, e.g. “Monday”, “Friday”. . .

- `void printMonth(int month)` prints the name of month. The months accepted are in the range 1-12, 1 being January. prints `invalid day(<month>)` if month is not in the range 1-12. The output produced contains no whitespace or other additional characters before or after the name. The names are the full English names, e.g. “January”, “February”...
- `void printNmberEnding(int n)` prints the English number ending for `n` the number ending is:
  - `st` for numbers ending with 1, except 11
  - `nd` for numbers ending with 2, except 12
  - `rd` for number ending with 3 except 13
  - `th` otherwise
- `void printDate(int year, int month, int day)` prints a given date in the following format: `<Weekday>, <Month> <day> <number_ending> <year>` prints `invalid date (<day><month>.<year>).` for invalid dates.

example:

```
Saturday, January 1st 2000
```

- `void printStatistics(int year, int month, int day)` prints various information on the given date:
  - date as described in printsDate(int,int,int)
  - the year and whether it is a leap year or not
  - the number of days in the month of the given date
  - the number of the day
  - the week of the date

  prints `invalid date.` for invalid dates.

example:

```
statistics for
Saturday, January 1st 2000
2000 is a leap year
January 2000 has 31 days
it is day 1 of 2000
it is in week 1
```

Applications

- `void printCalendar(int year, int month, int day, int highlight)` prints a calendar of the month of the given date. Each line of the calendar displays one week. The weeks in the calendar start on Monday. if highlight is set, the given date is highlighted with brackets <>in the calendar. see example calendar for Jan, 14 th 2000:

```
January 2000
Mon Tue Wed Thu Fri Sat Sun
                      1   2
 3   4   5   6   7   8   9
10  11  12  13 <14> 15  16
17  18  19  20  21  22  23
24  25  26  27  28  29  30
```

```
31
```

Examples:

(Text in red = user input) The examples is created using the provided main function and shows one complete run for JAnuary 1st 1970.

```
Current date: 01.01.1970
enter option: 1

isLeap(year): 0

current date: 01.01.1970
enter option: 2

daysInMonth(year, month): 31

current date: 01.01.1970
enter option: 3

checkDate1(year): 1
checkDate2(year, month): 1
checkDate3(year, month, day): 1

current date: 01.01.1970
enter option: 7

ymd2w(year, month, day): 4

current date: 01.01.1970
enter option: 8

dayNumber(year, month, day): 1

current date: 01.01.1970
enter option: 9

weekNumber(year, month, day): 1

current date: 01.01.1970
enter option: 20

printStatistics: statistics for
Thursday, January 1st 1970
1970 is not a leap year
month 1 in year 1970 has 31 days
it is day 1 of year 1970
it is in week 1 of year 1970

current date: 01.01.1970
enter option: 21

printCalendar(year, month, day, 1):
January 1970
Mon Tue Wed Thu Fri Sat Sun
```

```
          < 1>  2   3   4
 5   6   7   8   9  10  11
12  13  14  15  16  17  18
19  20  21  22  23  24  25
26  27  28  29  30  31

current date: 01.01.1970
enter option: 22

printCalendar(year, month, day, 0):
January 1970
Mon Tue Wed Thu Fri Sat Sun
            1   2   3   4
 5   6   7   8   9  10  11
12  13  14  15  16  17  18
19  20  21  22  23  24  25
26  27  28  29  30  31

current date: 01.01.1970
enter option: 666
```

Hints:

- Eine Auflistung welche Testfälle was testen finden Sie in der Tabelle::

test case(s)	Test
101-106	isLeap
201-215	daysInMonth
301-310	checkDate

- use the provided file with the following `main` function as a starting point:

```
#include <stdio.h>
#include <stdlib.h>
int isLeap(int year);
int daysInMonth(int year, int month);
int checkDate1(int year);
int checkDate2(int year, int month);
int checkDate3(int year, int month, int day);



int ymd2w(int year, int month, int day);
int dayNumber(int year, int month, int day);
int weekNumber(int year, int month, int day);


void printDayName(int w);
void printMonthName(int month);
void printNmberEnding(int n);

void printDate(int year, int month, int day);
void printStatistics(int year, int month, int day);
```

```
void printCalendar(int year, int month, int day, int highlight);

int main() {

#define SET_DATE  0
#define IS_LEAP  1
#define DAYS_IN_MONTH  2
#define CHECK_DATE  3

#define YMD_2_W  7
#define DAY_NUMBER  8
#define WEEK_NUMBER  9
#define COMPARE  10

#define DAY_NAME  13
#define PRINT_DAY  14
#define DAY_SHORT_NAME  15
#define PRINT_DAY_SHORT  16
#define PRINT_MONTH  17
#define PRINT_MONTH_SHORT  18
#define PRINT_DATE  19
#define PRINT_STATISTICS  20

#define PRINT_CALENDAR_HIGHLIGHT  21
#define PRINT_CALENDAR  22

#define EXIT 666

    int day=1, month=1, year=1970;
    while (1) {

        int op;
        printf("current date: %02d.%02d.%04d\n", day,month, year);

        printf("enter option: ");
        scanf("%d", &op);



        switch (op) {
        case SET_DATE:
            printf("day: ");
            scanf("%d", &day);
            printf("month: ");
            scanf("%d", &month);
            printf("year: ");
            scanf("%d", &year);

            break;
        /*------------------------ BASICS ---------------------*/
        case IS_LEAP:
            printf("\nisLeap(year): %d\n", isLeap(year));
            break;

        case DAYS_IN_MONTH:
            printf("\ndaysInMonth(year, month): %d\n", daysInMonth(year, month));
```

```
            break;

        case CHECK_DATE:
            printf("\ncheckDate1(year): %d\n", checkDate1(year));
            printf("checkDate2(year, month): %d\n", checkDate2(year, month));
            printf("checkDate3(year, month, day): %d\n", checkDate3(year, month, day));
            break;


        /*------------------------- CALCULATIONS ----------------------*/
        case YMD_2_W: {
            int w;
            printf("\nymd2w(year, month, day): %d\n", w=ymd2w(year, month, day));
            break;
        }
        case DAY_NUMBER:
            printf("\ndayNumber(year, month, day): %d\n", dayNumber(year, month, day));
            break;


        case WEEK_NUMBER:
            printf("\nweekNumber(year, month, day): %d\n", weekNumber(year, month, day));
            break;


            /*------------------------- OUTPUT ----------------------*/


        case PRINT_STATISTICS:
            printf("\nprintStatistics: ");
            printStatistics(year, month, day);

            break;

        /*------------------------- APPLICATIONS ----------------------*/
        case PRINT_CALENDAR_HIGHLIGHT:
            printf("\nprintCalendar(year, month, day, 1): \n");
            printCalendar(year, month, day, 1);

            break;
        case PRINT_CALENDAR:
            printf("\n01printCalendar(year, month, day, 0): \n");
            printCalendar(year, month, day, 0);

            break;
        case EXIT:
            printf("closing.\n");
            return 0;

        default:
            ;
        }
        printf("\n");
    }

    return 0;
```

```
}
```