TASK 1 Calling time(NULL) gives you a large integer. Use this value to calculate today's year, date, hours, and minutes.

//

// LAB POINTERS TASK 1

// 4/12/20

// Sidi Liang

//

#include <stdio.h>

#include <time.h>

#define TOTAL\_SEC\_IN\_DAY 86400

void calculateYearAndDay(time\_t now, int \*year, int \*day); //Function to calculate Year and Day

void calculateHourAndMinute(time\_t now, long \*hour, long \*minute);//Function to calculate hour and minute

void printTime(int year, int day, long hour, long minute);//Function to print time

int main() {

time\_t now;

now = time(NULL); //total seconds

//printf("%li\n", now);

int year = 0, day = 0;

long hour = 0, minute = 0;

calculateYearAndDay(now, &year, &day);

calculateHourAndMinute(now, &hour, &minute);

printTime(year, day, hour, minute);

return 0;

}

void calculateYearAndDay(time\_t now, int \*year, int \*day){

long int totalDay = now / TOTAL\_SEC\_IN\_DAY;

\*year = (int)totalDay / 365 + 1970;

\*day = (int)totalDay - (\*year - 1970) \* 365 + 1;

}

void calculateHourAndMinute(time\_t now, long \*hour, long \*minute){

long totalDay = now / TOTAL\_SEC\_IN\_DAY;

long totalMinute = now / 60, totalHour = now / 3600;

\*hour = totalHour - (24 \* totalDay);

\*minute = totalMinute - totalHour \* 60;

}

void printTime(int year, int day, long hour, long minute){

printf("%d, %d\n%ld, %ld\n", year, day, hour, minute);

}

Code for TASK 1

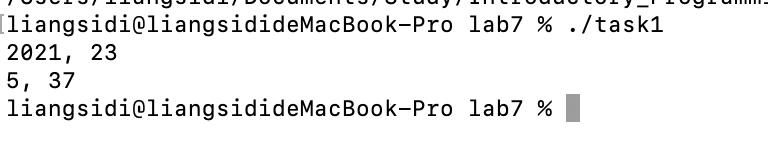


Figure 1: Output of Code in Task 1.

1. TASK 1 optional optional: handle leap-years correctly, and print the month as well

//

// LAB POINTERS TASK 1

// 4/12/20

// Sidi Liang

//

#include <stdio.h>

#include <time.h>

#define TOTAL\_SEC\_IN\_DAY 86400

void calculateYearAndDay(time\_t now, int \*year, int \*day, int \*month); //Function to calculate Year and Day

void calculateHourAndMinute(time\_t now, long \*hour, long \*minute);//Function to calculate hour and minute

void printTime(int year, int day, int month, long hour, long minute);//Function to print time

int isLeapYear(int year){

if(year % 100 == 0 && year % 400 != 0){

return 0;

}

if(year % 4 == 0){

return 1;

}

return 0;

}

int main() {

time\_t now;

now = time(NULL); //total seconds

//printf("%li\n", now);

int year = 0, day = 0, month = 0;

long hour = 0, minute = 0;

calculateYearAndDay(now, &year, &day, &month);

calculateHourAndMinute(now, &hour, &minute);

printTime(year, day, month, hour, minute);

return 0;

}

void calculateYearAndDay(time\_t now, int \*year, int \*day, int \*month){

long int totalDay = now / TOTAL\_SEC\_IN\_DAY - 12; //hard coded leap year

\*year = (int)totalDay / 365 + 1970;

\*day = (int)totalDay - (\*year - 1970) \* 365 + 1;

if(\*day <= 31){

\*month = 1;

}else if(\*day <= 60){

\*month = 2;

\*day -= 31;

}else if(\*day <= 91 && isLeapYear(\*year)){//hard coded 2020

\*month = 3;

\*day -= 60;

}else if(\*day <= 90 && !isLeapYear(\*year)){//hard coded 2020

\*month = 3;

\*day -= 60;

}else if(\*day <= 121){

\*month = 4;

if(isLeapYear(\*year)) \*day -= 91;

else \*day -= 90;

}else if(\*day <= 152){

\*month = 5;

\*day -= 121;

}else if(\*day <= 182){

\*month = 6;

\*day -= 152;

}else if(\*day <= 213){

\*month = 7;

\*day -= 182;

}else if(\*day <= 244){

\*month = 8;

\*day -= 213;

}else if(\*day <= 274){

\*month = 9;

\*day -= 244;

}else if(\*day <= 305){

\*month = 10;

\*day -= 274;

}else if(\*day <= 335){

\*month = 11;

\*day -= 305;

}

else if(\*day <= 366){

\*month = 12;

\*day -= 335;

}

}

void calculateHourAndMinute(time\_t now, long \*hour, long \*minute){

long totalDay = now / TOTAL\_SEC\_IN\_DAY;

long totalMinute = now / 60, totalHour = now / 3600;

\*hour = totalHour - (24 \* totalDay);

\*minute = totalMinute - totalHour \* 60;

}

void printTime(int year, int day, int month, long hour, long minute){

printf("%d, %d, %d\n%ld, %ld\n", year, month, day, hour, minute);

}

Code for TASK 1.1

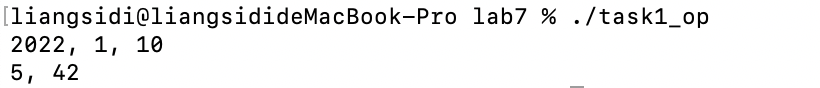


Figure 2: Output of Code in Task 1.1.