

|  |
| --- |
| **R E P O R T** |

|  |  |
| --- | --- |
| 제 목 | 그레고리 달력의 요일 계산하기 |
| 과 목 | 컴퓨터 개론 및 실습 |
| 날짜(년/월/일) | 2017/04/13 |
| 소속 학과 | 컴퓨터 전자 시스템 공학부 |
| 학번 | 201702234 |
| 이름 | 유동혁 |

|  |  |
| --- | --- |
| 본 보고서의 내용 중 다른 문서(자료)를 인용한 것이 있습니까?  예 ( ) 아니오 ( V ) | |
| 위에서 ‘예’로 답한 경우, 인용한 다른 문서는 무엇인지 아래에 명시해 주세요.(여러 개의 경우 주요 자료 2개 까지) | |
| - 저자 1 : |  |
| * 제목 1 : |  |
| * 저자 2 : |  |
| * 제목 2 : |  |

1. **날짜를 입력 받으면 그 날짜의 요일을 알려주는 프로그램**
   1. **Solution**

- char 과 %c는 문자 하나를 저장하고 char\* 과 %s는 문자열을 저장한다. 따라서 요일을 저장하기위해 char\* 과 %s를 사용한다.

- 조건에 따라 다른 결과를 도출하기위해서 if문을 사용한다.

- 조건들이 중첩되어야 하므로 if문을 중첩해서 사용한다.

- 상수끼리 단순 비교를 여러 번 할 때는 if문 보다는 switch-case문을 사용하는게 편하다. 그러므로 요일, 월을 단순 비교할 때 switch-case문을 사용한다.

* 1. **Source code**

/\*

\* gregory.c

\*

\* Created on: 2017. 4. 6.

\* Author: 유동혁

\*/

#include<stdio.h>

//그레고리 달력의 요일 계산하기

int main()

{

setvbuf(stdout, NULL, \_IONBF, 0);

int year, month, day;

char\* yoill;

int juli, y, m, d;//julian값을 계산하기위해 계산용을 분리

printf("Enter gregorian year(>= 1583) : ");

scanf("%d", &year);

printf("Enter gregorian month(1, 2, ..., 12) : ");

scanf("%d", &month);

printf("Enter gregorian day(1, 2, ..., 28|29|30|31) : ");

scanf("%d", &day);

//Line 28 ~ 40 : julian값 계산하기

y = year;

m = month;

d = day;

y += 8000;

if(month < 3)

{

y--;

m += 12;

}

juli = (y \* 365) + (y / 4) - (y / 100) + (y / 400) - 1200820 + (m \* 153 + 3) / 5 - 92 + (d - 1);

//Line 43 ~ 72 : julian값을 7로 나눈 나머지에 따라 요일 정하기

switch(juli % 7)

{

case 0:

yoill = "Monday";

break;

case 1:

yoill = "Tuesday";

break;

case 2:

yoill = "Wednesday";

break;

case 3:

yoill = "Thursday";

break;

case 4:

yoill = "Friday";

break;

case 5:

yoill = "Saturday";

break;

case 6:

yoill = "Sunday";

break;

}

//Line 76 ~ 147 : 범위 밖 날짜 에러처리하기

if (year < 1583)//만약 1583미만의 연도가 입력되면

{

printf("Year Error! Enter year >= 1583");

}

else//입력된 연도가 1583 이상이면

{

if (month < 1 || month > 12)//입력된 달어 1 미만 12 초과면

{

printf("Month Error! Enter 1<=month<=12");

}

else//입력된 달이 1이상 12 이하면

{

if (month == 2)//2월이면

{

if (((year % 4 == 0) && (year % 100 != 0)) || (year % 400 == 0))//윤년이면

{

if(day < 1 || day > 29)//입력된 날짜가 1미만 29초과면

{

printf("Day Error! Enter 1<=day<=29");

}

else//어떤 에러에도 해당되지 않으면

{

printf("%d-%d-%d is %s", year, month, day, yoill);

}

}

else//윤년이 아니면

{

if(day < 1 || day > 28)//입력된 날짜가 1미만 28초과면

{

printf("Day Error! Enter 1<=day<=28");

}

else//어떤 에러에도 해당되지 않으면

{

printf("%d-%d-%d is %s", year, month, day, yoill);

}

}

}

else//2월이 아니면

{

switch(month)//월에 따라

{

case 4:

case 6:

case 9:

case 11://4,6,9,11월일때

if(day < 1 || day > 30)//입력된 날짜가 1미만30 초과면

{

printf("Day Error! Enter 1<=day<=30");

}

else//어떤 에러에도 해당되지 않으면

{

printf("%d-%d-%d is %s", year, month, day, yoill);

}

break;

default://1,3,5,7,8,10,12월일때

if(day <1 || day > 31)//입력된 날짜가 1미만 31초과면

{

printf("Day Error! Enter 1<=day<=31");

}

else//어떤 에러에도 해당되지 않으면

{

printf("%d-%d-%d is %s", year, month, day, yoill);

}

break;

}

}

}

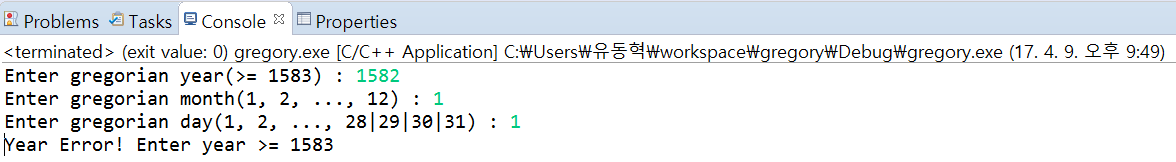
}

return 0;

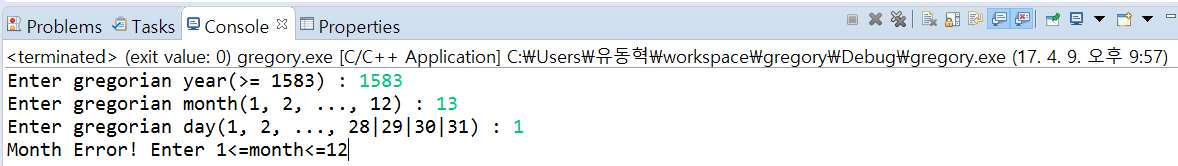
}

* 1. **Result (Snapshot)**

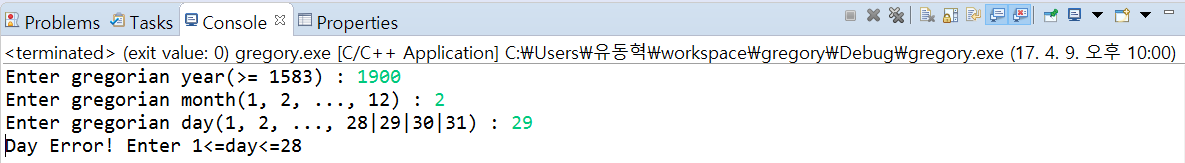
↓1582/01/01



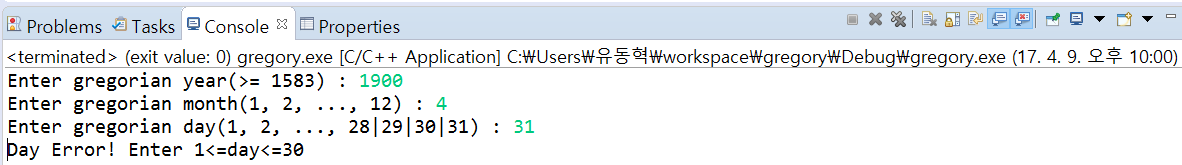
↓1583/13/01



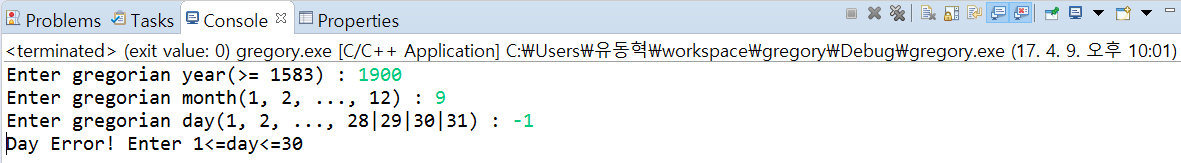
↓1900/02/29



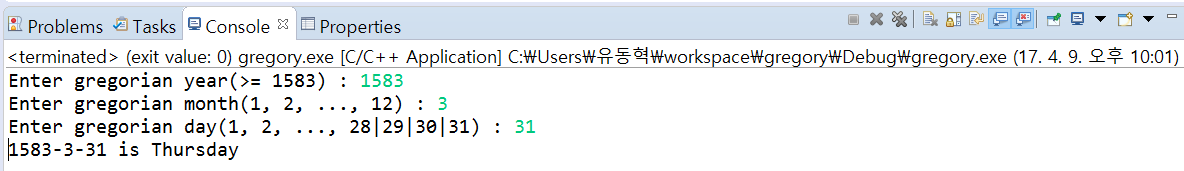
↓1900/04/31



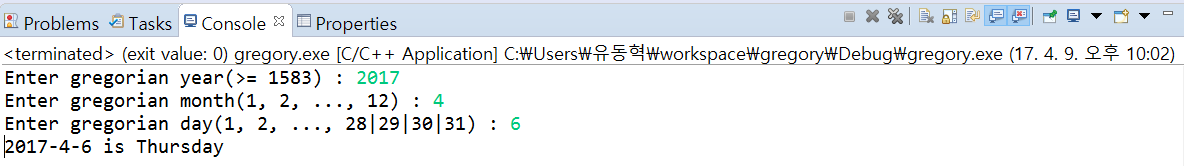
↓1900/09/-1



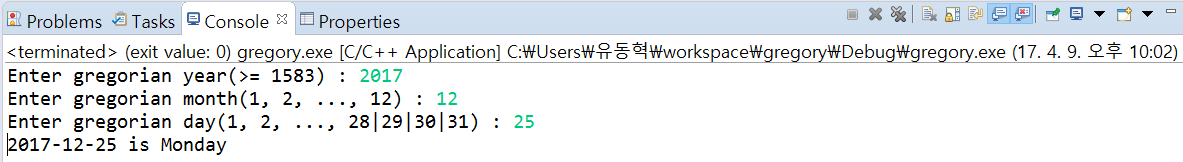
↓1583/03/31



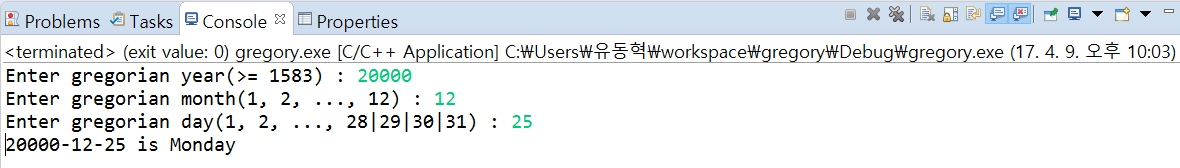
↓2017/04/06



↓2017/12/25



↓20000/12/25



* 1. **Analysis & Discussion**

- 코드를 너무 비효율적으로(길게) 짠 것 같다. 더 효율적으로 짜려면 어떻게 해야할까?

🡺 모든 ‘어떤 에러에도 해당되지 않는 경우’ 마다 printf를 넣지 않고 printf를 한번만 써야 할 것이다.

🡺 Jullian값 계산을 위한 y, m, d를 분리하지않고 year, month, day선에서 끝내야 할 것이다.

🡺 분리된 if문들의 조건들의 개수를 줄여야 할 것 이다.