

달력 출력하기(과제11번) 문제 해설

20243108 장민주

달력 출력하기

그레고리언 방법으로 달력을 제작할 경우 주어진 달(몇년 몇월)에 해당하는 달력을 출력하는 프로그램을 작성하시오. 그레고리언 방법이 제정된 1582년 1월 1일은 금요일이라고 가정한다.

입력

입력 파일의 이름은 "input.txt" 이다. 입력은 t 개의 테스트 케이스로 주어진다. 입력 파일의 첫 번째 줄에 테스트 케이스의 개수를 나타내는 정수 t 가 주어진다. 두 번째 줄부터 t 개의 줄에는 한 줄에 한 개의 테스트 케이스에 해당하는 두 개의 정수 $Y M$ 이 입력된다. 입력되는 첫 번째 정수 Y ($1582 \leq Y \leq 99999$) 는 년도를 나타내고, 두 번째 정수 M ($1 \leq M \leq 12$) 은 월을 나타낸다. 두 정수 사이에는 한 개의 공백이 있으며, 잘못된 데이터가 입력되는 경우는 없다.

출력

출력은 표준출력(standard output)을 사용한다. 입력되는 테스트 케이스의 순서대로 다음 줄에 이어서 각 테스트의 결과를 출력한다. 각 테스트 케이스에 해당하는 출력의 첫 줄에는 입력되는 달의 년도와 달을 나타내는 정수를 입력 형식과 동일하게 출력한다. 그 다음 줄부터는 한 줄에 한 주에 해당하는 날짜를 일요일부터 토요일까지 차례대로 출력한다. 단, 첫 줄에서 주어진 달의 첫째 날의 요일이 일요일이 아닌 경우에는 첫째 날의 요일 전까지 해당하는 요일에 모두 0 을 출력한다. 또한 마지막 날짜의 요일이 토요일이 아닌 경우에는 마지막 날의 요일 다음부터 토요일까지 모두 0 을 출력한다. 따라서, 한 주에 관한 데이터를 출력하는 한 개의 줄에는 모두 7 개의 정수가 출력된다.

1. 그 달이 며칠까지 있는지

-> 윤년 계산 필요

2. 그 달의 1일이 무슨 요일인지

-> zeller 공식 이용

```
#include <iostream>
#include <fstream>
using namespace std;
```

1. 그 달이 며칠까지 있는지 계산

```
int calculateEndDay(int year, int month) {
    if (month == 2) {
        if ((year % 4 == 0 && year % 100 != 0) || (year % 400 == 0)) return 29;
        else return 28;
    } else if (month == 4 || month == 6 || month == 9 || month == 11) return 30;
    return 31;
}
```

2. 무슨 요일인지 계산

```
int calculateDayOfWeek(int year, int month, int day) {
    if (month == 1 || month == 2) {
        year--;
        month += 12;
    }

    int k = year % 100;
    int j = year / 100;
    // 0 = 토요일, 1 = 일요일, ..., 6 = 금요일
    int result = (day + 13*(month+1)/5 + k + k/4 + j/4 + 5*j) % 7;

    // 0 = 일요일, 1 = 월요일, ..., 6 = 토요일로 변환
    return (result+6)%7;
}
```

```
int main() {    메인 함수
    ifstream inputFile("input.txt");
```

```
    int t;
    inputFile >> t;
```

```
    for (int i = 0; i < t; i++) {
        int Y,M;
        inputFile >> Y >> M;
```

```
        cout << Y << " " << M << "\n";
```

```
        int startDayOfWeek = calculateDayOfWeek(year: Y, month: M, day: 1);
```

```
        int endDay = calculateEndDay(year: Y, month: M);
```

```
        int Day = 1;
```

```
        while (true) {
```

```
            for (int j = 0; j < 7; j++) {
```

```
                if (j > 0) cout << " ";
```

```
                if (--startDayOfWeek >= 0) cout << 0;
```

```
                else if (Day > endDay) cout << 0;
```

```
                else cout << Day++;
```

```
            }
```

```
            cout << "\n";
```

```
            if (Day > endDay) break;
```

```
        }
```

```
    }
}
```

1. 그 달이 며칠까지 있는지 계산

```
int calculateEndDay(int year, int month) {  
    if (month == 2) {  
        if ((year % 4 == 0 && year % 100 != 0) || (year % 400 == 0)) return 29;  
        else return 28;  
    } else if (month == 4 || month == 6 || month == 9 || month == 11) return 30;  
    return 31;  
}
```

2. 특정 날짜가 무슨 요일인지 계산

```
int calculateDayOfWeek(int year, int month, int day) {  
    if (month == 1 || month == 2) {  
        year--;  
        month += 12;  
    }  
  
    int k = year % 100;  
    int j = year / 100;  
    // 0 = 토요일, 1 = 일요일, ..., 6 = 금요일  
    int result = (day + 13*(month+1)/5 + k + k/4 + j/4 + 5*j) % 7;  
  
    // 0 = 일요일, 1 = 월요일, ..., 6 = 토요일로 변환  
    return (result+6)%7;  
}
```

2. 특정 날짜가 무슨 요일인지 계산

<Zeller의 공식>

- $h = (d + 13(m + 1)/5 + K + K/4 + J/4 + 5J) \bmod 7$
- h: 요일 결과 (0=토요일, 1=일요일, ..., 6=금요일)
- d: 일 (day)
- m: 월 (month)
→ 3=3월, ..., 12=12월, 1=1월, 2=2월은 작년의 13월, 14월로 간주
- K: year % 100 (세기 안의 연도, 즉 연도의 뒷 두 자리)
- J: year / 100 (세기, 즉 연도의 앞 두 자리)

2. 특정 날짜가 무슨 요일인지 계산

```
int calculateDayOfWeek(int year, int month, int day) {  
    if (month == 1 || month == 2) {  
        year--;  
        month += 12;  
    }  
  
    int k = year % 100;  
    int j = year / 100; Zeller 공식  
    // 0 = 토요일, 1 = 일요일, ..., 6 = 금요일  
    int result = (day + 13*(month+1)/5 + k + k/4 + j/4 + 5*j) % 7;  
  
    // 0 = 일요일, 1 = 월요일, ..., 6 = 토요일로 변환  
    return (result+6)%7;  
}
```

2. 특정 날짜가 무슨 요일인지 계산

```
int calculateDayOfWeek(int year, int month, int day) {  
    if (month == 1 || month == 2) {  
        year--;  
        month += 12;  
    }  
  
    int k = year % 100;  
    int j = year / 100;  
    // 0 = 토요일, 1 = 일요일, ..., 6 = 금요일  
    int result = (day + 13*(month+1)/5 + k + k/4 + j/4 + 5*j) % 7;  
  
    // 0 = 일요일, 1 = 월요일, ..., 6 = 토요일로 변환  
    return (result+6)%7;  
}
```



```

#include <iostream>
#include <fstream>
using namespace std;

int calculateEndDay(int year, int month) {
    if (month == 2) {
        if ((year % 4 == 0 && year % 100 != 0) || (year % 400 == 0)) return 29;
        else return 28;
    } else if (month == 4 || month == 6 || month == 9 || month == 11) return 30;
    return 31;
}

int calculateDayOfWeek(int year, int month, int day) {
    if (month == 1 || month == 2) {
        year--;
        month += 12;
    }

    int k = year % 100;
    int j = year / 100;
    // 0 = 토요일, 1 = 일요일, ..., 6 = 금요일
    int result = (day + 13*(month+1)/5 + k + k/4 + j/4 + 5*j) % 7;

    // 0 = 일요일, 1 = 월요일, ..., 6 = 토요일로 변환
    return (result+6)%7;
}

```

```

int main() {
    ifstream inputFile("input.txt");

    int t;
    inputFile >> t;

    for (int i = 0; i < t; i++) {
        int Y,M;
        inputFile >> Y >> M;

        cout << Y << " " << M << "\n";

        int startDayOfWeek = calculateDayOfWeek(year: Y, month: M, day: 1);
        int endDay = calculateEndDay(year: Y, month: M);
        int Day = 1;
        while (true) {
            for (int j = 0; j < 7; j++) {
                if (j > 0) cout << " ";

                if (--startDayOfWeek >= 0) cout << 0;
                else if (Day > endDay) cout << 0;
                else cout << Day++;
            }
            cout << "\n";

            if (Day > endDay) break;
        }
    }
}

```

달력 출력하기

그레고리언 방법으로 달력을 제작할 경우 주어진 달(몇년 몇월)에 해당하는 달력을 출력하는 프로그램을 작성하시오. 그레고리언 방법이 제정된 1582년 1월 1일은 금요일이라고 가정한다.

입력

입력 파일의 이름은 "input.txt" 이다. 입력은 t 개의 테스트 케이스로 주어진다. 입력 파일의 첫 번째 줄에 테스트 케이스의 개수를 나타내는 정수 t 가 주어진다. 두 번째 줄부터 t 개의 줄에는 한 줄에 한 개의 테스트 케이스에 해당하는 두 개의 정수 $Y\ M$ 이 입력된다. 입력되는 첫 번째 정수 Y ($1582 \leq Y \leq 99999$)는 년도를 나타내고, 두 번째 정수 M ($1 \leq M \leq 12$)은 월을 나타낸다. 두 정수 사이에는 한 개의 공백이 있으며, 잘못된 데이터가 입력되는 경우는 없다.

출력

출력은 표준출력(standard output)을 사용한다. 입력되는 테스트 케이스의 순서대로 다음 줄에 이어서 각 테스트의 결과를 출력한다. 각 테스트 케이스에 해당하는 출력의 첫 줄에는 입력되는 달의 년도와 달을 나타내는 정수를 입력 형식과 동일하게 출력한다. 그 다음 줄부터는 한 줄에 한 주에 해당하는 날짜를 일요일부터 토요일까지 차례대로 출력한다. 단, 첫 줄에서 주어진 달의 첫째 날의 요일이 일요일이 아닌 경우에는 첫째 날의 요일 전까지 해당하는 요일에 모두 0 을 출력한다. 또한 마지막 날짜의 요일이 토요일이 아닌 경우에는 마지막 날의 요일 다음부터 토요일까지 모두 0 을 출력한다. 따라서, 한 주에 관한 데이터를 출력하는 한 개의 줄에는 모두 7 개의 정수가 출력된다.

```
int main() {
    ifstream inputFile("input.txt");

    int t;
    inputFile >> t;

    for (int i = 0; i < t; i++) {
        int Y, M;
        inputFile >> Y >> M;

        cout << Y << " " << M << "\n";

        int startDayOfWeek = calculateDayOfWeek(year: Y, month: M, day: 1);
        int endDay = calculateEndDay(year: Y, month: M);
        int Day = 1;
        while (true) {
            for (int j = 0; j < 7; j++) {
                if (j > 0) cout << " ";

                if (--startDayOfWeek >= 0) cout << 0;
                else if (Day > endDay) cout << 0;
                else cout << Day++;
            }
            cout << "\n";

            if (Day > endDay) break;
        }
    }
}
```

입력	출력
3	1582 1
1582 1	0 0 0 0 0 1 2
2003 9	3 4 5 6 7 8 9
99999 12	10 11 12 13 14 15 16
	17 18 19 20 21 22 23
	24 25 26 27 28 29 30
	31 0 0 0 0 0 0
	2003 9
	0 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 0 0 0 0
	99999 12
	0 0 0 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 0

- startDayOfWeek = 3
- endDay = 31

```
int main() {
    ifstream inputFile("input.txt");

    int t;
    inputFile >> t;

    for (int i = 0; i < t; i++) {
        int Y,M;
        inputFile >> Y >> M;

        cout << Y << " " << M << "\n";

        int startDayOfWeek = calculateDayOfWeek(year: Y, month: M, day: 1);
        int endDay = calculateEndDay(year: Y, month: M);
        int Day = 1;
        while (true) {
            일주일 출력
            for (int j = 0; j < 7; j++) {
                if (j > 0) cout << " ";

                if (--startDayOfWeek >= 0) cout << 0;
                else if (Day > endDay) cout << 0;
                else cout << Day++;
            }
            cout << "\n";

            if (Day > endDay) break;
        }
    }
}
```