

C1 - D 日期统计

题目描述

本题中所使用的纪年体系为公元，仅考虑年份位于 $[2000, 9999]$ 的日期。

所有日期均可以表示为一个唯一的 8 位十进制数字，前四位表示年，中间两位表示月，最后两位表示日，例如 2022 年 9 月 6 日可以表示为 20220906。

一个日期是好的当且仅当这个日期按上述规则表示为 8 位十进制数字 $a_1a_2a_3a_4a_5a_6a_7a_8$ 时满足下面两个条件中至少一条：

- $\exists i \in \{1, 2, \dots, 8\}$ 使得 $\forall 1 \leq j < i, a_j \leq a_{j+1}$ 并且 $\forall i < j \leq 8, a_j \leq a_{j-1}$ 。
- $\exists i \in \{1, 2, \dots, 8\}$ 使得 $\forall 1 \leq j < i, a_j \geq a_{j+1}$ 并且 $\forall i < j \leq 8, a_j \geq a_{j-1}$ 。

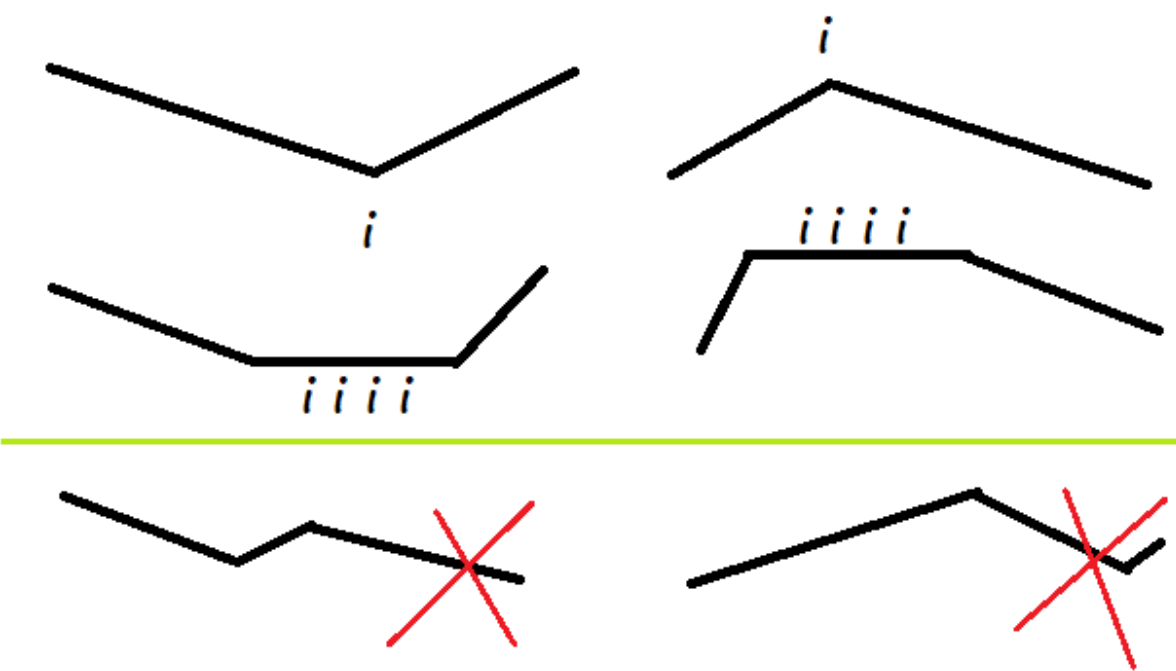
比如 20000123 就是一个好的日期。小水獭想知道对于给定两个日期，这两个日期之间（包含这两个日期）共有多少个好日期。

题解思路

首先分析题目要求，是要统计日期之间满足条件的好日期个数，判断条件也不复杂，因此大概是线性的时间复杂度。结合数据范围，8000 年也没有太多天数，因此是可以接受的。

这是一道日期相关的题，那么我们可以构造一个简单的日期类型来支持日期的相关运算。题目中日期范围不大，只需要完成日期遍历即可，因此实现基本的赋值、递增、相等判断即可。

对于好日期的判断，则是将日期拆分为 8 个数字，判断其之间的关系。通过题中的描述，不难看出好日期有如下特点，在 i 两侧均（不严格）单调。



因此，将日期转化为数组后，分两种情况，从左右两侧分别寻找单调区间。如果两区间有重合，那么就是好日期；如果无重合，则不是好日期。注，这里的重合指至少有相同的区间端点。

代码

```
#include <cstdio>

const int DAY_OF_MONTH[] = {
    0, 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31
};

class Date
{
public:
    void Init(int year, int month, int day)
    {
        // Assume that it is always legal.
        _year = year;
        _month = month;
        _day = day;

        _checkLeap();
    }

    void Read()
    {
        scanf("%4d%2d%2d", &_year, &_month, &_day);
        _checkLeap();
    }

    bool IsLeap() { return _isLeap; }

    void Convert(int* arr)
    {
        int date = _year;
        date = date * 100 + _month;
        date = date * 100 + _day;

        for (int i = 0; i < 8; i++)
        {
            arr[i] = date % 10;
            date /= 10;
        }
    }

    Date& operator=(Date& date)
    {
        _year = date._year;
        _month = date._month;
        _day = date._day;
        _isLeap = date._isLeap;

        return *this;
    }

    bool operator==(Date& date)
    {

```

```

        return ((_day == date._day) && (_month == date._month) && (_year ==
date._year));
    };

    bool operator!=(Date& date)
    {
        return !operator==(date);
    };

    Date& operator++()
    {
        int days = DAY_OF_MONTH[_month];
        if (_isLeap && _month == 2)
            days++;

        _day++;
        if (_day > days)
        {
            _day = 1;
            _month++;
            if (_month > 12)
            {
                _month = 1;
                _year++;
                _CheckLeap();
            }
        }

        return *this;
    };

private:
    void _CheckLeap()
    {
        if (_year % 100 == 0)
            _isLeap = (_year % 400 == 0) ? true : false;
        else
            _isLeap = (_year % 4 == 0) ? true : false;
    }

    int _year;
    int _month;
    int _day;
    bool _isLeap;
};

bool IsGoodDate(int* date);
int main()
{
    int arr[12];
    int cnt;
    Date begin;
    Date end;

    begin.Read();

```

```

end.Read();
++end;

cnt = 0;
for (Date date = begin; date != end; ++date)
{
    date.Convert(arr);
    if (IsGoodDate(arr))
        cnt++;
}

printf("%d\n", cnt);

return 0;
}

bool IsGoodDate(int* date)
{
    int left, right;

    // Condition 1: (\/)
    left = 7;
    while ((left > 0) && (date[left] >= date[left - 1]))
        left--;
    right = 0;
    while ((right < 7) && (date[right] >= date[right + 1]))
        right++;
    if (left <= right)
        return true;

    // Condition 2: (/)
    left = 7;
    while ((left > 0) && (date[left] <= date[left - 1]))
        left--;
    right = 0;
    while ((right < 7) && (date[right] <= date[right + 1]))
        right++;
    if (left <= right)
        return true;

    return false;
}

```

AUTHOR: 21371300 柳政尧