

```
1  /*
2  给你一个日期，请你设计一个算法来判断它是对应一周中的哪一天。
3
4  输入为三个整数：day、month 和 year，分别表示日、月、年。
5
6  您返回的结果必须是这几个值中的一个 {"Sunday", "Monday", "Tuesday", "Wednesday",
7  "Thursday", "Friday", "Saturday"}。
8
9
10 示例 1:
11
12 输入: day = 31, month = 8, year = 2019
13 输出: "Saturday"
14
15 示例 2:
16
17 输入: day = 18, month = 7, year = 1999
18 输出: "Sunday"
19
20 示例 3:
21
22 输入: day = 15, month = 8, year = 1993
23 输出: "Sunday"
24
25
26
27 提示:
28
29 给出的日期一定是在 1971 到 2100 年之间的有效日期。
30
31 来源：力扣（LeetCode）
32 链接：https://leetcode-cn.com/problems/day-of-the-week
33 著作权归领扣网络所有。商业转载请联系官方授权，非商业转载请注明出处。
34 */
```

分析:

方法一:造个轮子:

- 1970.1.1是周四,计算从1970.1.1到当日的天数差,然后对7取余即可.

方法一:C++_造轮子

```
1  class solution
2  {
3
4      private:
5          const int month_sum[2][12] =
6          {
7              {0, 31, 59, 90, 120, 151, 181, 212, 243, 273, 304, 334},
```

```

8         {0,31,60,91,121,152,182,213,244,274,305,335}
9     };
10
11
12     int isLeapYear(int year)
13     {
14         //判断是否是闰年
15         if( (year%4==0 && year%100!=0)
16             ||(year%400 == 0)
17         )
18         {
19             return 1;
20         }
21         else
22         {
23             return 0;
24         }
25     }
26
27
28     public:
29     string dayOfTheWeek(int day, int month, int year)
30     {
31         string s[] = {"Sunday", "Monday", "Tuesday", "wednesday",
32 "Thursday", "Friday", "Saturday"};
33         int leaves = 4; /*1970年1月1日 周4*/
34         int i = 0;
35
36         for( i = 1971 ; i < year ; i++)
37         {
38             if(isLeapYear(i))
39             {
40                 leaves += 2;
41             }
42             else
43             {
44                 leaves += 1;
45             }
46         }
47
48         leaves += month_sum[isLeapYear(year)][month-1];
49         leaves += day;
50
51         return s[leaves%7];
52     }
53 };
54
55 /*
56 执行结果:
57 通过
58 显示详情
59 执行用时 :0 ms, 在所有 cpp 提交中击败了100.00% 的用户
60 内存消耗 :8 MB, 在所有 cpp 提交中击败了100.00%的用户
61 */

```

