DesignCrowd Technical Challenge

Hi!

Thanks for taking the time to do our technical challenge. The goal of this challenge is for you to demonstrate:

- How you approach solution design;
- What production quality code means to you;
- Your ability to implement the correct solution to a spec;
- The tools, frameworks and language features you think are most suitable to use;

This challenge is concerned with counting the number of days between two dates.

Good luck! We're keen to see what you come up with.

As a starting point, create a class that matches the following:

```
public class BusinessDayCounter
{
    public int WeekdaysBetweenTwoDates(DateTime firstDate, DateTime secondDate)
    {
        //todo
    }

    public int BusinessDaysBetweenTwoDates(DateTime firstDate, DateTime secondDate,
IList<DateTime> publicHolidays)
    {
        //todo
    }
}
```

You will use this scaffold as the basis for building the solution to the following tasks.

Task One: Weekdays Between Two Dates

Calculates the number of weekdays in between two dates.

- Weekdays are Monday, Tuesday, Wednesday, Thursday, Friday.
- The returned count should not include either firstDate or secondDate -
  e.g. between Monday 07-Oct-2013 and Wednesday 09-Oct-2013 is one weekday.
- If secondDate is equal to or before firstDate, return 0.

Test Data

The following scenarios will allow you to validate your application is working as expected:

Start Date	End Date	Result
7th October 2013	9th October 2013	1
5th October 2013	14th October 2013	5
7th October 2013	1st January 2014	61
7th October 2013	5th October 2013	0

Task Two: Business Days Between Two Dates

Calculate the number of business days in between two dates.

- Business days are Monday, Tuesday, Wednesday, Thursday, Friday, but excluding any dates which appear in the supplied list of public holidays.
- The returned count should not include either firstDate or secondDate - e.g. between Monday 07-Oct-2013 and Wednesday 09-Oct-2013 is one weekday.
- If secondDate is equal to or before firstDate, return 0.

Test Data

Sample list of Public Holidays:

- 25th December 2013
- 26th December 2013
- 1st January 2014

Given those public holidays, the following scenarios will allow you to validate your application is working as expected:

Start Date	End Date	Result
7th October 2013	9th October 2013	1
24th December 2013	27th December 2013	0
7th October 2013	1st January 2014	59

Task Three: More Holidays

Design a data structure or hierarchy of structures which can define public holidays in a more complex fashion than simple dates.

This should cater for things such as:

- Public holidays which are always on the same day, e.g. Anzac Day on April 25th every year.
- Public holidays which are always on the same day, except when that falls on a weekend. e.g. New Year's Day on January 1st every year, unless that is a Saturday or Sunday, in which case the

holiday is the next Monday.
- Public holidays on a certain occurrence of a certain day in a month. e.g. Queen's Birthday on the second Monday in June every year.

Given this data structure, the BusinessDaysBetweenTwoDates() function should be able to be extended to take a list of public holiday rules, rather than a list of DateTimes, and calculate the number of business days between two dates using those rules to define public holidays.

Submission

Submit your solution as a git repo on Github, Bitbucket or similar. If you would like to keep it private, please contact us and we'll let you know some email addresses to share with, otherwise just make it public.