CALIFORNIA STATE POLYTECHNIC UNIVERSITY
Computer Science Department
CS 1400 02 (4) ***T. Nguyen/F23***

PROJECT: **3**
DUE DATE: **October 19, 2022**

Description:

Write a Java application named **MonthlyCalendarGenerator** that outputs a monthly calendar for a given month and year into a file.

1. For the calendar output, the line is terminated at the last character, no extra space(s).
2. The last line MUST have a terminating new line.
3. You must implement and use the following methods:

```
int getDayOfTheWeek (int day, int month, int year)
         return the day of the week, 0:sun, 1:mon, …, 6:sat
boolean isLeapYear (int year)
         return true if year is a leap year
int getNumberOfDaysInMonth (int month, int year)
         return the number of days in a given month and year
String getMonthName (int month)
         return the name of the month, 1:January, 2:February, …, 12:December
```

Required I/O:

```
Calendar by F. Last

Enter month year ? 11 2023

11-2023.txt generated.
```

Where F. Last is your last name.

Output file 11-2023.txt:

```
November 2023¶
¶
Sun··Mon··Tue··Wed··Thu··Fri··Sat¶
---------------------------------¶
·················1····2····3····4¶
··5····6····7····8····9···10···11¶
·12···12···14···15···16···18···18¶
·19···20···21···22···23···24···25¶
·26···27···28···29···30···31¶
```

Where: · is the space character, ¶ is the new line character.

Turn in

Submit the source code to Canvas.

Notes:

The following information is required at the beginning of every source file.

```
//
//    Name:        Last, First
//    Project:     #
//    Due:         date
//    Course:      cs-1400-02-f23
//
//    Description:
//                 A brief description of the project.
//
```

Hints:

Calendar related information:

Leap year: *year is divisible by 4 and not divisible by 100 or divisible by 400*

Day of the week:

$$a = \frac{14 - month}{12}$$

$$y = year - a$$

$$m = month + 12a - 2$$

For Gregorian calendar: $$d = \left(day + y + \frac{y}{4} - \frac{y}{100} + \frac{y}{400} + \frac{31m}{12}\right) \bmod 7$$

The value of d is 0 for a Sunday, 1 for a Monday, 2 for a Tuesday, etc.

Pseudocode:

```
1. Prompt for the month and year
2. Compute the day of the week for the 1st of the month (getDayOfTheWeek)
3. Compute the number of days for the month (getNumberOfDaysInMonth)
4. Open output file
5. Output the month name (getMonthName)
6. Output the day name header (Sun …. Sat) and dash line
7. Using day of the week to skip to the first day of the month
8. For the number of days in the month, output each day using print
9. Use a counter to break to a new week using println (every 7 days, output a newline)
10. Close output file
```