

Programming for Engineers (EG1302)

PROJECT # 1

Write a program that prompts the user for any year higher or equal to 1900 and prints its monthly calendar. The calendar runs through the entire year displaying each month in a format that looks like the following:

```

          February 1992
Su  Mo  Tu  We  Th  Fr  Sa
                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
```

Task 1: Your program must display the data in a form in which everything lines up correctly on the screen. Your program should be able to determine the day of the week when the first day of the month starts. To determine the first day of the month for a given year, you need to mark a reference day in history and count from that date. For this, use January 1, 1900, which fell on a Monday. For every year since then, you need to add 365 or 366 days, depending on whether the year was a leap year.

Note: A year is a leap year if one of the following conditions hold:

- The year is divisible by 4 but not divisible by 100, or
- The year is divisible by 400.

Task 2: Give the user the option to print the calendar with one month per row, two months per row, or three months per row.

Example of one-month per row calendar:

January 2005						
Su	Mo	Tu	We	Th	Fr	Sa
						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					

February 2005						
Su	Mo	Tu	We	Th	Fr	Sa
		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					

Example of two-month per row calendar:

January 2005							February 2005						
Su	Mo	Tu	We	Th	Fr	Sa	Su	Mo	Tu	We	Th	Fr	Sa
						1			1	2	3	4	5
2	3	4	5	6	7	8	6	7	8	9	10	11	12
9	10	11	12	13	14	15	13	14	15	16	17	18	19
16	17	18	19	20	21	22	20	21	22	23	24	25	26
23	24	25	26	27	28	29	27	28					
30	31												

Example of three-months per row:

January 2005							February 2005							March 2005						
Su	Mo	Tu	We	Th	Fr	Sa	Su	Mo	Tu	We	Th	Fr	Sa	Su	Mo	Tu	We	Th	Fr	Sa
						1			1	2	3	4	5			1	2	3	4	5
2	3	4	5	6	7	8	6	7	8	9	10	11	12	6	7	8	9	10	11	12
9	10	11	12	13	14	15	13	14	15	16	17	18	19	13	14	15	16	17	18	19
16	17	18	19	20	21	22	20	21	22	23	24	25	26	20	21	22	23	24	25	26
23	24	25	26	27	28	29	27	28						27	28	29	30	31		
30	31																			