CS1010: LAB ASSIGNMENT 3

0 INTRODUCTION

This lab contains 2 exercises.

To receive the attempt mark for this lab assignment, you must submit all programs and get a passing feedback mark for each of the program.

If you have any questions on the task statements, you may post your queries on the relevant IVLE discussion forum. However, do not post your programs (partial or complete) on the forum before the deadline!

Please be reminded that lab assignments must be done in your own effort.

Mark allocation is as follows (60% correctness, 20% style, 20% design):

• Exercise 1: 2 marks

• Exercise 2: 5 marks

1 EXERCISE 1: MATRIX ROTATION

1.1 LEARNING OBJECTIVES

- Using repetition statement.
- Learning basic operation to matrix.
- Applying neat logic in problem solving.

1.2 TASK STATEMENT

Given a 2 dimensional matrix of size N x N, you are required to rotate the matrix by 90 degree clockwise and output the result.

The first line of the input file is an integer, N (1 \leq N \leq 20). The following is a 2 dimensional array with size N x N. The output is the array after being rotated by 90 degree clockwise. For example, given a 3 dimensional array:

123

456

789

After rotation of 90 degree clockwise, it becomes:

741

852

963

1.3 SAMPLE RUNS

Sample run using interactive input (user's input shown in blue; output shown in bold purple). Note that the first two lines (in green below) are commands issued to compile and run your program on UNIX.

Sample run #1:

```
$ gcc -Wall rotation.c -o rotation
$ ./rotation
4
2 4 1 3
3 5 8 1
2 9 4 6
6 4 2 8
6 2 3 2
4 9 5 4
```

```
2481
8613
```

Sample run #2:

```
$ gcc -Wall rotation.c -o rotation
$ ./rotation
2
5 9
4 1
4 5
1 9
```

1.4 SKELETON PROGRAM AND TEST DATA

The skeleton program is provided: rotation.c

Test input: Input Files

Test output: Output Files

1.5 IMPORTANT NOTES

- Write a function rotate() to rotate the matrix and a function print_matrix() to print the result.
- In writing functions, we would like you to include function prototypes before the main() function, and the function definitions after the main() function.
- This is a problem—solving task where we look for neat logic in your program. Using descriptive variable names, and adding appropriate comments will help the readers (and yourself) to understand the logic better.

1.6 ESTIMATED DEVELOPMENT TIME

The time here is an estimate of how much time we expect you to spend on this exercise. If you need to spend way more time than this, it is an indication that some help might be needed.

- Devising and writing the algorithm (pseudo-code): 15 minutes
- Checking/tracing the algorithm: 10 minutes
- Translating pseudo-code into code: 5 minutes
- Typing in the code: 5 minutes
- Testing and debugging: 5 minutes
- Total: 40 minutes

2 EXERCISE 2: PRINT THE CALENDAR

2.1 LEARNING OBJECTIVES

- Using repetition statements.
- Using 3 dimensional array
- Using special values to deal with special cases
- Printing in a given format
- Problem solving.

2.2 TASK STATEMENT

In this task, you are required to print the calendar of all the months in a randomly given year. Please refer to Lab 2 exercise 3 for the format of one month. But to make every month have the same size, we assume every month has eight lines. The first line is the name of the month, which is above "Wed". We use three-letter abbreviation to represent the names of the months. They are "JAN", "FEB", "MAR", "APR", "MAY", "JUN", "JUL", "AUG", "SEP", "OCT", "NOV", "DEC" respectively. The second line is "Sun Mon Tue Wed Thu Fri Sat". Then follows the dates in that month. If the dates only occupy 4 lines, fill the

remaining two lines with blank spaces (see Figure 2.1). If the dates only occupy 5 lines, fill the remaining one line with blank spaces(see Figure 2.2). If the dates occupy all the 6 lines, fill the remaining part in last line with blank spaces (see Figure 2.3).

												F	Е	В												
S	u	n		M	0	n		Т	u	е		W	е	d		Τ	h	u		F	r	i		S	a	t
		1				2				3				4				5				6				7
		8				9			1	0			1	1			1	2			1	3			1	4
	1	5			1	6			1	7			1	8			1	9			2	0			2	1
	2	2			2	3			2	4			2	5			2	6			2	7			2	8

Figure 2.1

												А	Р	R												
S	u	n		M	О	n		Т	u	е		W	е	d		Т	h	u		F	r	i		S	a	t
						1				2				3				4				5				6
		7				8				9			1	0			1	1			1	2			1	3
	1	4			1	5			1	6			1	7			1	8			1	9			2	0
	2	1			2	2			2	3			2	4			2	5			2	6			2	7
	2	8			2	9			3	0																

Figure 2.2

												J	Α	N												
S	u	n		М	О	n		Τ	u	е		W	е	d		Τ	h	u		F	r	i		S	a	t
																										1
		2				3				4				5				6				7				8
		9			1	0			1	1			1	2			1	3			1	4			1	5
	1	6			1	7			1	8			1	9			2	0			2	1			2	2
	2	3			2	4			2	5			2	6			2	7			2	8			2	9
	3	0			3	1																				

Figure 2.3

You are required to print the calendar in both horizontal and vertical format. For horizontal version, print JAN, FEB and MAR in the first row, and APR, MAY and

JUN in the second row, so on and so forth. Here is an example: FEB Wed Sun 1 8 15 22 11 18 25 11 18 25 15 22 29 13 20 27 21 28 MAY Wed 16 23 30 17 24 17 24 12 19 26 14 21 28 14 21 28 20 27 22 29 18 25 20 27 22 29 21 28 22 29 23 30 Wed 2 9 16 23 30 Thu 2 9 16 23 30 Tue 1 8 15 22 29 18 25 15 22 29 11 18 25 12 19 26 13 20 27 14 21 28 16 23 30 15 22 29 17 24 31 20 27 21 28 20 27 NOV Wed Thu 5 12 19 26 Fri 6 13 20 27 11 18 25

For vertical version, print JAN, FEB, MAR and APR in the first column, and MAY, JUN, JUL and AUG in the second column, so on and so forth. Here is an example:

										201	5									
			JAN							MAY	_						SEP			
Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat
B				1	2	3						1	2			1	2	3	4	5
4	5	6	7	8	9	10	3	4	5	6	7	8	9	6	7	8	9	10	11	12
11	12	13	14	15	16	17	10	11	12	13	14	15	16	13	14	15	16	17	18	19
18	19	20	21	22	23	24	17	18	19	20	21	22	23	20	21	22	23	24	25	26
25	26	27	28	29	30	31	24	25	26	27	28	29	30	27	28	29	30			10000
							31													
			FEB							JUN							OCT			
Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat
1	2	3	4	5	6	7		1	2	3	4	5	6					1	2	3
8	9	10	11	12	13	14	7	8	9	10	11	12	13	4	5	6	7	8	9	10
15	16	17	18	19	20	21	14	15	16	17	18	19	20	11	12	13	14	15	16	17
22	23	24	25	26	27	28	21	22	23	24	25	26	27	18	19	20	21	22	23	24
							28	29	30					25	26	27	28	29	30	31
			MAR							JUL							NOV			
Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat
1	2	3	4	5	6	7				1	2	3	4	1	2	3	4	5	6	7
8	9	10	11	12	13	14	5	6	7	8	9	10	11	8	9	10	11	12	13	14
15	16	17	18	19	20	21	12	13	14	15	16	17	18	15	16	17	18	19	20	21
22	23	24	25	26	27	28	19	20	21	22	23	24	25	22	23	24	25	26	27	28
29	30	31					26	27	28	29	30	31		29	30					
			APR	-						AUG	-						DEC			
Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat
200			1	2	3	4							1			1	2	3	4	5
5	6	7	8	9	10	11	2	3	4	5	6	7	8	6	7	8	9	10	11	12
12	13	14	15	16	17	18	9	10	11	12	13	14	15	13	14	15	16	17	18	19
19	20	21	22	23	24	25	16	17	18	19	20	21	22	20	21	22	23	24	25	26
26	27	28	29	30			23	24	25	26	27	28	29	27	28	29	30	31		
							30	31												

Between two months in the horizontal direction, there is a Tab to separate them. The first line is the year. There are seven Tabs in front of and behind the year. See following figure.

_										201	5									
			JAN							FEB				44			MAR			
Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat
				1	2	3	1	2	3	4	5	6	7	7XTah1	2	3	4	5	6	7
4	5	6	7	8	9	10	7XTah8	9	10	11	12	13	14	8	9	10	11	12	13	14
11	12	13	14	15	16	17	15	16	17	18	19	20	21	15	16	17	18	19	20	21
18	19	20	21	22	23	24	22	23	24	25	26	27	28	22	23	24	25	26	27	28
25	26	27	28	29	30	31								29	30	31				
			APR							MAY							JUN			
Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat
20065			1	2	3	4						1	2		1	2	3	4	5	6
5	6	7	8	9	10	11	3	4	5	6	7	8	9	7	8	9	10	11	12	13
12	13	14	15	16	17	18	1XTab10	11	12	13	14	15	16	14	15	16	17	18	19	20
19	20	21	22	23	24	25	17	18	19	20	21	22	23	21	22	23	24	25	26	27
26	27	28	29	30			24	25	26	27	28	29	30	28	29	30				
							31													
_			JUL							AUG							SEP			
Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat
200		1000	1	2	3	4						100	1		2	1	2	3	4	5
5	6	7	8	9	10	11	2	3	4	5	6	7	8	6	7	8	9	10	11	12
12	13	14	15	16	17	18	9	10	11	12	13	14	15	13	14	15	16	17	18	19
19	20	21	22	23	24	25	16	17	18	19	20	21	22	20	21	22	23	24	25	26
26	27	28	29	30	31		23	24	25	26	27	28	29	27	28	29	30			
			OCT				30	31		NOW							DEC			
-		+	ОСТ	+4					+0.0	NOV	TF.					+.00	DEC	T L		C-4
Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat
4	-	-	-	1	2	3 10	1	2	3	4	5	6	7		-	1	2	3	4	5
4	5 12	6 13	7 14	8 15	9 16	17	8 15	9 16	10 17	11	12 19	13	14 21	6 13	7 14	8 15	9 16	10 17	11 18	12
18	19	20	21	22	23	24	22	23	24	25	26	27	28	20	21	22	23	24	25	19 26
25	26	27	28	29	30	31	29	30	24	25	20	21	28	27	28	29	30	31	23	20
23	20	21	20	29	30	31	29	30						21	28	29	30	31		
							, III				*									

The first line of the input file is the year, follows printing method (in vertical or horizontal way). Input "vertical" or "horizontal" to choose the printing way.

Hint:

- 1. You may want to store all the months into a 3 dimensional array calendar[12][6][7]. calendar[0][0...5][0...6] stores the dates of January, calendar[1][0...5][0...6] stores the dates of February, so on and forth. And then print them.
- 2. You can use the functions in lab2 exercise3 to write the dates into arrays.
- 3. You are free to introduce additional functions if you deem it necessary. This must be supported by well-thought-out reasons, not a haphazard decision.
- 4. In writing functions, please put function prototypes before the main() function, and the function definitions after the main() function.

Remark: The easier way to solve this problem is to use a 3-d array. Actually it can also be solved using a 2-d array. Can you think of how to do that? (Hint: you can use an array of size 24x21 to store all the dates in the year, i.e., map the 3-d array calendar_3d[12][6][7] into the 2-d array calendar_2d[24][21]. and then change the index of the array so that you can set the dates in different month. For example, if you are using the horizontal way to print the month, the dates of June are stored in calendar_3d[5][0...5][0...6] and after being mapped into 2-d array, it becomes calendar_2d[6x1...6x1+5][7x2...7x2+6].)

2.3 SAMPLE RUNS

Sample run using interactive input

Sample run #1:

										197	9									
			JAN							FEB							MAR			
un	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Si
	1	2	3	4	5	6					1	2	3					1	2	
7	8	9	10	11	12	13	4	5	6	7	8	9	10	4	5	6	7	8	9	
4	15	16	17	18	19	20	11	12	13	14	15	16	17	11	12	13	14	15	16	
1	22	23	24	25	26	27	18	19	20	21	22	23	24	18	19	20	21	22	23	
28	29	30	31				25	26	27	28				25	26	27	28	29	30	
			APR							MAY							JUN			
n	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	S
1	2	3	4	5	6	7			1	2	3	4	5						1	
8	9	10	11	12	13	14	6	7	8	9	10	11	12	3	4	5	6	7	8	
5	16	17	18	19	20	21	13	14	15	16	17	18	19	10	11	12	13	14	15	
2	23	24	25	26	27	28	20	21	22	23	24	25	26	17	18	19	20	21	22	
9	30						27	28	29	30	31			24	25	26	27	28	29	
			JUL							AUG							SEP			
n	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	9
1	2	3	4	5	6	7				1	2	3	4							
3	9	10	11	12	13	14	5	6	7	8	9	10	11	2	3	4	5	6	7	
5	16	17	18	19	20	21	12	13	14	15	16	17	18	9	10	11	12	13	14	
2	23	24	25	26	27	28	19	20	21	22	23	24	25	16	17	18	19	20	21	
9	30	31					26	27	28	29	30	31		23	24	25	26	27	28	
														30						
			OCT							NOV							DEC			
1	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	9
	1	2	3	4	5	6					1	2	3							
7	8	9	10	11	12	13	4	5	6	7	8	9	10	2	3	4	5	б	7	
1	15	16	17	18	19	20	11	12	13	14	15	16	17	9	10	11	12	13	14	
1	22	23	24	25	26	27	18	19	20	21	22	23	24	16	17	18	19	20	21	
8	29	30	31				25	26	27	28	29	30		23	24	25	26	27	28	

Sample run #2:

Sun		uanfu vert		uanfu	-Opti	Plex-	9020:	~/Document	s/CS1	010/l	ab3/s	oluti	ons\$./cal	endar.						
Sun Mon Tue Wed Thu Fri Sat Sun Mon Tue Wed Thu Sat Sun Mon Tue Wed Thu Sat Sun Mon Sat San Mon Sat San Mon Sat San Mon Sat San											209	9									
The color of the				JAN							MAY							SEP			
4 5 6 7 8 9 10 11 12 13 14 15 16 17 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 24 25 26 27 28 29 30 31 29 30 29 30 29 30 29 30 29 30 29 30 29 30 29 30 29 30 29 30 29 30 29 30 29 30 29 30 31 29 30 31 29 30 31 29 30 31 29 30 31 29 30 31 29 30 31 29 30 31 29 30 31 29 30 31 29 30 31 29 30 31 30 30 30 30 30 30 30 30 30 30 30 30 30	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat
11 12 13 14 15 16 17 10 11 12 13 14 15 16 17 10 11 12 13 14 15 16 13 14 15 16 17 18 18 19 20 21 22 23 24 25 26 27 28 29 30 31 29 30 27 28 29 30 31 30 31 30					1	2	3						1	2			1	2	3	4	5
18	4	5	6	7	8	9	10	3	4	5	6	7	8	9	6	7	8	9	10	11	12
25	11	12	13	14	15	16	17	10	11	12	13	14	15	16	13	14	15	16	17	18	19
Sun Mon Tue Wed Tu Fri Sat Sun S	18	19	20	21	22	23	24	17	18	19	20	21	22	23	20	21	22	23	24	25	26
Sun Mon Tue Wed Thu Fri Sat Sun Mon Tue Sun Mon Sun Sun Sun Mon Sun	25	26	27	28	29	30	31	24	25	26	27	28	29	30	27	28	29	30			
Sun Mon Tue Wed Thu Fri Sat Sun Mon Tue Wed Thu Fri Sat Sun Mon Tue Wed Thu Fri Sat Sun Mon Tue Wed Thu Fri Sat Sun Mon Tue Wed Thu Fri Sat Sun Mon Tue Wed Thu Fri Sat Sun Mon Tue Wed Thu Fri Sat Sun Mon Tue Wed Thu Fri Sat Sun Mon Tue Wed Thu Fri Sat Sun Mon Tue Wed Thu Fri Sat Sun Mon Tue Wed Thu Fri Sat Sun Mon Tue Wed Thu Fri Sat Sun Mon Tue Wed Thu Fri Sat Sun Mon Tue Wed Thu Fri Sat Sun Mon Tue Wed Thu Fri Sat Sun Tue								31													
1 2 3 4 5 6 7 8 9 10 11 12 13 14 7 8 9 10 11 12 13 4 5 6 7 8 9 15 16 17 18 19 20 21 14 15 16 17 18 19 20 11 12 13 14 15 16 22 23 24 25 26 27 28 21 22 23 24 25 26 27 18 19 20 21 22 23 28 29 30																					-
8 9 10 11 12 13 14 7 8 9 10 11 12 13 14 7 18 9 10 11 12 13 4 5 6 7 8 9 15 16 17 18 19 20 21 14 15 16 17 18 19 20 11 12 13 14 15 16 16 22 23 24 25 26 27 28 21 22 23 24 25 26 27 18 19 20 21 22 23 28 29 30								Sun							Sun	Mon	Tue	Wed			Sat
15 16 17 18 19 20 21 14 15 16 17 18 19 20 21 22 23 24 25 26 27 18 19 20 21 22 23 24 25 26 27 28 29 30																					3
22 23 24 25 26 27 28 29 30																					10
MAR																					17
Sun Mon Tue Wed Thu Fri Sat Sun Sun Mon Tue Wed Thu Fri Sat Sun Mon Tue Wed Thu Fri Sat Sun Sun Mon Tue Wed Thu Fri Sat Sun Sun Mon Tue Wed Thu Fri Sat Sun	22	23	24	25	26	27	28				24	25	26	27							24
Sun Mon Tue Wed Thu Fri Sat Sun Mon Tue Wed Thu Fri Sat Sun Mon Tue Wed Thu Fri Sat Sun Mon Tue Wed Thu Fri Fri Sat Sun Mon Tue Wed Thu Fri Sat Sun Mon Tue Wed Thu Fri Sat Sun Mon Tue Wed Thu Fri Sat Sun Mon Tue Wed Tue <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td>28</td> <td>29</td> <td>30</td> <td></td> <td></td> <td></td> <td></td> <td>25</td> <td>26</td> <td>27</td> <td>28</td> <td>29</td> <td>30</td> <td>31</td>								28	29	30					25	26	27	28	29	30	31
Sun Mon Tue Wed Thu Fri Sat Sun Mon Tue Wed Thu Fri Sat Sun Mon Tue Wed Thu Fri Sat Sun Mon Tue Wed Thu Fri Fri Sat Sun Mon Tue Wed Thu Fri Sat Sun Mon Tue Wed Thu Fri Sat Sun Mon Tue Wed Thu Fri Sat Sun Mon Tue Wed Tue <td></td>																					
1 2 3 4 5 6 7 8 9 10 11 12 13 14 5 6 7 8 9 10 11 8 9 10 11 12 13 14 15 16 17 18 19 20 21 12 13 14 15 16 17 18 15 16 17 18 19 20 21 22 23 24 25 26 27 28 19 20 21 22 23 24 25 26 27 28 29 30 31 APR Sun Mon Tue Wed Thu Fri Sat 1 2 3 4 5 6 7 8 9 10 11 2 3 4 5 6 7 8 6 7 8 9 10 11 12 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 12 2 23 24 25 26 27 28 29 30 31 APR Sun Mon Tue Wed Thu Fri Sat 1 2 3 4 5 6 7 8 9 10 11 2 3 4 5 6 7 8 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 Sun Mon Tue Wed Thu Fri Sat 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 20 21 22 23 24 25 26 27 28 29 30 31 Sun Mon Tue Wed Thu Fri Sat 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 20 21 22 23 24 25 26 27 28 29 30 31 Sun Mon Tue Wed Thu Fri Sat 3 4 5 6 7 8 9 10 11 12 13 14 15 13 14 15 16 17 18 19 20 21 22 20 21 22 23 24 25 26 27 28 29 30 31 Sun Mon Tue Wed Thu Fri Sat 4 5 6 7 8 9 10 11 12 13 14 15 13 14 15 16 17 18 19 20 21 22 20 21 22 23 24 25 26 27 28 29 30 31 Sun Mon Tue Wed Thu Fri Sat 2 3 4 5 6 7 8 9 10 11 12 13 14 15 13 14 15 16 17 18 19 20 21 22 20 21 22 23 24 25 26 27 28 29 30 31 Sun Mon Tue Wed Thu Fri Sat 3 4 5 6 7 8 9 10 11 12 13 14 15 13 14 15 16 17 18 19 20 21 22 20 21 22 23 24 25 26 27 28 29 30 31 Sun Mon Tue Wed Thu Fri Sat 3 4 5 6 7 8 9 10 11 12 13 14 15 13 14 15 16 17 18 19 20 21 22 20 21 22 23 24 25 26 27 28 29 30 31 Sun Mon Tue Wed Thu Fri Sat 3 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 20 21 22 23 24 25 26 27 28 29 30 31 Sun Mon Tue Wed Thu Fri Sat 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 20 21 22 23 24 25 26 27 28 29 30 31 Sun Mon Tue Wed Thu Fri Sat 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 10 11 12 13 14 15 16 17 18 10 11 12 13 14 15 16 17 18 10 11 12 13 14 15 16 17 18 10 11 12 13 14 15 16 17 18 10 11 12 13 14 15 16 17 18 10 11 12 13 14 15 16 17 18 10 11 12 13 14 15 16 17 18 10 11 12 13 14 15 16 17 18 10 11 12 13 14 15 16 17 18 10 11 12 13 14 15 16 17 18 10 11 12 13 14 15 16 17 18 10 11 12 13 14 15 16 17 18 10 1			_											-			-		-		
8 9 10 11 12 13 14 5 6 7 8 9 10 11 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 19 20 21 22 23 24 25 26 27 28 29 30 31 Separate Sep								Sun	Mon	Tue											Sat
15 16 17 18 19 20 21 12 13 14 15 16 17 18 15 16 17 18 19 20 20 22 23 24 25 26 27 28 19 20 21 22 23 24 25 26 27 28 29 30 31 APR AUG Sun Mon Tue Wed Thu Fri Sat 1 2 3 4 5 6 7 8 9 10 11 2 3 4 5 6 7 8 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 11 12 13 14 15 16 17 18 19 20 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31								-		-											1
22 23 24 25 26 27 28 19 20 21 22 23 24 25 26 27 28 29 30 31																					14
29 30 31																					21 28
APR AUG AUG AUG Tue Wed Thu Fri Sat Sun Mon Tue Wed Thu Fri 1 2 3 4 5 6 7 8 9 10 11 2 3 4 5 6 7 8 6 7 8 9 10 11 12 13 14 15 16 17 18 9 10 11 12 13 14 15 13 14 15 16 17 18 19 20 21 22 23 24 25 16 17 18 19 20 21 22 20 21 22 23 24 25 26 27 28 29 30 23 24 25 26 27 28 29 30 31				25	20	21	28							25			24	25	20	21	28
Sun Mon Tue Wed Thu Fri Sat Sun Mon Tue Wed Thu Fri 1 2 3 4 5 6 7 8 6 7 8 9 10 11 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1<	29	30	21					20	21	20	29	30	21		29	30					
Sun Mon Tue Wed Thu Fri Sat Sun Mon Tue Wed Thu Fri 12 13 14 15 16 17 18 9 10 11 12 13 14 15 16 17 18 19 20 21				APP							ALIC							DEC			
1 2 3 4 5 6 7 8 9 10 11 2 3 4 5 6 7 8 6 7 8 9 10 11 12 13 14 15 16 17 18 9 10 11 12 13 14 15 13 14 15 16 17 18 19 20 21 22 23 24 25 16 17 18 19 20 21 22 20 21 22 23 24 25 26 27 28 29 30 23 24 25 26 27 28 29 27 28 29 30 31	Sun	Mon	Tue		Thu	Eci	Sat	Sun	Mon	Tue		Thu	Fri	Sat	Sun	Mon	Tue		Thu	Eci	Sat
5 6 7 8 9 10 11 2 3 4 5 6 7 8 6 7 8 9 10 11 12 13 14 15 16 17 18 9 10 11 12 13 14 15 13 14 15 16 17 18 19 20 21 22 23 24 25 16 17 18 19 20 21 22 20 21 22 23 24 25 26 27 28 29 30 31 23 24 25 26 27 28 29 27 28 29 30 31	Sull	HOH	rue					3011	HOH	Tue	wed	THU	111		3011	HOH					5
12 13 14 15 16 17 18 9 10 11 12 13 14 15 13 14 15 16 17 18 19 20 21 22 23 24 25 16 17 18 19 20 21 22 20 21 22 23 24 25 26 27 28 29 30 31	5	6	7					2	3	4	5	6	7		6	7					12
19 20 21 22 23 24 25 16 17 18 19 20 21 22 20 21 22 23 24 25 26 27 28 29 30 31 23 24 25 26 27 28 29 27 28 29 30 31																					19
26 27 28 29 30 23 24 25 26 27 28 29 27 28 29 30 31																					26
						-															
30 31					- 50				31												

2.4 SKELETON PROGRAM AND TEST DATA

The skeleton program is provided here: calendar.c

Test input: input files

Test output: output files

2.5 ESTIMATED DEVELOPMENT TIME

The time here is an estimate of how much time we expect you to spend on this exercise. If you need to spend way more time than this, it is an indication that some help might be needed.

- Devising and writing the algorithm (pseudo-code): 80 minutes
- Translating pseudo-code into code: 50 minutes
- Testing and debugging: 50 minutes
- Total: 3 hours

3 DEADLINE

The deadline for submitting all programs is 9 Mar, 11:59pm, 2015. Late submission will NOT be accepted.