

Brushing up Your PDS Skills

Figure 1 shows the calendar for the year 2020. A one-page printed calendar commonly represents the months in a 4×3 array. Moreover, each month is represented by a 5×7 array of dates (or blank spaces). Sometimes, the first row is used to store the last one or two dates of a month (see May and August).

Figure 1: Calendar of 2020

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

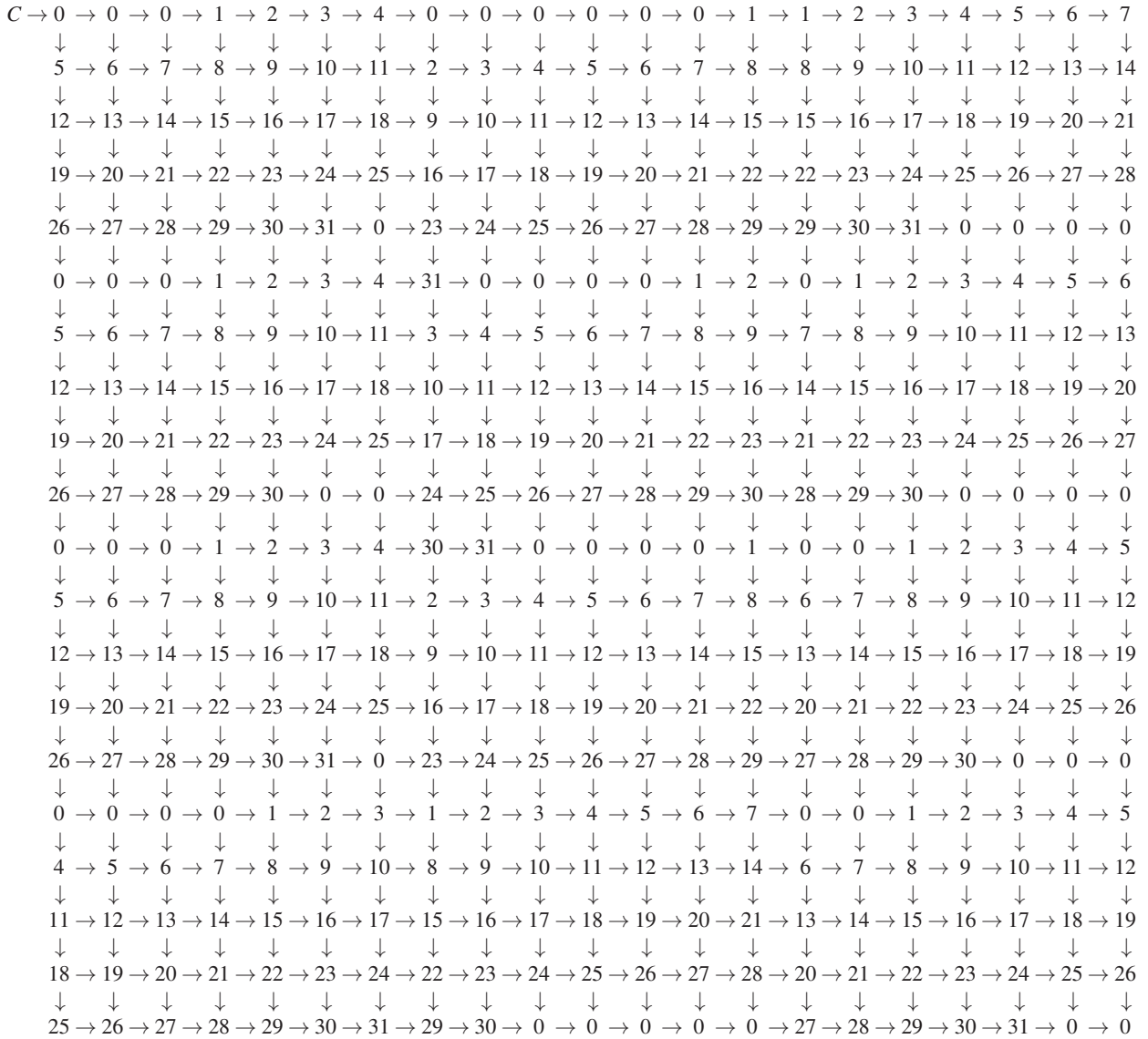
If we substitute each blank space by a 0, the entire calendar is specified by a 20×21 array of integers. Let us make a two-dimensional linked-list representation of this array. See Figure 2 for the organization of the integers as a 20×21 mesh. The entire structure is accessed by a single pointer C pointing to the cell at the top left corner of the mesh.

Part 1: Define a data type to store a node in the mesh. Each node should store an integer date, and two pointers: horizontal and vertical. Also, define a pointer to a node of this type to point to the mesh.

Part 2: Write a function *initcal* to create a 20×21 mesh using dynamic memory allocation to the cells of the mesh. The dates in all the cells are initialized to 0. A pointer C to the cell at the top left corner is to be returned by the function. In all future references to the mesh, you pass only this pointer C to access the entire mesh.

Part 3: Let us number the months as $0, 1, 2, \dots, 11$ (starting from January), and the days of a week as $0, 1, 2, \dots, 6$ (starting from Sunday). Write a function *storemonth* that, given the pointer C and a month (an integer in the range $[0, 11]$), updates the appropriate cells of the mesh by the dates in that month. You also need to specify to the function the first day in that month (an integer in the range $[0, 6]$).

Figure 2: Two-dimensional linked-list representation of the calendar



Part 4: Write a function *neatprint* that, given *C* as input (assuming that all months are stored in it), prints the calendar of the year in the format specified in Figure 1.

The *main()* function

- Read from the user (or randomly generate) the first day of the year (that is, the day of the week on first January of that year), and the information whether that year is a leap year or not.
- Call *initcal* to allocate memory to the mesh and initialize all dates to 0.
- Call *storemonth* for each of the twelve months of the year.
- Call *neatprint* to print the calendar of the year in the specified format.

Submit a single C/C++ source file. Do not use global/static variables. Do not use STL calls.