

## 2) (5 pts) ALG (Sorting)

Show the result after each iteration of performing Bubble Sort on the array shown below. For convenience, the result after the first and last iterations are provided. The first row of the table contains the original values of the array.

Iteration	Index 0	Index 1	Index 2	Index 3	Index 4	Index 5	Index 6	Index 7
0	12	2	8	19	13	7	1	16
1	2	8	12	13	7	1	16	19
2	2	8	12	7	1	13	16	19
3	2	8	7	1	12	13	16	19
4	2	7	1	8	12	13	16	19
5	2	1	7	8	12	13	16	19
6	1	2	7	8	12	13	16	19
7	1	2	7	8	12	13	16	19

**Grading: 1 per row, must get the whole row to get the point.**

## 3) (10 pts) DSN (Bitwise Operators)

There are 20 light switches, numbered 0 to 19, each which control a single light. Initially, all of the lights the switches control are off. There are several buttons. Each button toggles several switches, when pressed. For example, if a button toggles the switches 3, 5 and 9, then pressing the button changes the state of the three switches 3, 5 and 9, leaving the other switches in the same state. (So, if lights 3 and 5 were on and light 9 was off, after the button is pressed, lights 3 and 5 would be off and light 9 would be on.) Each button can be stored in a single integer, where the  $k^{\text{th}}$  bit is set to 1 if that button toggles the  $k^{\text{th}}$  switch, and set to 0 if pressing the button doesn't affect the  $k^{\text{th}}$  switch. For example, the button described would be stored as the integer 552 since  $2^3 + 2^5 + 2^9 = 552$ . Write a function that takes in an array, **buttons**, storing the buttons to press and an integer **len**, representing the length of the array **buttons** and returns a single integer storing the state of the **lights** after each of the buttons has been pressed once, assuming that all of the lights were off before any of the button presses. The format for storing the state of the lights should be identical to the format of the buttons.

```
int pressButtons(int buttons[], int len) {  
  
    int res = 0;                                // 1 pt  
  
    for (int i=0; i<len; i++)                    // 2 pts  
        res ^= buttons[i];                      // 6 pts (partial possible)  
  
    return res;                                  // 1 pt  
}
```

**Grading Partial: 1 pt update res, 3 pts XOR, 2 pts access buttons[i]**  
**If they use | instead of ^ (-2), & instead of ^ (-3), + instead of ^ (-4).**