Name:

1. (1 pt.) Unless otherwise noted, please use the definitions and notational conventions described in class for all questions. You may refer to books, papers, and notes during the exam, but material may not be shared. Ask for clarifications if needed.

   Write your name in the space provided near the top of this page.

2. (9 pts.) Answer the following briefly based on the GPU sorting paper.[1]

   (a) How many clock cycles are required for an access by a scalar processor to its shared memory? What is the size of shared memory? How many processors share it?

   (b) Depict the byte-wise **linear** memory layout of the following matrix using **column-major** format, assuming all matrix elements are one-byte integers.
   $$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

   (c) Provide a precise mathematical definition of the _scan_ operator and illustrate it by computing the scan of the array `a` of Question 4.

---

[1]Nadathur Satish, Mark Harris and Michael Garland, "Designing Efficient Sorting Algorithms for Many-core GPUs," in _Proceedings of the 23rd IEEE International Parallel and Distributed Processing Symposium (IPDPS)_ (Rome, Italy, 2009).

3. (5 pts.) List (names and conventional symbols) **three** operators that are associative and **three** that are not associative. Justify your answers briefly, and try to pick simple operators.

4. (5 pts.) Depict the **bitwise** memory layout of the following array `a`, assuming each element is an unsigned nybble (4-bit quantity).

```
   i:    0  1  2  3   4  5  6  7  8  9
 a[i]:  14  1  7  9  11  4  6  5  8  2
```

5. (10 pts.) Trace the operation of *bit-splitting radix sort* on the array `a` of Question 4, depicting the state of the array after each bitwise pass of the algorithm.

6. (15 pts.) Provide Lex[2] code for a lexer that processes standard input as follows:

- The '.' (period) character has a special meaning, denoting the beginning of a *command word*, if and only if the character immediately before it is a whitespace character (space, tab, or newline) and the character immediately after it is one of the 26 letters of the alphabet in upper case. Thus, ".ZAP" and ".A" qualify for special meaning but "12.33" and "Done." do not. The command words are terminated by a whitespace character.

- Each command word .P in the input is replaced with the string "<p>" in the output.

- Every command word other than .P is replaced in the output with the string "<span class="s">" where s is the command word excluding the initial period. For example, ".PRICE" in the input is replaced by "<span class="PRICE">" in the output.

- The rest of the input is passed to the output unmodified.

---

[2]Michael E. Lesk and Eric Schmidt, "Lex—A Lexical Analyzer Generator," in Andrew G. Hulme and M. Douglas McIlroy (eds.), *UNIX Vol. II: research system*, 10th edition (Philadelphia, Pennsylvania: W. B. Saunders Company, 1990).

7. (15 pts.)

    (a) Provide a **detailed** code fragment in C, Java, or a similar language for generating all strings of $2n$ balanced parentheses in lexicographic order, treating '(' as 1 and ')' as 0. (Please check with me for languages other than C and Java.) In more detail, your code fragment should read an integer n from standard input and write the strings to standard output in lexicographic order, one per line.

    (b) Quantify the running time and working memory (not output) used by your code as accurately as possible.

    (c) Explain your code (and algorithm) and illustrate it by tracing its operation for $n = 4$.

[additional space for answering the earlier question]

[additional space for answering the earlier question]