Pseudocode, Programming Languages, & Compilation Assignments

This assignment may be done in teams of two, in which chase the team submits a single copy with both name.

Programming Paradigm Assignment

1) For each of the following programming paradigms:
   - Imperative
   - Functional
   - Object Oriented
   - Logic

   a) Select an example language and an application that language has been used for. Provide a reference.
   b) Briefly discuss why that language is appropriate for the application you described.

Parallel Matrix Multiply Assignment

Consider matrix multiply (A=B*C), where A, B, and C are nxn arrays, as follows:

A(1,1)	A(1,2)	A(1,3)	A(1,4)
A(2,1)	A(2,2)	A(2,3)	A(2,4)
A(3,1)	A(3,2)	A(3,3)	A(3,4)
A(4,1)	A(4,2)	A(4,3)	A(4,4)

=

B(1,1)	B(1,2)	B(1,3)	B(1,4)
B(2,1)	B(2,2)	B(2,3)	B(2,4)
B(3,1)	B(3,2)	B(3,3)	B(3,4)
B(4,1)	B(4,2)	B(4,3)	B(4,4)

*

C(1,1)	C(1,2)	C(1,3)	C(1,4)
C(2,1)	C(2,2)	C(2,3)	C(2,4)
C(3,1)	C(3,2)	C(3,3)	C(3,4)
C(4,1)	C(4,2)	C(4,3)	C(4,4)

where A(1,1) is the product of row one of B and column one of C and A(1,3) is the product of row one of B and column three of C.

1) Write pseudocode for matrix multiply. Submit your pseudocode. Your pseudocode must show what iteration is needed and the computation for an element of A based on the iteration over elements.
2) Select one of the languages that you reviewed and translate your pseudocode into that language. Submit your source code. The instructions that follow are based on using C and the GCC compiler, but you can do this in any language where your compiler will generate assembly code output.
3) Compile your code and output the assembly code. Highlight the part of the code that computes the value of B(i,k) * C(k,j) including the indexing into B and C. Using gcc, the command to output the assembly code is as follows.

   ```
   gcc -c -g -Wa,-ahl=MatMul.s ./MatMul.c
   ```

   Assuming you name your C source file MatMul.c and want your assembly in MatMul.s, this will generate your assembly code with the source code embedded to ease the next two steps.
4) Highlight the assembly code in your listing that performs the multiplication (this is a single assembly statement).
5) How many lines of code are generated for the source statement that includes that multiplication?