(a) What is algebraic expression? How does it differs from mathematical formula?
Algebraic expressions are finite combinations of symbols (numbers, variables, operations) that can be solved. For example 2+3, 2x-5. It differs from the mathematical formula in such a way that it does not include an equal sign. The mathematical formula is instructions for creating desired rules. For example, c^2 = a^2 + b^2 is for finding the length of the side of a right triangle. It includes = sign and shows rule. But a^2 + b^2 itself would be an algebraic expression.

(b) What is term rewriting? Is it the same as symbolic computation?
Rewriting is replacing subterms of formula with other terms. For example, double negation can be rewritten by eliminating negation: ¬¬A -> A

(c) What is symbolic computation comparing with numerical computation?
Symbolic computation is computation with symbolic expressions (mostly rewriting). Numeric computation is an approximation for numerically solving mathematical problems. For example, an answer for 3/9 would be ⅓ in symbolic computation, but the value answer for that would be 0.33 in numerical computation.

(d) What is the difference between evaluation and interpretation (in math sense)?
Evaluation is extracting value from expression. The evaluation machine (calculator) will give a single value 4 for the expression 2+2. Interpretation is giving (meaning) to mathematical expression. In the previous example, “2” means the numerical value 2 and 2+2 gives 4. So giving the value 2 to the symbol “2” is interpretation. We also did interpretation (to the diamonds, +, etc..) in our MSS course.

(e) What is lazy evaluation comparing with eager evaluation?
Lazy evaluation is delaying the evaluation of an expression until its value is needed. Once the evaluation is needed after only that it will be done. Whereas, in eager evaluation, the expression is evaluated as soon as it is bound to a variable. Example (in swift):
Lazy: `lazy var importer = DataImporter()`
Eager: `var size = Size()`

(f) How functional programming related to algebra systems?
The core reason for saying “functional programming (FP) as algebra” is language’s main features like pure functions and immutable values. That is writing functional program is writing algebraic equations as we discussed in previous questions. Creating pure functions is like combining a series of algebraic equations together.

From algebra to FP:

`f(x) = x + 1`
`f(x,y) = x + y`
	`def f(x: Int) = x + 1`
	`def f(x: Int, y: Int) = x + y`

From FP to algebra:

`val emailDoc = getEmailFromServer(src)      // val b = f(a)`
`val emailAddr = getAddr(emailDoc)           // val c = g(b)`
`val domainName = getDomainName(emailAddr)   // val d = h(c)`
`val domainName = getDomainName(getAddr(getEmailFromServer(src)))`
	`val b = f(a)`
	`val c = g(b)`
	`val d = h(c)`
	`val d = h(g(f(a)))`