HAL/S PROGRAMMING FUNDAMENTALS
QUIZ 4

1. True or False. The DO UNTIL *(condition)* … END loop will execute at least once.

Then, if the *(condition)* is false the loop will be terminated. Ans. _____

2. True or False. In a DO FOR *(loop var)* = *(initial)* TO *(final)* BY *(increment)* … END loop, the *(final)* expression is calculated at the beginning of each cycle of the loop.
Ans. _____

3. True or False. In a **discrete** DO FOR *(loop var)* = *(exp1)*,*(exp2)*,…,*(expn)* … END loop, the individual expressions *(expi)* are evaluated just prior to their actual use.
Ans. _____

4. True or False. In both iterative DO FOR and discrete DO FOR loops the *(loop var)* can be either an unarrayed integer or scalar. Ans. _____

5. In the following case, what is the value of I on the second cycle? Ans. _____

```
J = 1;
DO FOR I = J, J+2, J+4, J-6;
        …
        J = J(J+1);
        …
END;
```

6. True or False. A WHILE or UNTIL clause (**but not both**) may be added to an iterative or discrete DO FOR loop. Ans. _____

7. True or False. In a DO CASE *(exp)* statement, the *(exp)* must be an integer or scalar expression of either single or double precision. Ans. _____

8. True or False. In a DO CASE *(exp)* statement, the *(exp)* is converted to an integer by truncation. Ans. _____

9. True or False. Error checking (i.e., for legal case numbers) is only performed for a DO CASE statement if an ELSE clause is present. Ans. _____

10. True or False. Any of the cases in a DO CASE statement can be a NULL statement (a semicolon), an assignment statement (possibly with a function invocation), a Procedure CALL, or any DO … END group (DO, DO WHILE, DO UNTIL, DO FOR), or even another DO CASE statement. Ans. _____

11. True or False. An IF… THEN statement by itself cannot form a single CASE within a DO CASE statement. Ans. _____

12. True or False. The following statement has correct semicolon placement.

DO CASE I(J+2); ELSE; Ans. _____

13. True or False. The REPEAT statement causes an immediate branch out of the innermost statement group in which it is enclosed. The REPEAT *(label)* statement causes an immediate exit from the statement group labeled *(label)*. Ans. _____

14. True or False. Use of an IF *(exp)* THEN EXIT statement within an iterative DO loop provides essentially the equivalent of adding a WHILE or UNTIL clause to the DO statement. Ans. _____

15. True or False. The REPEAT statement is only legal if it is enclosed within at least one iterative DO loop (iterative DO FOR, discrete DO FOR, DO WHILE, DO UNTIL). Ans. _____

16. True or False. A HAL/S PROCEDURE divides parameters into two classes: input parameters and assign parameters. Any procedure may have input parameters, assign parameters, or no parameters at all. Ans. _____

17. True or False. A HAL/S FUNCTION can have ASSIGN parameters but they cannot be arrayed variables. Ans. _____

18. True or False. All HAL/S PROCEDURES and FUNCTIONS must end with an END statement. If the END statement has a label it must match the label on the procedure or function. Ans. _____

19. True or False. All HAL/S FUNCTIONS must have a RETURN statement which returns an expression which matches the *(attributes)* of the FUNCTION or can be implicitly converted to a matching value. Ans. _____

20. True or False. A RETURN statement is optional in a HAL/S PROCEDURE. If used, of course, it cannot return an *(expression)*. Ans. _____

21. True or False. A CLOSE statement is considered to be an executable statement (except in COMPOOL blocks). If it has a label, e.g., CLOSE *(label)*, the label must match the label on the PROGRAM, COMPOOL, PROCEDURE, or FUNCTION.
Ans. _____

22. True or False. We use the term "parameter" when discussing the formal variables appearing in the input or assign clauses of a PROCEDURE definition. We use the term "argument" when discussing the actual variables or expressions appearing in the CALL statement (or function invocation). Ans. _____

23. True or False. ASSIGN variables are passed by NAME (i.e., address pointer) – and never by value. Ans. _____

24. True or False. INPUT variables/expressions are passed by NAME (i.e., address pointer) unless they are unarrayed INTEGERs or SCALARs. In the latter case they are passed by value. Ans. _____

25. True or False. In the definition of a PROCEDURE block, the same formal parameter name cannot appear in both the input and assign clauses. Ans. _____

26. True or False. In the CALL to a PROCEDURE, the same variable can appear in both the input and assign clauses. Ans. _____

27. True or False. Within a PROCEDURE input parameters may be referenced only (NOT assigned), whereas assign parameters may be both referenced and assigned.
Ans. _____

28. True or False. HAL/S imposes more subscripting constraints on variables passed as assign arguments versus variables passed as input arguments. Ans. _____

29. True or False. Input arguments to a PROCEDURE or FUNCTION can be expressions rather than simple variables or subscripted variables. If such expressions result in an integer, scalar, or bit string value then the resulting value is simply passed by VALUE (i.e., it is loaded into an AP-101 register). If such expressions result in an array or a VECTOR or MATRIX data type (for example), then the resulting value is stored in a temporary location and a "pointer" to that location is passed to the procedure.
Ans. _____

30. True or False. Input arguments are often implicitly converted to the data type and precision of the matching input parameter. The valid conversions are the same as for assignment statements, e.g., INTEGER → SCALAR, SCALAR → INTEGER, SINGLE → DOUBLE, DOUBLE → SINGLE, INTEGER → CHARACTER, SCALAR → CHARACTER, … Ans. _____

31. True or False. NO implicit conversions are allowed for assign arguments.
Ans. _____

32. True or False. ASSIGN arguments can be expressions but only if they are unarrayed.
Ans. _____

33. True or False. ASSIGN arguments must match the corresponding ASSIGN parameter in both data type and precision. Ans. _____

34. True or False. If the ASSIGN parameter is a SCALAR DOUBLE, then a VECTOR DOUBLE argument can be supplied in the CALL statement as long as it has a component subscript that reduces it to a single element. Ans. _____

35. True or False. The only restriction on ARRAY subscripting in an ASSIGN expression is that the dimension of the subarray be known at compile time.
Ans. _____

36. True or False. ASSIGN arguments of VECTOR or MATRIX type can be component subscripted as long as the VECTOR is reduced to a SCALAR, and the MATRIX is reduced to a scalar or a row vector (i.e., contiguous) Ans. _____

37. True or False. Although the same variable cannot appear as a parameter in both the input and assign clauses of a PROCEDURE definition, it can appear as an argument in both the input and assign clauses of a CALL statement:

CALL PROC(A,B,C) ASSIGN (C,D); Ans. _____

38. True or False. In the preceding statement if C is a VECTOR then it is passed by name (pointer) in both the input and assign clause, whereas if it is a SCALAR then it is passed by value in the input clause, and by name (pointer) in the assign clause.
Ans. _____

39. True or False. A FUNCTION can be defined to return an ARRAY but only if it is 1-dimensional (2- and 3-dimensional arrays are prohibited). Ans. _____

40. True or False. In the RETURN (exp) statement of a FUNCTION, the standard implicit conversions are legal (e.g., INTEGER → SCALAR). Ans. _____