CSE324 Quiz #4 Name----------

True or False: (we only address the **early** FORTRAN covered in class)

1. FORTRAN has the **recursive calls** feature where SUBROUTINEs & FUNCTIONs can call themselves recursively, for execution time power and efficiency. F
   (recursion leads to dynamic memory de/allocation of AR's in the stack, involving OS)

2. The compiler will generate the code that will build the subroutines/functions calls' activation records (ARs) later at run time. T

3. The FORTRAN COMMON blocks facility is mainly an ad-hoc controlled ***virtual*** global block-name visibility to save/share memory space at run time. T

4. In FORTRAN, any declared name outside the main program is ***global*** in visibility. F
   Only SUB's/FUNC's names are global!

5. FORTRAN would be ***secure***, the moment it adopts the explicit name declaration. F
   NOP! Still there are other features that makes the language insecure, e.g., overworking types!

6. In a modern block-structured HLL where the contour diagram definition is used to define name visibility, assume subroutine ***A*** declares subroutine ***S*** which declares function ***F***, the names ***S&F*** are considered part of ***A***'s locally declared names and will be seen everywhere within ***A***'s contour box. F
   Only S will be visible inside A's box, never F, since it is declared inside S, and S's box is one way mirror that never lets a name declaration (F) seeps to the outside!

7. According to the contour diagram rules, an id-name ***X*** might have different meanings (declarations) in the program code (nested box's). Its meaning will depend on where we use it! T
   Yes, statically scoped block-structured HLL's allows the same name to have different declaration in different box's in the program!

8. According to the contour diagram definition, the compiler will report an "undeclared name/id ***X***" error when the compiler encounters the use of ***X*** outside the ***scope*** of its declaration. T

9. FORTRAN is a clear example of a language that strived for maximum ***efficiency*** & ***power*** (memory and execution time) while sacrificing ***security***. F
   Gone all the way for ***efficiency*** but not ***power***!

10. The activation record (AR) of the main program in any block-structured HLL must be the first AR to be generated at run time in the memory. T

11. The *caller*'s return address will be save in its own AR. T

12. The dynamic link (DL) is a pointer from the *callee*'s AR to the *caller*'s AR. T

13. The use of subroutines/functions in a program is a tradeoff between ***abstraction*** (with all associated advantages) and the dynamic ***overhead*** on the increase of the host code execution time. T

14. FORTRAN is *dynamically* type checked. F (No way! We strive for efficiency!)

15. FORTRAN uses ***stack*** to de/allocate subroutines/functions' AR's at run time. F (No way! We strive for efficiency! Stack will force dynamic de/allocation of memeory)