Question 1. In Ada it is possible for two integer types to have identical representations and yet be distinct types (which for example cannot be assigned to one another).

(a) Give an example of declarations that have this result
(b) In one or two sentences say why this is a desirable feature

Question 2. A type in Ada can be declared as a private type. In one sentence each:

(a) What is the purpose of declaring a type as private in a non-generic package?
(b) What is the purpose of declaring a generic private formal type in a generic package?

Question 3. Ada mandates the strict separation of specifications and declarations when packages are declared. Using two or three sentences at most:

(a) How would this same separation be achieved when writing in C or C++?
(b) What is the Java view of this separation?

Question 4. Write a Python function that takes a string as an argument, and determines whether the string contains an instance of any three or more vowels (aeiouAEIOU) in a sequence. Are the features of python particularly helpful in this case? (e.g. true for “HeAollooo!”, false for “Heehaw”).

Question 5. Rewrite the problem of question 4 in Scheme, assuming that the input is given in the form of a list of characters. Do not use loops or assignments!

Question 6. In ML, write a function `pal` that operates on a list of integers, and determines whether the list contains a contiguous subsequence that is a palindrome of length 2 or more. A palindrome is a list that is the same left- to-right as right-to-left For example `(1 2 5 7 8 7 2)` contains the palindrome `(7 8 7)`. What type would ML give for your function `pal` (as you have written it)? To make this a bit easier, you can assume that you have a function `reverse` that reverses a list (you don’t need to write reverse).

Question 7. Briefly describe your overall design for the fifth assignment, in particular, what communication is there between threads? This solution has a separate thread for each person, so there are some 60 threads that communicate with one another. Would your solution scale up if there were thousands of people in the room? What would be the problems if any? Keep your answer under 100 words.

Question 8. In Ada, basic task communication is by means of the rendezvous. Why is this superior to the use of semaphores for controlling task synchronization? One or two sentences is sufficient here.
**Question 9.** Consider the following grammar for a silly little language

```
PROGRAM ::= DECLLIST  STMTLIST
DECLLIST ::= DECL  |  DECL DECLLIST
STMTLIST ::= STMT  |  STMTLIST STMT
DECL ::= TYPE  IDENTIFIER ;
TYPE ::= integer  |  float
IDENTIFIER ::= LETTER  |  LETTER IDENTIFIER
LETTER ::= a  |  b  |  c  |  d  |  e  |  f  |  g  |  h  |  i  |  j  |  k  |  l  |  m
STMT ::= IFSTMT  |  ASSIGNSTMT
IFSTMT ::= if EXPR then STMT else STMTLIST
ASSIGNSTMT ::= IDENTIFIER := EXPR
EXPR ::= PRIM OP PRIM  |  ( EXPR )
PRIM ::= IDENTIFIER  |  NUMBER  |  NUMBER . NUMBER
NUMBER ::= DIGIT  |  DIGIT NUMBER
DIGIT ::= 0  |  1  |  2  |  3  |  4  |  5  |  6  |  7  |  9
OP ::= +  |  -  |  *  |  /
```

(a) Generate appropriate diagnostic messages for this illegal program. Assume you know only what the grammar tells you! In other words, an illegal program is one that does not obey the grammar rules (there might be other rules for correct programs but we are not concerned with them here!)

```
integer abc
float def;
abc := 79 * 41 + (26 * 0)
def := 82.0;
if abc = 3 then
def := 24
abc := 13
else
abd := 14
```

(b) The form of the if statement is peculiar, and results in an ambiguity (assuming that white space is not significant, e.g. no layout conventions like Python). What is the ambiguity?

(c) You can guess a lot about what the program means, but there are some rules you cannot guess. Give one good example of a reasonable question whose answer you cannot guess from the grammar.

**Question 10.** Suppose you are attempting to translate C++ into C, but you discover that your C program does not implement pointers to functions properly, so you can’t use them. How would you translate object oriented dynamic dispatching without using pointers to functions? No need to work the details of a complete example, just give a general idea of how the translation would work. Hint: you might need the whole program to do the translation.