1. Write a real-valued function DIST() that has arguments the coordinates of two points $P_1(x_1, y_1)$ and $P_2(x_2, y_2)$, (i.e., there are a total of four real arguments) and returns the distance $\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$ between the points.

```fortran
real function dist(x1,y1,x2,y2)
implicit none
real x1, y1, x2, y2
dist = sqrt((x2-x1)**2 + (y2-y1)**2)
return
end
```

(continued on back)
2. Write a real-valued function \texttt{NGRADE()} that accepts a letter grade as a single-character argument and returns the corresponding numeric value (i.e., A=4.0, B=3.0, C=2.0, D=1.0, and F=0.0). Your code should work for upper- or lower-case letters. Character comparisons are done using the same relational operators as always, i.e., test for equality with \texttt{.EQ.} Note that FORTRAN code is typically case insensitive but that doesn’t hold for character comparisons. “A” is not equal to “a”.

```fortran
real function ngrade(score)
imPLICIT none
character score
if (score .eq. 'A' .or. score .eq. 'a') then
  ngrade = 4.0
elseif (score .eq. 'B' .or. score .eq. 'b') then
  ngrade = 3.0
elseif (score .eq. 'C' .or. score .eq. 'c') then
  ngrade = 2.0
elseif (score .eq. 'D' .or. score .eq. 'd') then
  ngrade = 1.0
else
  ngrade = 0.0
endif
return
end
```

3. Write a character-valued function \texttt{LGRADE()} that returns a letter grade based on the integer argument. The assignments should be based on 90-100=A, 80-89=B, 70-79=C, 60-69=D, and other scores are an F.

```fortran
character function lgrade(score)
imPLICIT none
integer score
if (score .ge. 90) then
  lgrade = 'A'
elseif (score .ge. 80) then
  lgrade = 'B'
elseif (score .ge. 70) then
  lgrade = 'C'
elseif (score .ge. 60) then
  lgrade = 'D'
else
  lgrade = 'F'
endif
return
end
```