CPS 5401 Fall 2015 Name _____
Homework 2
Due Tuesday, Oct 6

Fortran Exercises

1. Fill in the missing parts in the Fortran90 program below that solves a quadratic equation. Remember to be careful when checking for equality of real numbers.

```
program quadratic
   implicit none
   real :: a, b, c, root1, root2
   real :: bsq, ac4, d
   print *, 'Please enter the coefficients a, b, and c as real numbers'
   read *, a, b, c
   bsq = b*b
   ac4 = 4*a*c
   if (                          ) then
        d = sqrt(bsq – ac4)
        root1 =
        root2 =
        print *, 'The real roots are ', root1, root2
   else if (                                  ) then
        root1 =
        print *, 'There is one real root which is ', root1
   else
        print *, 'There are no real roots'
  end if
end program quadratic
```

2. Suppose you have the following parameter declaration:

```
integer, parameter :: dp = kind(0.d0)
```

a. Declare variables a, b, and c to be of kind dp

b. Write the constants 1.0, 3.5, and 1.34e8 to have kind dp

c. Suppose N is an integer. Write an expression that will compute $\frac{N-1}{N}$ as kind dp. For example, if N = 3, then you should obtain the value 0.666666666666667

3. The general form of an array slice specifier is

```
[< bound1 >]:[< bound2 >][:< stride >]
```

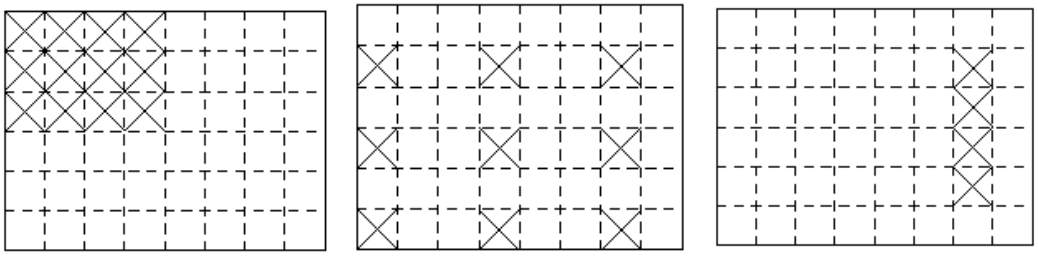
The slice starts at < *bound1* > and ends at or before < *bound2* >. < *stride* > is the increment by which the locations are selected. < *bound1* >, < *bound2* > and < *stride* > must all be scalar integer expressions. Thus

```
A(:)         ! the whole array
A(3:9)       ! A(3) to A(9) in steps of 1
A(3:9:1)     ! as above
A(m:n)       ! A(m) to A(n)
A(m:n:k)     ! A(m) to A(n) in steps of k
A(8:3:-1)    ! A(8) to A(3) in steps of -1
A(8:3)       ! A(8) to A(3) step 1 => Zero size
A(m:)        ! from A(m) to default upper bound
A(:n)        ! from default lower bound to A(n)
A(::2)       ! from default lower bound to upper bound with stride 2
A(m:m)       ! 1 element slice
A(m)         ! scalar element - not a slice
```

a. Suppose you have the following array declaration:

```
real, dimension(6, 8) :: P
```

Write slice expressions for the slices shown, where the X's represent the slice.



b. Declare a 2D integer array to represent an 8 x 8 checker board. Then write an array assignment statement to initialize the entire board to zero, which represents white. Then write two slice assignment statements to set the black squares to 1, which represents black.

4. a. Write a subroutine swap that will swap the values of its two arguments. Specify the intent of the arguments. What will happen if you pass in constants for the arguments 1) with the intent of the arguments declared, 2) with the intent not declared?

b. Write a subroutine reverse1 that will reverse the elements of its array argument in place. Declare the intent of your argument. Write your subroutine so that it can be called with array arguments of different sizes.

c. Write a function reverse2 that will return the reverse of its array argument but leave the argument unchanged.