Name: ____

Directions: MAKE SURE TO COPY YOUR ANSWERS TO A SEPARATE SHEET FOR SENDING ME AN ELECTRONIC COPY LATER.

1. (10) With the new reference classes, R has been moving somewhat away from its philosophy of having no _______________.

2. (30) The **pdist()** function in the R package of the same name computes distances of rows of a matrix A and rows of a matrix B. (For our purposes, they will be distinct matrices.) Say A has n rows and B has p rows (the two matrices must have the same number of columns). Then the major output of **pdist()** is an $n \times p$ matrix, with the (i, j) element giving the distance from row i of A to row j of B. However, these distances are stored in linear form, in row-major order, i.e. first all distances from row 1 of A are stored, then all distances from row 2 of A, etc.

Here is an example:

```
> a
     [,1] [,2] [,3] [,4]
[1,]    0    1    1    1
[2,]    1    0    0    1
> b
     [,1] [,2] [,3] [,4]
[1,]    1    1    0    1
[2,]    0    0    0    1
> str(pdist(a,b))
Formal class ’pdist’ [package ”pdist”] with 4 slots
  ..@ dist   : atomic [1:4] 1.41 1.41 1 1
  .. ..- attr(*, ”Csingle”)= logi TRUE
  ..@ n      : int 2
  ..@ p      : int 2
  ...
```

The function below takes a **pdist** object **pdout** and returns the distances in R matrix form, again the numbers in row 1 being distances from row 1 of A to rows of B. For example,

```
> pdtomat(pdist(a,b))
         [,1] [,2]
[1,] 1.414214    0
[2,] 1.732051    1
```

Fill in the blank. **NOTE:** Write this as “#1” in your quiz file, not “#1a”.

```
pdtomat <- function(pdout) {
   n <-  pdout@n
   p <-  pdout@p
   blank   (a whole line)
}
```

3. (60) The R **head()** generic function prints the first few pieces of the object it is called on. For vectors, this is the first few elements; for matrices and data frames, it is the first few rows. The default view of “few” is 6.

Here we will extend **head()** to objects of class **”ut”** in Section 12.3.2. Fill in the blanks:

```
blank (a) <- function(utmat,k=6) {
   n <- blank (b)
   k <- blank (c)
   for (row in 1:k) {
      zeros <- blank (d)
      cat(zeros,” ”)
      for (col in row:n) {
         rowcolval <- blank (e)
         cat(rowcolval,” ”)
      }
      cat(”\n”)
   }
}
```

Solutions:

1. side effects

2.

```
pdtomat <- function(pdout) {
   n <- pdout@n
   p <- pdout@p
   matrix(pdout@dist,byrow=TRUE,ncol=p)
}
```

3.

```
head.ut <- function(utmat,k=6) {
   n <- length(utmat$ix)
   k <- min(k,n)
   for (row in 1:k) {
      zeros <- rep(0,row-1)
      cat(zeros,” ”)
      for (col in row:n) {
         rowcolval <- utmat$mat[utmat$ix[col]+row-1]
         cat(rowcolval,” ”)
      }
      cat(”\n”)
   }
}
```