1. What is the return value of the following R function when invoked with 10 as its argument, e.g. my_func(10)?

```r
my_func <- function(val) {
  if (val < 10) {
    2 * val
  } else {
    3 * val
  }
}
```

Since `val >= 10`, the return value is 30. Recall that R will implicitly return the last evaluated expression.

A 10
B 20
C 30
D None
E NA

2. What is the value of result at the end of this code?

```r
val <- c(a=10, b=20, c=30)
val <- val + 1
result <- val["c"]
```

`val + 1` will add 1 to all elements in the vector, so the item with name “c” will be 31

A There will be an error
B 1
C 30
D 31
E c(11, 21, 31)

3. Which of the following expressions are equivalent to result after the code below executes?

```r
val <- list(1:10, b=0, c(1, 2))
result <- val[2]
```

Single bracket indexing on lists returns a list, and R is 1-indexed so `val[2]` returns a list with the second element or `list(b=0)`

A 3
B numbers(3)
C list(b=0)
D c(1,2)
E list(c(1,2))

4. How many rows does result have after this code is executed?

```r
df <- data.frame(X=1:10, Y=10:1, Z=rnorm(10))
result <- df[df$Y >= 9,]
```
There will be an error

The `df` data.frame has 10 rows. In the second line we index using a vector of booleans (one for each row). The vector will have 2 true values, so the resulting data.frame has 2 rows.