Desireé Smith
R Fundamental 2

```
n = 12345
vec_1= sample(12, n, replace = TRUE)
head (vec_1)
vec_1
```

```
#Q1
vec_2 = vec_1==3
vec_2
vec_1[vec_2]
```

#Q2
Two reasons why determining which elements in vec_1 have the values of 3 by visual inspection is a bad idea because when you print out vec_1 it will give you a large amount of number that you will have
to sort through. And another reason is it makes it easier for you to make mistakes when working with such a large dataset.

```
n = 12345
vec_1 = sample(12, n, replace = TRUE)
head (vec_1)
vec_1

length(vec_1)

sum(vec_1 ==3)

n = 10
vec_1 = sample(12, n, replace = TRUE)
paste0("Sum of elements with value 3: ", sum(vec_1 == 3))
```

#Q3
I did not always get the same count of 3 entries each time because it is pulling the number at random from the sample and presenting them.
Sometimes I got 1, 0 or 3 as the number of entries.

#Q4
Using a logical test is the safest way to select entries with a value of 3 is a way to help avoid human errors that can come about when looking at a lot of numbers.

#Q5
If you preform logical subsetting by hand it is possible to make mistakes but if you us a program it can help catch mistakes. Also, if you are working with collaborators, it is easier to send them computer code that is more universal and on software.

```
#Q6
for (i in 1:10)
{
  print(paste0(
    "This is loop iteration:", i))
}

#Q7
n=24
for (n in 1:n)
{
  print(paste0(
    "This is loop iteration:", n))
}

#Q8
n=17

vec_1= sample(n)
vec_1


n=17
for (n in 1:n)

{
  print(paste0(
    "The element of vec_1 at index ", n," is " , vec_1[n]))
}

#Q9


cp_vec = function (n, min = 1, max = 10 )
{
  vec_1= sample(n)
  for (i in 1:n)

  {
    print(paste0(
      "The element at index at index ", i," is " , vec_1[i]))
  }

}

cp_vec(8)
```