First create the qubit (q):

```
> q <- ket(sqrt(0.6),sqrt(0.4))
> print(dirac(q))
[1] "0.775|0> + 0.632|1>"
```

Then we need to create the EPR pair in the state 00:

```
> a <- ket(1,0)
> b <- ket(1,0)
> a <- H(a)
> ab <- tensor(a,b)
> ab <- CX(ab)
> print(dirac(ab))
[1] "0.707|00> + 0.707|11>"
```

After all of these steps, we still need to combine the EPR pair and q (newQ):

```
> newQ <- tensor(q,ab)
> print(dirac(newQ))
[1] "0.548|000> + 0.548|011> + 0.447|100> + 0.447|111>"
```

Now we need to do the first step. This step I found a little tricky because you cannot just do CX(newQ). Instead we have to use the U function in order to properly do the CNOT function over the matrix values of newQ. The correct output of this looks like:

```
> newQ <- U(CX(),I(),newQ)
> print(dirac(newQ))
[1] "0.548|000> + 0.548|011> + 0.447|101> + 0.447|110>"
```

This output makes sense because all the CNOT function does is swap around the matrix locations of the last two values.

Now we do Hadamard Gate:

```
> newQ <- U(H(),I(),I(),newQ)
> print(dirac(newQ))
[1] "0.387|000> + 0.316|001> + 0.316|010> + 0.387|011> + 0.387|100> + -0.316|101> + -0.316|110> + 0.387|111>"
```

Now we use the measure function to find our original input:

```
> measure(q, l2r=TRUE)
[[1]]
     [,1]
[1,] 0+0i
[2,] 1+0i

[[2]]
[1] 1

> measure(q, l2r=TRUE)
[[1]]
     [,1]
[1,] 1+0i
[2,] 0+0i

[[2]]
[1] 0
```

For some reason, whenever I ran the measure command I would always get these two outputs. This makes sense because both values should be 1 and this measure is probably running the measure and the output is based on the probabilities of the inputs. Since the quantum teleportation circuit has two measurement gates, it makes sense that we would have to run the measure command twice, checking for the first and then the second input.

Since we got a combination 11 from the measurement, we know that we need to do the Z gate and then the X gate:

```
> newQ <- U(Z(),I(),I(),newQ)
> newQ <- U(X(),I(),I(),newQ)
> print(dirac(newQ))
[1] "-0.387|000> + 0.316|001> + 0.316|010> + -0.387|011> + 0.387|100> + 0.316|101> + 0.316|110> + 0.387|111>"
> print(dirac(ket(newQ[7,1],newQ[8,1])))
[1] "0.632|0> + 0.775|1>"
```

First I applied the Z and X gates and then outputted the result, just to show it is correct. Then, since we got 11 for the measurement, we will want the last 2 values in newQ. Since they are not integer values, I directly got the values by doing newQ[7,1] and newQ[8,1]. Finally I put those values into a new ket qubit so we can have the values we are looking for, outputted neatly.

This output makes sense because it is the correct value for our quantum teleportation circuit.