1) The two expression operands are always evaluated when using the logical XOR operator, therefore, it cannot be short-circuited.

   unless you consider lazy evaluation as a proper way to short circuit XOR, it would look something like this…

   ```
   if (not A) return B;
   if (not B) return A;
   return 0;
   ```

   would be short-circuiting because returning B doesn't require evaluating B. (lazy eval)

2)

```
function recursiveWhile(input) {
  if (condition according to input) {
    // execute “while loop”



    // call recursiveWhile with the same/updated condition until condition is false
    recursiveWhile(input);

  } else {
    // exit recursion to main
  }
}

// start the recursion with an initial input
recursiveWhile(initialCondition);
```

3)

Q3

$\underline{XOR}\ p\ (NOT\ p)$

$\overset{N}{\Rightarrow}_{\alpha} (\lambda p.\lambda q.p(\underline{NOT}\ q)q)\ p\ (NOT\ p)$

$\overset{N}{\Rightarrow}_{\beta} (\lambda p.\lambda q.p((\underline{\lambda p}.\lambda q.\lambda r.prq)\underline{q})q)\ p\ (NOT\ p)$

$\overset{N}{\Rightarrow}_{\beta} (\underline{\lambda p}.\lambda q.p(\lambda q.\lambda r.qrq)q)\ \underline{p}\ (NOT\ p)$

$\overset{N}{\Rightarrow}_{\beta} (\lambda q.p(\lambda q.\lambda r.qrq)q)(\underline{NOT}\ p)$

$\overset{N}{\Rightarrow}_{\beta} (\lambda q.p(\lambda q.\lambda r.qrq)q)((\underline{\lambda p}.\lambda q.\lambda r.prq)\underline{p})$

$\overset{N}{\Rightarrow}_{\beta} (\underline{\lambda q}.p(\lambda q.\lambda r.qrq)q)\ \underline{(\lambda q.\lambda r.prq)}$

$\overset{N}{\Rightarrow}_{\beta} (p(\lambda q.\lambda r.(\underline{\lambda q}.\lambda r.prq)\underline{r}\ q)(\lambda q.\lambda r.prq)$

$\overset{N}{\Rightarrow}_{\beta} (p(\lambda q.\lambda r.(\underline{\lambda r}.prr)\underline{q}(\lambda q.\lambda r.prq))$

$\overset{N}{\Rightarrow}_{\beta} (p(\lambda q.\lambda r.(pqr)(\lambda q.\lambda r.prq))$

XOR p (NOT p)

$\overset{A}{\Rightarrow}_{\alpha} (\underline{\lambda p}.\lambda q.p(\text{NOT } q)\, q)\, \underline{p}(\text{NOT } p)$

$\overset{A}{\Rightarrow}_{\beta} \underline{\lambda q}.p(\text{NOT } q)\, q\, \underline{(\text{NOT } p)}$

$\overset{A}{\Rightarrow}_{\beta} p(\underline{\text{NOT}}(\text{NOT } p)(\text{Not } p)$

$\overset{A}{\Rightarrow}_{\alpha} p((\underline{\lambda p}.\lambda q.\lambda r.prq)\underline{(\text{NOT } p)})\,(\text{NOT } p)$

$\overset{A}{\Rightarrow}_{\beta} p(\lambda q.\lambda r.(\underline{\text{NOT}}\, p) r\, q)(\text{NOT } p)$

$\overset{A}{\Rightarrow}_{\alpha} p(\lambda q.\lambda r.(\underline{\lambda p}.\lambda q.\lambda r.prq)\underline{p}\, r\, q)(\text{NOT } p)$

$\overset{A}{\Rightarrow}_{\beta} p(\lambda q.\lambda r.(\underline{\lambda q}.\lambda r.prq)\underline{r}\, q)(\text{Not } p)$

$\overset{A}{\Rightarrow}_{\beta} p(\lambda q.\lambda r.(\underline{\lambda r}.pr'r')\,\underline{q})(\text{NOT } p)$

$\overset{A}{\Rightarrow}_{\beta} p(\lambda q.\lambda r.\; pqr)(\underline{\text{NOT}}\, p)$

$\overset{A}{\Rightarrow}_{\alpha} p(\lambda q.\lambda r.pqr)((\underline{\lambda p}.\lambda q.\lambda r.prq)\underline{p})$

$\overset{A}{\Rightarrow}_{\beta} p(\lambda q.\lambda r.pqr)(\lambda q.\lambda r.prq)$

(b) Test with XOR T T ⇝ should return false

XOR T T

$\Rightarrow \lambda p.\lambda q.p(\lambda p.\lambda q.\lambda r.prq)q)q)(\lambda p.\lambda q.p)(\lambda p.\lambda q.p)$

$\Rightarrow \lambda q(\lambda p.\lambda q.p)(\lambda p.\lambda q.\lambda r.prq)q)q(\lambda p.\lambda q.p)$

$\Rightarrow (\lambda p.\lambda q.p)((\lambda p.\lambda q.\lambda r.prq)(\lambda p.\lambda q.p))(\lambda p.\lambda q.p)$

$\Rightarrow (\lambda q.(\lambda p.\lambda q.\lambda r.prq)(\lambda p.\lambda q.p))(\lambda p.\lambda q.p)$

$\Rightarrow (\lambda p.\lambda q.\lambda r.prq)(\lambda p.\lambda q.p)$

$\Rightarrow \lambda q.\lambda r.(\lambda p.\lambda q.p)rq$

$\Rightarrow \lambda q.\lambda r.(\lambda q.r)q$

$\Rightarrow \lambda q.\lambda r.r$ → After reduction, it seems that XOR is behaving correctly since XOR TT has returned false.

note: definition for false is
$F \equiv \lambda p.\lambda q.q$

after reduction, the output closely matches
ie. $\lambda q.\lambda r.r$

∴ Correct.