Time: 30 mins   Total marks: 10

- **You must write your answers only in the spaces provided.**

- **The exam is open book and notes.**

- **Results/proofs covered in class/problem sessions/assignments may simply be cited, unless specifically asked for.**

- **If you need to make any assumptions, state them clearly.**

- ## Do not copy solutions from others or indulge in unfair means.

1. Consider the following program `P` with location labels `Li`:

```
L1: while (p->next != NULL) {
L2:    q := p;
L3:    p := p->next;
L4: }
```

The above program tries to traverse a linked list, the head of which is initially pointed to by `p`. `p->next` denotes the element in the list next to the element pointed to by `p`. `q` is of the same type (pointer to an element of the list) as `p`.

We wish to construct a Boolean program out of the above program, such that it captures the behaviour of the above program as accurately as possible.

Use the following Boolean variables denoting the indicated predicates to construct a Boolean program from `P`:

Boolean variable	:	Corresponding predicate
p_c	:	Is the list pointed to by `p` circular?
p_n	:	Is the list pointed to by `p` NULL?
p_q	:	Are the lists pointed to by p and q the same?

You may construct the Boolean program by filling in the blanks in the following skeleton. You must indicate your justification for each expression you use to fill in the blanks. Note that the assignment `p_c, p_n, p_q := exp1, exp2, exp3` indicates a parallel assignment of `exp1` to `p_c`, `exp2` to `p_n` and `exp3` to `p_q`.

```
L1':   while (*) {
L1'':     assert(_____);
L1''':    p_c, p_n, p_q := _____, _____, _____;
                              // L1', L1'', L1''' corr. to
                              // stmt at L1
L2':      p_c, p_n, p_q := _____, _____, _____;
                              // corr. to stmt at L2
L3':      p_c, p_n, p_q := _____, _____, _____;
                              // corr. to stmt at L3
L4':   }
```