

Before completing this discussion, read the loop invariant summary [here](#).

For more information, including video demonstrations, consult the Java HyperText entry on [loop invariants](#).

In a file called `written.txt` answer the following questions:

1. Write the four loop questions, explaining what each of them requires you to check. For example, we have described the first question for you:
 - (a) (initialization) Does it start right? If the precondition holds and then the initialization routine is run, does the loop invariant become true?
 - (b)
 - (c)
 - (d)
2. **Initialization.** Given the following preconditions and loop invariants, write initialization statements that establish the loop invariant. Your statements should only update the variables `b` and `k`, and should not update the contents of the array `c`.
 - (a) Precondition: `true`¹
Loop invariant: “`b` is `true` if and only if all elements of `c[0..k)` are 0”
 - (b) Precondition: “`true`”
Loop invariant: “`b` is `true` if and only if all elements of `c[k..c.length)` are 0”
3. **Termination.** Given the following loop invariants and postconditions, write loop guards such that if the loop guard is false and the invariant is true then the postcondition must be true.
 - (a) Loop invariant: “`b` is `true` if and only if all elements of `c[0..k)` are 0”
Postcondition: “`b` is `true` if and only if all elements of `c[0..c.length)` are 0.”
 - (b) Loop invariant: “`b` is `true` if and only if all elements of `c[k..c.length)` are 0”
Postcondition: “`b` is `true` if and only if all elements of `c[0..c.length)` are 0.”
4. **Preservation and progress.** For the given loop invariants and loop guards, give a loop body that makes progress and preserves the invariant.
 - (a) Loop invariant: “`b` is `true` if and only if all elements of `c[0..k)` are 0”
Loop guard: “`k < c.length`”
 - (b) Loop invariant: “`b` is `true` if and only if all elements of `c[k..c.length)` are 0”
Loop guard: “`0 < k`”

¹A precondition of “`true`” means that the code assumes that “`true`” is “`true`”; in other words, it doesn’t say anything at all.

5. **Linear search.** We want a loop (with initialization) that finds the largest index h with $b[h] == v$, or with $h == -1$ if v does not occur in b .

Precondition:

```

1 |      0      b.length
2 | b [  ?  ]

```

Invariant:

```

1 |      0      h      b.length
2 | b [ ? | != v ]

```

Postcondition:

```

1 | either h == -1 or
2 |
3 |      0      h      b.length
4 | b [ ? | v | != v ]

```

Give the loop, using the four loopy questions. Hint: it is very simple.

6. Below is a pair of assertions—given as array diagrams—for the Dutch National Flag algorithm. In this algorithm, array b contains red, white, and blue balls. The idea is to swap array elements so that the red ones are first, then the white ones, and then the blue ones. There are four possible invariants that have 4 segments; draw all 4 diagrams.

Precondition:

```

1 |      0      b.length
2 | b: [  ?  ]

```

Postcondition:

```

1 |      0      b.length
2 | b: [ red | white | blue ]

```