Assignment 1

Question 1

(a) Let P denote "there is a king in the hand" and Q "there is an ace in the hand".

The two given statements are formalised as:

$P \rightarrow Q$: if there is a king in the hand, then there is an ace in the hand;

$\neg P \rightarrow Q$: if there isn't a king in the hand, then there is an ace in hand.

We have the truth table:

P	Q	$P \rightarrow Q$	$\neg P \rightarrow Q$
T	T	T	T
T	**F**	**F**	**T**
F	T	T	T
F	**F**	**T**	**F**

Look up the table, one of the following statements is true and the other is false,

there are two possible cases, for $P \rightarrow Q$ is true and $\neg P \rightarrow Q$ is false,

we have P is true and Q is false, the other way P is false and Q is false.

For the statements 1. P, 2. $\neg P$, 3. Q and 4. $\neg Q$, only $\neg Q$ is bound to be true,

so the one statement that follows is *4. There is not an ace in the hand*.

(b) $[nn, ii := nn + ss(ii), ii + 1](nn = \sum xx \cdot (xx \in 1..ii - 1 \mid ss(xx)))$

= $(nn = \sum xx \cdot (xx \in 1..ii - 1 \mid ss(xx)))[nn + ss(ii), ii + 1/nn, ii]$

= $nn + ss(ii) = \sum xx \cdot (xx \in 1..ii \mid ss(xx))$

= $nn = \sum xx \cdot (xx \in 1..ii - 1 \mid ss(xx))$

Here $nn = \sum xx \cdot (xx \in 1..ii - 1 \mid ss(xx))$ acts like an invariant (within the scope),

nn seems to be keeping a record of the sum of elements in sequence ss with

index smaller than ii, the statement increased ii by one, and nn is update accordingly.

Note that the substitution of nn and ii should be done at the same time, so

in step two we have $nn + ss(ii)$ on the left hand side instead of $nn + ss(ii + 1)$.

(c) The correctness requirement for initialisation is

$[record := \{\}](record \in 0..mmax \rightarrow NAT)$

= $(record \in 0..mmax \rightarrow NAT)[\{\}/record]$

= $(\{\} \in 0..mmax \rightarrow NAT)$

= $true$

The requirement for abstract machine operation $addmark(mm)$ is

$record \in 0..mmax \rightarrow NAT \wedge mm \in NAT \Rightarrow$

$[record(mm) := record(mm) + 1](record \in 0..mmax \rightarrow NAT)$

= $record \in 0..mmax \rightarrow NAT \wedge mm \in NAT \Rightarrow$

$(record \in 0..mmax \rightarrow NAT)[record(mm) + 1/record(mm)]$

There is a problem with the substitution $[record(mm) + 1/record(mm)]$,

(or we may also write it as $[record$ <+ $\{mm \mapsto record(mm) + 1\}/record]$)

for at this point mm might not even be in $dom(record)$,

and calling $record(mm)$ might result in an error.

Another problem is with the range of mm, if it does not belong to $0..mmax$,

then $record$ <+ $\{mm \mapsto record(mm) + 1\} \notin 0..mmax \rightarrow NAT$

One possible correction is:

```
addmark(mm) =
PRE mm : 0..mmax
THEN IF mm /: dom(record)
    THEN record(mm) := 1
    ELSE record(mm) := record(mm) + 1
    END
END
```

Requirement for operation $addmark(mm)$ then becomes

$record \in 0..mmax \rightarrow NAT \wedge mm \in 0..mmax \Rightarrow$

$[IF \ldots\ END](record \in 0..mmax \rightarrow NAT)$

= $record \in 0..mmax \rightarrow NAT \wedge mm \in 0..mmax \Rightarrow ((mm \notin dom(record) \wedge$

$[record(mm) := 1](record \in 0..mmax \rightarrow NAT)) \vee (mm \in dom(record) \wedge$

$[record(mm) := record(mm) + 1](record \in 0..mmax \rightarrow NAT)))$

= $record \in 0..mmax \rightarrow NAT \wedge mm \in 0..mmax \Rightarrow ((mm \notin dom(record) \wedge$

$record$ <+ $\{mm \mapsto 1\} \in 0..mmax \rightarrow NAT) \vee (mm \in dom(record) \wedge$

$record$ <+ $\{mm \mapsto record(mm) + 1\} \in 0..mmax \rightarrow NAT))$

Here we can be sure that all necessary properties for the function

$record \in 0..mmax \rightarrow NAT$ in the invariant are preserved.

(d) Operation for deciding contiguous subsequence:

```
ii <-- con_subsequence(ss, tt) =
PRE ss : seq(NAT) & tt : seq(NAT) & card(ss) >= card(tt)
    /* sequence ss should not be shorter than tt */
THEN ii := bool(#(mm).(mm : 0..(card(ss) - card(tt)) &
        /* check if there exists a certain "offset" value mm,
           between 0 and length of ss - length of tt */
        (!(nn).(nn : 1..card(tt) => ss(nn + mm) = tt(nn)))))
        /* so that for each No. nn element in tt,
           the corresponding element in ss is No. (nn + mm) */
END
```

Operation for deciding generous subsequence:

```
ii <-- gen_subsequence(ss, tt) =
PRE ss : seq(NAT) & tt : seq(NAT) & card(ss) >= card(tt)
    /* sequence ss should not be shorter than tt */
THEN ii := bool(#(mm).(mm : seq(1..card(ss)) &
        /* check if there exists a sequence with elements from
           1..length of ss, in other words, a subset of index */
        card(mm) = card(tt) &
        /* the size of this subset is the same as length of tt */
        (!(xx, yy).(xx : 1..card(mm) & yy : 1..card(mm) &
                xx < yy => mm(xx) < mm(yy))) &
        /* make sure this "subset of index" is ordered */
        (!(nn).(nn : 1..card(tt) => ss(mm(nn)) = tt(nn)))))
        /* so that for each No. nn element in tt,
           the corresponding element in ss is No. (mm(nn)),
           in other words, the nn'th index in subset mm */
END
/* basically we are constructing a mapping from the index of tt
   to this ordered subset (sequence) of the index of ss */
```