Handout for Pseudo-code convention

The following conventions must be used to present your pseudo-code. Please note that the code should be **clear and easy to understand**. Otherwise, we take off your points.

1. Give a valid name for the pseudo-code procedure, specify the input variables’ names and types, specify the output type. (See sample code for insertion sort at the end).

2. Use the line numbers for each line of code.

3. Use proper **Indentation** for every statement in a block structure.

4. For a flow control statements use **if-else**. Always end an **if** statement with an **end-if**. Both **if**, **else** and **end-if** should be aligned vertically in same line.

   Ex:

   ```
   If (conditional expression)
       statements (see the indentation)
   else
       statements
   end-if
   ```

5. Use = or ← operator for assignment statements.

   Ex:

   ```
   i = j or i ← j
   n = 2 to length[A] or n ← 2 to length[A]
   ```

6. Array elements can be represented by specifying the array name followed by the index in square brackets. For example, A[i] indicates the *ith* element of the array A.

7. For looping or iteration use **for** or **while** statements. Always end a **for** loop with **end-for** and a **while** with **end-while**.
8. The conditional expression of **for** or **while** can be written as shown in rule (4). You can separate two or more conditions with an **and**.

Sample pseudo-code for insertion sort using the above conventions:

```plaintext
INSERTION-SORT(A)

Input: array A[0...n-1] with n items in random order

Output: array A whose items are in ascending order

1. for j <- 2 to length[A]
2.   key <- A[j]
3.   i <- j-1
4.   while i > 0 and A[i] < key  // If required, use this convention for a comment
6.     i <- i + 1
7.   end-while
8.   A[i+1] <- key
9. end-for
```