LaTeX : testing algos

1

.a

test

$A \leftarrow \varnothing$

Algorithm 1: test

Input: input
Output: output

```
1  Function func(a, b):
2      if this is true then
3          do this;
4          do that;
5      while a ≠ b do
6          a ← a + 1;
7      for k = 1 to b do
8          a ← k^2 − √a;
9          if a is not a square then
10             func(b, a);
11     return a^2 − b^2;
12 Function test(a, b):
13     if a ⩾ b then
14         Print a − b;
15     else
16         Print b − a;
```

Algorithm 2: Petersen

```
1  turn ← 0;
2  want ← 0;
3  Thread i:
4      want[i] ← true;
5      turn ← 1 − i;
6      while want[1 − i] && turn = 1 − i do
7          Wait
       // Critical section
8      want[i] ← false;
```

Algorithm 3: Phase d'attente active pour le fil i

```
1 for j = 0 to n − 1 do
2     while ticket[j] != 0 et (ticket[j] < ticket[i] ou (ticket[j] = ticket[i]
       et j < i)) do
3         Attendre;
  // Sortie de section critique :  on détruit le ticket
4 ticket[i] ← 0 ;
```

2 sec

2.1 sub

2.1.1 subsub

.a

Algorithm 5: Test

```
1 Function test():
2     return 0;
3 This is a long line or probably not or qls flsdkcldkclsk nflk sjd,lkdjf, lskdj,fl
    skd,lk:dj ,lsdkjf, lskdjnf klsjd.
```

2.2 Sub2

Algorithm 6: Test2

```
1 return 1 ;                                                    // Comment
```