Listing 4: question4

```
1  ---------- MODULE qquestion3a ----------
2  EXTENDS Naturals, Sequences, TLC
3  CONSTANT p
4
5  Remove(i, seq) == [j \in 1..(Len(seq)-1) |-> IF j < i THEN seq[j] ELSE seq[j+1]]
6
7  (*
8  --algorithm algo {
9
10 variable
11     requests = <<>>, reply = [i \in 1..p |-><<>>], msgok = <<>>;
12
13 macro Send(m, chan) {
14     chan := Append(chan, m);
15   };
16
17 macro Recv(v, chan) {
18     await chan # <<>>; \* could also do Len(chan) > 0 ??
19     v := Head(chan);
20     chan := Tail(chan);
21   };
22
23  process (C \in 1..p )
24     variable request = 0, mes, cs = 0;
25   {
26   s: while (TRUE) {
27     c1: request := 1;
28     c2: Send(self,requests);
29     c3: Recv(mes,reply[self]);
30     c4: cs := 1;
31     c5: request := 0;
32     c6: Send(self,msgok);
33     };
34 }
35
36
37   process (Server = 0 )
38     variables cs=0,v;
39   {
40   r: while (TRUE) {
41       if (requests # <<>> /\ cs=0)
42       {
43            a: Recv(v,requests);
44            b: cs := 1;
45            c: Send(v,reply[v]);
46
47       } else if (msgok # <<>>)
48       {
49            d: Recv(v,msgok);
50            e: cs := 0;
51       } else
52       { v:skip;
53       }
54       ;
55
56
57         };
58   };
59
60 } \* end algorithm
61
62 *)
63
64 ==========
```

Figure 4: Programme de gestion de la population de l'exercice 4