

Programme :

```
int position (char** T, char* X, int N) {  
    Int p=0 ;  
    While ( T[p] != X ∧ p != N) {  
        P++ ;  
    }  
    If ( p==N ) {  
        P=-1 ;  
    }  
    Return (p) ;  
}
```

SPECIFICATION :

/* (N>0) */

Cherche (N,T,X,p) :

/* $(0 \leq p < N) \wedge (\forall I : 0 \leq I < p \rightarrow T[I] \neq X) \wedge (p \neq N \rightarrow T[p] = X)$ */

Invariant :

/* $(0 \leq p < N) \wedge (\forall I : 0 \leq I < p \rightarrow T[I] \neq X)$ */

Condition de Continuation :

/* $(p \neq N \wedge T[p] \neq X)$ */

Etape n°1 :

Pfp(" p=0 ",Inv) : $(0 \leq 0 < N) \wedge (\forall I : 0 \leq I < 0 \rightarrow T[I] \neq X)$

$$\begin{array}{ccc} (N > 0) & \perp & \longrightarrow ? \\ \text{✓} & & \text{T} \quad \text{✓} \end{array}$$

Etape n°2 :

Pfp(" p=p+1 ", /* Inv */): $(0 \leq p+1 < N) \wedge (\forall I : 0 \leq I < p+1 \rightarrow T[I] \neq X)$

Inv : $0 \leq p \rightarrow 0 \leq p+1$ ✓

Inv : $p < N \rightarrow p+1 < N+1$

$p+1 < N$ ✓

CC : $p \neq N$

Inv : $\forall I : 0 \leq I < p \rightarrow T[I] \neq X$

$(\forall I : 0 \leq I < p+1 \rightarrow T[I] \neq X)$ ✓

CC : $T[p] \neq X$

Etape n°3 :

$(\text{Inv} \wedge \neg \text{CC}) \rightarrow$ /* PS sortie de boucle soit $\neg \text{CC}$ */ ✓

$(\text{Inv} \wedge \neg \text{CC} \wedge (p=N)) \rightarrow$ /* $(\text{Inv} \wedge \neg \text{CC} \wedge (p=-1 \vee T[p]=X))$ */ ✓

Etape n°4 :

On choisit pour variante $f=N-p$

Inv : $(0 \leq p < N) \quad (-p \leq 0 < N-p) \rightarrow f > 0$ ✓

Etape n°5 :

On définit $T=f$

Pfp (" $p=p+1$ ", $T > f$) : $(T > N-p-1)$

$T=f=N-p$ $T > f$ ✓

Inv : $0 \leq p < N$