

## Informatique Théorique 1. Travaux Dirigés : 2

### 1 Exercice

Montrer que les triplets suivants sont corrects :

- $\{ x = 1 \vee x = -1 \}$   
 $x := x + 1$   
 $\{ 0 \leq x^2 \leq 4 \}$
- $\{ x \in \mathbb{Z} \wedge (x < -1 \vee x > 1) \}$   
**if**  $x \leq 0$   
**then**  $x := -x$   
**else skip**  
**endif**  
 $\{ x \geq 2 \}$

### 2 Exercice

Soient  $x, y, z$  trois variables dans  $\mathbb{Z}$ , et l'instruction  $\mathcal{S}$  ci-dessous

```
if  $y < x$   
then  
   $z := x ;$   
   $x := y ;$   
   $y := z$   
else  
  skip  
endif
```

Montrer que le triplet  $\{\text{true}\} \mathcal{S} \{x \leq y\}$  est correct. (On donnera les diverses étapes de la preuve en précisant les règles appliquées).

### 3 Exercice

- Prouver le triplet suivant :

$$\{y = b \wedge x = a\} z := y ; y := x ; x := z \{x = b \wedge y = a\}$$

- En déduire que le triplet suivant est correct :

$$\{x = a \wedge y = b\} z := y ; y := x ; x := z \{a \neq b \implies x \neq y\}$$

*Comme dans l'exercice précédent et les exercices suivants, on donnera le plus de détails possible dans les démonstrations*

## 4 Exercice

On considère des variables sur  $\mathbb{N}$ . Montrer que le triplet suivant est correct :

```
{  $x^n = a y^p$  }  
if p = 0  
then  
  skip  
else  
  if p % 2 = 0  
  then  
    y := y*y ; p := p/2  
  else  
    a := a*y ; p := p-1  
  endif  
endif  
{  $x^n = a y^p$  }
```

## 5 Exercice

Considérez le triplet suivant :

```
{ $n \in \mathbb{N}_1$ }  
while 0 < n do  
  n := n+1  
done  
{2 = 3}
```

1. Ce triplet vous paraît-il correct ? Pourquoi ?
2. Si vous avez répondu “oui” à la question précédente, prouver ce triplet en utilisant les règles du calcul de Hoare.

## 6 Exercice

Prouver le triplet ci-dessous (en précisant bien quelles règles sont appliquées)

```
{ $x \in \mathbb{Z} \wedge y \in \mathbb{Z}$ }  
while  $x \neq y$  do  
  y := y - 1  
done  
{ $x = y$ }
```

## 7 Exercice

Prouver le triplet ci-dessous (en précisant bien quelles règles sont appliquées)

```
{x ∈ ℕ ∧ y ∈ ℕ}
a := 0 ;
k := y ;
while k > 0
do
  a := a + x ;
  k := k - 1
done
{a = xy}
```