Licence S.T.A. - semestre 2
Informatique - A.P.I.1
TD n°2 passage de paramètres

Exercice 1 : *Reconnaître des instructions valides et comprendre ce qu'elles font*

Q 1 . On suppose dans cette question déclarées les procédures, fonctions et variables suivantes :

```
procedure p(const x : CARDINAL; out y : BOOLEAN);
function f(const x : CARDINAL) : CARDINAL;

var
  x,y : BOOLEAN;
  z,t : CARDINAL;
```

Dans ce contexte, quelles sont les instructions valides parmi celles ci-dessous ?

1. `p(z,x);`
2. `p(1,true);`
3. `p(2*z,x);`
4. `p(z,x and y);`
5. `a := p(z,x);`
6. `f(z) := t;`
7. `z := f(t);`
8. `z := f(2*z);`
9. `if f(z) then ...`
10. `f(z);`
11. `p(f(z),x);`
12. `z := f(p(z,x));`

Q 2 . On suppose dans cette question déclarées les procédures, fonctions et variables suivantes :

```
procedure p(const x : INTEGER; var y : INTEGER);
var
  z : INTEGER;
begin
   z := x+y;
   y := z-2*y;
end {p};

var
  x,y,z : INTEGER;
```

En supposant les variables initialisées par `{x:=1;y:=2;z:=3}`, indiquez la valeur de ces variables après chacune des instructions suivantes :

1. `p(x,y);`
2. `p(y,z);`
3. `p(x+y,z);`

Exercice 2 : *Écrire des procédures*

Le type `JOUR` étant supposé défini par

```
type JOUR = (LUNDI,MARDI,MERCREDI,JEUDI,VENDREDI,SAMEDI,DIMANCHE) ;
```

Q 1 . Écrivez une procédure `LireJour` de saisie d'une valeur de type `JOUR`
(vous pouvez vous poser la question : sous quelle forme sont saisies ces valeurs ? Un nombre 1 pour lundi, 2 pour mardi, etc. ou bien une chaîne de caractères 'lundi' pour lundi, 'mardi' pour mardi, etc. et donc faire deux versions de cette procédure)

Q 2 . Écrire la procédure `inc` qui permet de modifier la valeur d'une variable `j` de type `JOUR` en la remplaçant par son successeur.
(On rappelle que cette procédure existe déjà, on vous demande donc comment elle est écrite)

Exercice 3 : *Procédure ou fonction ?*

Q 1 . Voici une fonction et une séquence d'instructions utilisant cette fonction

```
// Calcule le plus grand naturel p tel que p*p <= n
function racineEntiere(n : CARDINAL): CARDINAL ;
begin
  ...
end {racineEntiere} ;

var n,m : CARDINAL ;
```

```
begin
  ...
  n := ... ; m :=
  while racineEntiere(n) <= m do begin
     n := n+5 ;
  end {while} ;
  ...
end.
```

Écrivez une procédure `calculerRacineEntiere` à deux paramètres `n` et `p` qui fait la même chose que la fonction `racineEntiere` en stockant le résultat dans `p`.

Comment s'écrit alors la séquence d'instructions du programme principal ?

Q 2 . La division entière entre deux naturels possède deux résultats le quotient et le reste d'où l'existence en Pascal des deux opérateurs `div` et `mod` ; cela dit la séquence d'instructions

```
n := ... ; p := ... ;
writeln(n div p) ;
writeln(n mod p) ;
```

fait deux fois la même division de `n` par `p`

Écrivez une procédure `diviser` qui calcule ces deux résultats en utilisant les deux opérateurs `div` et `mod`. A-t-on résolu le problème des deux divisions ?

Écrivez une seconde version de cette procédure `diviser` qui ne fait qu'une seule fois la division

Q 3 . Pour calculer le triple d'un naturel on peut faire trois sous-programmes différents

- une fonction `triple` à un seul paramètre `n` qui renvoie le triple de `n` ;
- une procédure `tripler` à un paramètre `n` qui triple la valeur de `n`, mais dans ce cas l'ancienne valeur de `n` est perdue ;
- une procédure `tripler` à deux paramètres `n` et `p` qui stocke dans `p` le triple de la valeur de `n`, et dans ce cas la valeur de `n` n'est pas perdue ;

En supposant l'un des trois sous-programmes écrits, écrivez les deux autres en utilisant celui déjà écrit. (il y a donc trois questions)