TD N°2
Bases de l'algorithmique (suite)

1. Opérations avec les pointeurs

Exemple

Commentez le schéma d'adressage de la séquence suivante :

```
Déclaration
      i : Entier
      j : Entier
      p : Pointeur sur Entier
Début
      i ← 1
      j ← 2
      p ← &i
      *p ← 3
Fin
```

	i	j	p
Déclaration :	??	??	??
	0x01	**0x02**	**0x03**

Ligne 1 :	**1**	??	??
Ligne 2 :	1	**2**	??
Ligne 3 :	1	2	**0x01**
Ligne 4 :	**3**	2	0x01

Programme 1

Commentez le schéma d'adressage de la séquence suivante :

```
Déclaration
      i : Entier
      j : Entier
      p : Pointeur sur Entier
      q : Pointeur sur Pointeur sur Entier
Début
      i ← 1
      j ← 2
      p ← &i
      q ← &p
      *p ← 3
      **q ← 4
Fin
```

Corrections :

	i	j	p	q
Déclaration :	??	??	??	??
	0x01	**0x02**	**0x03**	**0x04**
Ligne 1 :	**1**	??	??	??
Ligne 2 :	1	**2**	??	??
Ligne 3 :	1	2	**0x01**	??
Ligne 4 :	1	2	0x01	**0x03**
Ligne 5 :	**3**	2	0x01	0x03
Ligne 6 :	**4**	2	0x01	0x03

Programme 2

Commentez le schéma d'adressage de la séquence suivante :

```
Déclaration
      A : Entier
      B : Entier
      p1 : Pointeur sur Entier
      p2 : Pointeur sur Entier
      p3 : Pointeur sur Réel
Début
      A ← 1
      B ← 4
      p2 ← &B
      p1 ← p2
      *p1 ← A
      p3 ← &B
      *p3 ← (*p3)/2
Fin
```

Corrections :

	A	B	p1	p2	p3
Déclaration :	??	??	??	??	??
	0x01	**0x02**	**0x03**	**0x04**	**0x04**
Ligne 1 :	**1**	??	??	??	??
Ligne 2 :	1	**4**	??	??	??
Ligne 3 :	1	4	??	**0x02**	??
Ligne 4 :	1	4	**0x02**	0x02	??
Ligne 5 :	1	**1**	0x02	0x02	??
Ligne 6 :	1	1	0x02	0x02	**0x02**
Ligne 7 :	1	**0**	0x02	0x02	0x02

Programme 3

Commentez le schéma d'adressage de la séquence suivante :

```
Déclaration
      v1 : Entier
      v2 : Entier
      p1 : Pointeur sur Entier
      p2 : Pointeur sur Entier
      p3 : Pointeur sur Pointeur sur Entier
Début
      p1 ← &v1
      p3 ← &p1
      p2 ← &(**p3)
      *p3 ← &v2
      *p2 ← 8
      **p3 ← 5
      p2 ← p1
      p1 ← p2
      *p1 ← 12
Fin
```

Corrections :

	v1	v2	p1	p2	p3
Déclaration :	??	??	??	??	??
	0x01	**0x02**	**0x03**	**0x04**	**0x05**
Ligne 1 :	??	??	**0x01**	??	??
Ligne 2 :	??	??	0x01	??	**0x03**
Ligne 3 :	??	??	0x01	**0x01**	0x03
Ligne 4 :	??	??	**0x02**	0x01	0x03
Ligne 5 :	**8**	??	0x02	0x01	0x03
Ligne 6 :	8	**5**	0x02	0x01	0x03
Ligne 7 :	8	5	0x02	**0x02**	0x03
Ligne 8 :	8	5	**0x02**	0x02	0x03
Ligne 9 :	8	**12**	0x02	0x02	0x03

2. Calcul du lendemain

Ecrivez un programme qui lit la date d'un jour, exprimée sous la forme de trois nombres j (jour), m (mois), a (année) et qui calcule et affiche la date du lendemain. On supposera que la date donnée est correcte.

Corrections :

```
Programme calculLendemain ()
Déclaration
      j : Entier
      m : Entier
      a : Entier
      j1 : Entier
      m1 : Entier
      a1 : Entier
      finMois : Entier

Début
      Lire(j)
      Lire(m)
      Lire(a)
      Selon(m)
            Si 2 alors
                  Si a%4 = 0 alors
                        finMois = 29
                  sinon
                        finMois = 28
                  FinSi
            Si 1 ou 3 ou 5 ou 7 ou 8 ou 10 ou 12 alors
                  finMois ← 31
            Sinon
                  finMois ← 30
            FinSi
      FinSelon
      Si j = finMois alors
            j1 ← 1
            Si m = 12 alors
                  m1 ← 1
                  a1 ← a + 1
            Sinon
                  m1 ← m + 1
                  a1 ← a

            FinSi
      Sinon
            j1 ← j + 1
            m1 ← m
            a1 ← a1

      FinSi

      Ecrire(j1)
      Ecrire(m1)
      Ecrire(a1)
Fin
```

3. Triangle de Pascal

Ecrire une fonction qui calcule la nième ligne du triangle de Pascal.

Exemple :

ligne 1: 1

ligne 2: 1 1

ligne 3: 1 2 1

ligne 4: 1 3 3 1

ligne 5: 1 4 6 4 1

Donnée : un entier n

Résultat : un tableau d'entiers de n entiers correspondant à la nième ligne du triangle de Pascal.

Cas triviaux:

`n = 1`	`ligne[1] = 1`
`n = 2`	`ligne[1] = 1`
	`ligne[2] = 1`

Cas général :

`n = 1`	`ligne[1] = 1`
`n = 2`	`ligne[1] = 1`
	`ligne[2] = 1`
`n > 2`	`ligne[1] = 1`
	`ligne[n] = 1`
	`ligne[j] = ligne_précédente[j-1] + ligne_précédente[j] pour 2 <= j < n` `où ligne_précédente est la (n-1)ième ligne du triangle de Pascal`

Solution :

```
fonction TrianglePascal(E n: Entier) : tableau[1..n] d'Entier
déclaration j : Entier
             ligne0 : tableau[1..n-1] d'Entier
             ligne : tableau[1..n] d'Entier
début
      Si n=1 alors
             ligne[1]←1
      sinon si n=2
            ligne[1]←1
            ligne[2]←1
      sinon
            ligne[1]←1
            ligne[n]←1
            ligne0←trianglePascal(n-1)
            pour j←2 à n-1 par pas de 1 faire
                  ligne[j] ← ligne0[j-1] + ligne0[j]
            fait

      finsi
      retourner ligne
fin trianglePascal
```