

## PARTIE 5

### CORRIGÉS DES EXERCICES

#### Exercice 5.1

```
Variable N en Entier
Debut
N ← 0
Ecrire "Entrez un nombre entre 1 et 3"
TantQue N < 1 ou N > 3
    Lire N
    Si N < 1 ou N > 3 Alors
        Ecrire "Saisie erronée. Recommencez"
    FinSi
FinTantQue
Fin
```

[énoncé](#) - [retour au cours](#)

#### Exercice 5.2

```
Variable N en Entier
Debut
N ← 0
Ecrire "Entrez un nombre entre 10 et 20"
TantQue N < 10 ou N > 20
    Lire N
    Si N < 10 Alors
        Ecrire "Plus grand !"
    SinonSi N > 20 Alors
        Ecrire "Plus petit !"
    FinSi
FinTantQue
Fin
```

[énoncé](#) - [retour au cours](#)

#### Exercice 5.3

On peut imaginer deux variantes, strictement équivalentes :

```
Variables N, i en Entier
Debut
Ecrire "Entrez un nombre : "
Lire N
Stop ← N+10
Ecrire "Les 10 nombres suivants sont : "
TantQue N < Stop
    N ← N+1
    Ecrire N
FinTantQue
Fin
```

Ou bien :

```
Variables N, i en Entier
Debut
Ecrire "Entrez un nombre : "
Lire N
i ← 0
Ecrire "Les 10 nombres suivants sont : "
TantQue i < 10
    i ← i + 1
    Ecrire N + i
FinTantQue
Fin
```

[énoncé](#) - [retour au cours](#)

#### Exercice 5.4

Là encore, deux variantes, correspondant trait pour trait à celles du corrigé précédent :

```
Variables N, i en Entier
Debut
Ecrire "Entrez un nombre : "
Lire N
Ecrire "Les 10 nombres suivants sont : "
Pour i ← N + 1 à N + 10
    Ecrire i
i Suivant
Fin
```

Ou bien :

```
Variables N, i en Entier
Debut
Ecrire "Entrez un nombre : "
Lire N
Ecrire "Les 10 nombres suivants sont : "
Pour i ← 1 à 10
    Ecrire N + i
i Suivant
Fin
```

[énoncé](#) - [retour au cours](#)

#### Exercice 5.5

```
Variables N, i en Entier
Debut
Ecrire "Entrez un nombre : "
Lire N
Ecrire "La table de multiplication de ce nombre est : "
Pour i ← 1 à 10
    Ecrire N, " x ", i, " = ", n*i
i Suivant
Fin
```

[énoncé](#) - [retour au cours](#)

#### Exercice 5.6

```
Variables N, i, Som en Entier
Debut
Ecrire "Entrez un nombre : "
Lire N
Som ← 0
Pour i ← 1 à N
    Som ← Som + i
i Suivant
Ecrire "La somme est : ", Som
Fin
```

[énoncé](#) - [retour au cours](#)

#### Exercice 5.7

```
Variables N, i, F en Entier
Debut
Ecrire "Entrez un nombre : "
Lire N
F ← 1
Pour i ← 2 à N
    F ← F * i
i Suivant
Ecrire "La factorielle est : ", F
Fin
```

[énoncé](#) - [retour au cours](#)

#### Exercice 5.8

```
Variables N, i, PG en Entier
Debut
PG ← 0
Pour i ← 1 à 20
    Ecrire "Entrez un nombre : "
    Lire N
    Si i = 1 ou N > PG Alors
        PG ← N
    FinSi
i Suivant
Ecrire "Le nombre le plus grand était : ", PG
Fin
```

En ligne 3, on peut mettre n'importe quoi dans PG, il suffit que cette variable soit affectée pour que le premier passage en ligne 7 ne provoque pas d'erreur.

Pour la version améliorée, cela donne :

```
Variables N, i, PG, IPG en Entier
Debut
PG ← 0
Pour i ← 1 à 20
    Ecrire "Entrez un nombre : "
    Lire N
    Si i = 1 ou N > PG Alors
        PG ← N
        IPG ← i
    FinSi
i Suivant
Ecrire "Le nombre le plus grand était : ", PG
Ecrire "Il a été saisi en position numéro ", IPG
Fin
```

[énoncé](#) - [retour au cours](#)

#### Exercice 5.9

```
Variables N, i, PG, IPG en Entier
Debut
N ← 1
i ← 0
PG ← 0
TantQue N <> 0
    Ecrire "Entrez un nombre : "
    Lire N
    i ← i + 1
    Si i = 1 ou N > PG Alors
        PG ← N
        IPG ← i
    FinSi
FinTantQue
Ecrire "Le nombre le plus grand était : ", PG
Ecrire "Il a été saisi en position numéro ", IPG
Fin
```

[énoncé](#) - [retour au cours](#)

#### Exercice 5.10

```
Variables E, somdue, M, Reste, Nb10E, Nb5E En Entier
Debut
E ← 1
somdue ← 0
TantQue E <> 0
    Ecrire "Entrez le montant : "
    Lire E
    somdue ← somdue + E
FinTantQue
Ecrire "Vous devez :", somdue, " euros"
Ecrire "Montant versé : "
Lire M
Reste ← M - somdue
Nb10E ← 0
TantQue Reste >= 10
    Nb10E ← Nb10E + 1
    Reste ← Reste - 10
FinTantQue
Nb5E ← 0
Si Reste >= 5
    Nb5E ← 1
    Reste ← Reste - 5
FinSi
Ecrire "Rendu de la monnaie : "
Ecrire "Billets de 10 E : ", Nb10E
Ecrire "Billets de 5 E : ", Nb5E
Ecrire "Pièces de 1 E : ", Reste
Fin
```

[énoncé](#) - [retour au cours](#)

#### Exercice 5.11

Spontanément, on est tenté d'écrire l'algorithme suivant :

```
Variables N, P, i, Numé, Déno1, Déno2 en Entier
Debut Ecrire "Entrez le nombre de chevaux partants : "
Lire N
Ecrire "Entrez le nombre de chevaux joués : "
Lire P
Numé ← 1
Pour i ← 2 à N
    Numé ← Numé * i
i Suivant
Déno1 ← 1
Pour i ← 2 à N-P
    Déno1 ← Déno1 * i
i Suivant
Déno2 ← 1
Pour i ← 2 à P
    Déno2 ← Déno2 * i
i Suivant
Ecrire "Dans l'ordre, une chance sur ", Numé / Déno1
Ecrire "Dans le désordre, une sur ", Numé / (Déno1 * Déno2)
Fin
```

Cette version, formellement juste, comporte tout de même deux faiblesses.

La première, et la plus grave, concerne la manière dont elle calcule le résultat final. Celui-ci est le quotient d'un nombre par un autre ; or, ces nombres auront rapidement tendance à être très grands. En calculant, comme on le fait ici, d'abord le numérateur, puis ensuite le dénominateur, on prend le risque de demander à la machine de stocker des nombres trop grands pour qu'elle soit capable de les coder (cf. le préambule). C'est d'autant plus bête que rien ne nous oblige à procéder ainsi : on n'est pas obligé de passer par la division de deux très grands nombres pour obtenir le résultat voulu.

La deuxième remarque est qu'on a programmé ici trois boucles successives. Or, en y regardant bien, on peut voir qu'après simplification de la formule, ces trois boucles comportent le même nombre de tours ! (si vous ne me croyez pas, écrivez un exemple de calcul et biffez les nombres identiques au numérateur et au dénominateur). Ce triple calcul (ces trois boucles) peut donc être ramenés(es) à un(e) seul(e). Et voilà le travail, qui est non seulement bien plus court, mais aussi plus performant :

```
Variables N, P, i, A, B en Numérique
Debut
Ecrire "Entrez le nombre de chevaux partants : "
Lire N
Ecrire "Entrez le nombre de chevaux joués : "
Lire P
A ← 1
B ← 1
Pour i ← 1 à P
    A ← A * (i + N - P)
    B ← B * i
i Suivant
Ecrire "Dans l'ordre, une chance sur ", A
Ecrire "Dans le désordre, une chance sur ", A / B
Fin
```

[énoncé](#) - [retour au cours](#)