

Exercice 5.1

```
entreeUtilisateur : ENTIER

Faire
    entreeUtilisateur = Lire Clavier
Tant que entreeUtilisateur < 1 ou entreeUtilisateur > 3

entreeUtilisateur = 0

Tant que entreeUtilisateur < 1 ou entreeUtilisateur > 3
    entreeUtilisateur = Lire Clavier
Fin tant que
```

Exercice 5.2

```
entreeUtilisateur : ENTIER

Faire

    // Lecture
    entreeUtilisateur = Lire Clavier

    // Message d'erreur
    Si entreeUtilisateur > 20
        Ecrire "Trop grand"
    Sinon si entreeUtilisateur < 10
        Ecrire "Trop petit"
    Fin si

Tant que entreeUtilisateur < 10 ou entreeUtilisateur > 20
```

Exercice 5.3

```
entreeUtilisateur : ENTIER

iteration = Lire Clavier
max = iteration + 10
```

```
Tant que iteration < max
  Ecrire iteration
  iteration = iteration + 1
Fin tant que
```

Exercice 5.4

:

```
entreeUtilisateur : ENTIER

n = Lire Clavier
max = n + 10

Pour (i = n; i < max; i = i + 1)
  Ecrire i
Fin pour

Pour (i = 0; i < 10; i = i + 1)
  Ecrire (n + i)
Fin pour
```

Exercice 5.5

```
n : ENTIER

n = Lire Clavier

Pour (i = 1; i <= 10; i = i + 1)
  Ecrire n + " x " + i + " = " + (i * n)

i = 1
Tant que i <= 10
  Ecrire n + " x " + i + " = " + (i * n)
  i = i + 1
Fin tant que
```

Exercice 5.6

```
n : ENTIER

n = Lire Clavier
somme = 0

Pour (i = 1; i <= n; i = i + 1)
    somme = somme + i
Fin pour

Ecrire somme
```

Exercice 5.7

```
n : ENTIER

n = Lire Clavier
factoriel = 1

Pour (i = 2; i <= n; i = i + 1)
    factoriel = factoriel * i
Fin pour

Ecrire factoriel
```

Exercice 5.8

```
entreeUtilisateur : ENTIER
lePlusGrand : ENTIER
indicePlusGrand : ENTIER
i : ENTIER

lePlusGrand = 0
indicePlusGrand = 0

Pour (i = 1; i <= 20; i = i + 1)
    Lire entreeUtilisateur

    Si i == 1 OU entreeUtilisateur > lePlusGrand Alors
        lePlusGrand = entreeUtilisateur
```

```
        indicePlusGrand = i
    Fin Si
Fin Pour

Ecrire lePlusGrand
Ecrire indicePlusGrand
```

Exercice 5.9

```
entreeUtilisateur : ENTIER
lePlusGrand : ENTIER
indicePlusGrand : ENTIER
entreeEstZero : BOOLEAN
estPlusGrand : BOOLEAN
i : ENTIER

i = 1
Faire
    Lire entreeUtilisateur

    entreeEstZero = entreeUtilisateur == 0
    estPlusGrand = i == 1 OU entreeUtilisateur > lePlusGrand

    Si NON(entreeEstZero) ET estPlusGrand Alors
        lePlusGrand = entreeUtilisateur
        indicePlusGrand = i
    Fin Si

    i = i + 1
Tant que entreeUtilisateur <> 0

// Dans le cas où l'utilisateur a rentré 0 immédiatement
// i vaut 2

Si i = 2
    Ecrire "Pas de plus grand"
Sinon
    Ecrire indicePlusGrand
    Ecrire lePlusGrand
Fin si
```

Exercice 5.10

```
somme : ENTIER
sommeARendre : ENTIER
entreeUtilisateur : ENTIER
difference : ENTIER
nbBillet10 : ENTIER

somme = 0

// Faire sa liste de course
Faire
    Lire entreeUtilisateur

    Si entreeUtilisateur < 0
        Ecrire "Vous devez rentrer un nombre positif (ou 0 pour terminer)"
    Sinon
        somme = somme + entreeUtilisateur
    Fin si
Tant que entreeUtilisateur <> 0

// Afficher la somme
Ecrire "Montant à payer : " + somme

// Payer
Lire entreeUtilisateur
somme = somme - entreeUtilisateur

// Ici somme est <= 0

// Rendre la monnaie, valeur >= 0
sommeARendre = somme * -1

// 10€ avec POUR
nbBillet10 = sommeARendre / 10
POUR (i = 0; i < nbBillet10; ++i)
    sommeARendre = sommeARendre - 10;
    Ecrire "10€"

// 10€ avec TANT QUE
Tant que sommeARendre >= 10
    sommeARendre = sommeARendre - 10;
    Ecrire "10€"
Fin Tant
```

```

// TANT QUE et SI
Tant que sommeARendre > 0
    Si sommeARendre >= 10
        // 10€
    Sinon si sommeARendre >= 5
        // 5€
    Sinon
        // 1€
    Fin si
Fin Tant que

```

Exercice 5.11

```

nbChevauxTotal : ENTIER
nbChavauxChoisis : ENTIER
factoN : ENTIER
factoP : ENTIER
factoPminusN : ENTIER
chancesOrdre : ENTIER
chancesDesordre : ENTIER

// n = nbChevauxTotal
// p = nbChavauxChoisis
// X = n! / (n - p)!
// Y = n! / (p! * (n - p)!)

// p!
// 1 * 2 * ... * p-1 * p

// si n > p
// 1 * 2 * ... * p-1 * p * p+1 * ... * n-1 * n

// Version optimisée
factoN = 1
factoP = 1
Pour (i = 2; i < n; i = i +1)
    factoN = factoN * i
    Si i == p
        factoP = factoN
    Sinon si i == (n - p)
        factoPminusN = factoN
    Fin si

```

```
FONCTION Factoriel(n ENTIER)
    resultat : ENTIER

    resultat = 1
    POUR (i = 2; i <= n; i = i + 1)
        resultat = resultat * i
    Fin POUR

    Renvoyer resultat
FIN FONCTION

// Version naïve
factoN = APPELER Factoriel(chevauxTotal)
factoP = APPELER Factoriel(chevauxChoisis)
factoPminusN = APPELER Factoriel(chevauxTotal - chevauxChoisis)

chancesOrdre = factoN / factoNminusP
chancesDesordre = factoN / (factoP * factoNminusP)
```