ALGORITHMIQUE

Annexes

Campus Booster ID : ***XXXX
Version 1.0

THE INTERNATIONAL INSTITUTE OF
SUPINFO
INFORMATION TECHNOLOGY

Contenu

1. Introduction et Généralités sur l'Algorithmique

1.1. Instructions conditionnelles et répétitives

1.1.1. SI..ALORS

```
ALGORITHME Calcul la valeur absolue
//BUT :calcul la val.abs. d'un entier saisi par l'util.
//ENTREE :un entier relatif saisi par l'utilisateur
//SORTIE :la valeur absolue de l'entier saisi
VAR :
    x             : ENTIER

DEBUT
    LIRE( x )     //affectation d'une valeur par l'util.
    SI( x < 0 )   //si cette valeur est négative
    ALORS         //alors cette condition est vraie
      x <-- -x    //et la variable se trouve réaffectée
    FINSI
    ECRIRE( "La val.abs. du nombre saisi vaut : ", x )
FIN
```

1.1.2. SI..ALORS..SINON

```
ALGORITHME Recherche du maximum entre deux nombres
//BUT : cherche la valeur max. parmi 2 valeurs saisies
//ENTREE : deux réels saisis par l'utilisateur
//SORTIE : le maximum des deux valeurs
VAR :
    a, b, max     : REEL

DEBUT
    LIRE( a, b)
    SI( a >= b )
    ALORS
            max <-- a
    SINON
            max <-- b
    FINSI
    ECRIRE ("Le Max entre", a, " et ", b," est:", max )
FIN
```

1.1.3. CAS..PARMI

```
ALGORITHME Affichage du nom du mois correspondant
//BUT : affiche le nom du mois en fonction du numéro
//ENTREE : un entier saisi par l'utilisateur
//SORTIE : le nom du mois correspondant au chiffre saisi
VAR :
    mois    : ENTIER

DEBUT
    LIRE( mois )
    CAS( mois )PARMI:
      CAS1:   1    ECRIRE( " Janvier" )
      CAS2:   2    ECRIRE( " Février" )
      ...
      CAS12: 12    ECRIRE( " Décembre" )
    FINCASPARMI
FIN
```

1.1.4. CAS..PARMI..PARDEFAUT

```
ALGORITHME Affichage du nom du mois correspondant
//BUT : affiche le nom du mois en fonction du numéro
//ENTREE : un entier saisi par l'utilisateur
//SORTIE : le nom du mois correspondant au chiffre saisi
VAR :
    mois    : ENTIER

DEBUT
    LIRE( mois )
    CAS( mois )PARMI:
      CAS1:   1    ECRIRE( " Janvier " )
      CAS2:   2    ECRIRE( " Février " )
      ...
      CAS12: 12    ECRIRE( " Décembre " )
      PARDEFAUT:   ECRIRE( " Erreur de saisie ! " )
    FINCASPARMI
FIN
```

1.1.5. TANTQUE

```
ALGORITHME Affichage d'un compteur à rebours
//BUT : affiche la valeur d'un compteur à rebours
//ENTREE : un entier saisi par l'utilisateur
//SORTIE : la valeur du compteur à rebours
VAR :
    nombre  : ENTIER

DEBUT
    LIRE( nombre )
    TANTQUE( nombre <> 0 )
    FAIRE
      ECRIRE( "Valeur du compteur à rebours: ", nombre )
      SAUTER_LIGNE( 1 )
    FINTANTQUE
FIN
```

1.1.6. REPETER..JUSQU'A

```
ALGORITHME Affichage d'un compteur à rebours
//BUT : affiche la valeur d'un compteur à rebours
//ENTREE : un entier saisi par l'utilisateur
//SORTIE : la valeur du compteur à rebours
VAR :
    nombre  : ENTIER

DEBUT
    LIRE( nombre )
    REPETER
      ECRIRE( "Valeur du compteur à rebours: ", nombre )
      SAUTER_LIGNE( 1 )
    JUSQU'A( nombre = 0 )
FIN
```

1.1.7. REPETER..TANTQUE

```
ALGORITHME Affichage d'un compteur à rebours
//BUT : affiche la valeur d'un compteur à rebours
//ENTREE : un entier saisi par l'utilisateur
//SORTIE : la valeur du compteur à rebours
VAR :
    nombre  : ENTIER
```

```
DEBUT
    LIRE( nombre )
    REPETER
      ECRIRE( "Valeur du compteur à rebours: ", nombre )
      SAUTER_LIGNE( 1 )
    TANTQUE( nombre <> 0 )
FIN
```

1.1.8. POUR

```
ALGORITHME Affichage d'un compteur à rebours
//BUT : affiche la valeur d'un compteur à rebours
//ENTREE : un entier saisi par l'utilisateur
//SORTIE : la valeur du compteur à rebours
VAR :
    nombre  : ENTIER

DEBUT
    LIRE( nombre )
    POUR( compteur <-- nombre :ENTIER a 0 au pas de -1 )
    FAIRE
      ECRIRE("Valeur du compteur à rebours: ", compteur)
      SAUTER_LIGNE( 1 )
    FINPOUR
FIN
```

1.2. Exercices résolus

1.2.1. SI..ALORS

```
ALGORITHME Obtention d'un examen
//BUT :détermine l'obtention d'un examen
//ENTREE :un entier saisi par l'utilisateur
//SORTIE :la valeur booléenne indiquant le résultat
VAR :
    moyenne             : ENTIER
    examen <-- FAUX     : BOOLEEN

DEBUT
    LIRE( moyenne )//affectation d'une valeur par l'util.
    SI( moyenne >= 12 )  //si moyenne est sup.ou ég.à 12
    ALORS        //alors cette condition est vraie
      examen <-- VRAI//et examen se trouve réaffectée
    FINSI
    ECRIRE( "Cet élève a réussi son examen : ", examen )
FIN
```

1.2.2. SI..ALORS..SINON

```
ALGORITHME Calcul du prix TTC d'un produit
CONST :
    tva_luxe <-- 19,6   : REEL
    tva_reduite <-- 5,5       : REEL
VAR :
    prix_ht, prix_ttc   : REEL
    produit_luxe        : BOOLEEN
DEBUT
    LIRE( prix_ht, produit_luxe)
    SI( produit_luxe )
    ALORS
            prix_ttc <-- prix_ht * tva_luxe
    SINON
            prix_ttc <-- prix_ht * tva_reduite
    FINSI
    ECRIRE ("Le prix TTC du produit est : ", prix_ttc)
FIN
```

1.2.3. CAS..PARMI

```
ALGORITHME Action relative au choix dans un menu FICHIER
VAR :
    choix_menu    : ENTIER

DEBUT
    LIRE( choix_menu )
    CAS( choix_menu )PARMI:
      CAS1:   1 CREER_FICHIER("Algo1.txt")
      CAS2:   2 ENREGISTRER_FICHIER("Algo1.txt")
      CAS3:   3 ENREGISTRER_SOUS_FICHIER("C:\Algo1.txt")
    FINCASPARMI
FIN
```

1.2.4. CAS..PARMI..PARDEFAUT

```
ALGORITHME Affichage du nombre de jours par mois
VAR :
    mois          : CHAINE
    nbre_jours    : ENTIER

DEBUT
    LIRE( mois )
    CAS( mois )PARMI:
      CAS1: "Février"   nbre_jours = 28
      CAS2: "Avril"     nbre_jours = 30
      CAS3: "Juin"            nbre_jours = 30
      CAS4: "Septembre"       nbre_jours = 30
      CAS5: "Novembre"  nbre_jours = 30
      PARDEFAUT:        nbre_jours = 31
    FINCASPARMI
    ECRIRE ("Il y a ", nbre_jours," jours en ", mois)
FIN
```

1.2.5. TANTQUE

```
ALGORITHME Affichage du nom de la touche appuyee
//affichage de la touche choisi dés lors que celle-ci
//n'est pas un 'q' qui correspond au menu "Quitter"
VAR :
    touche  : CARACTERE
```

```
DEBUT
    LIRE( touche )
    TANTQUE( touche <> 'q' ) //'q' pour Quitter
    FAIRE
      ECRIRE( "La touche est : ", touche )
      LIRE( touche )
    FINTANTQUE
FIN
```

1.2.6. REPETER..JUSQU'A

```
ALGORITHME Affichage des mots tapes ligne apres ligne
//affichage du mot tapé dés lors que celui-ci
//n'est pas le mot "fin"
VAR :
    mot     : CHAINE

DEBUT
    LIRE( mot )
    REPETER
      ECRIRE( mot )
      SAUTER_LIGNE( 1 )
      LIRE( mot )
    JUSQU'A( mot = "fin" )
FIN
```

1.2.7. REPETER..TANTQUE

```
ALGORITHME Affichage des mots tapes ligne apres ligne
//affichage du mot tapé dés lors que celui-ci
//n'est pas le mot "fin"
VAR :
    mot     : CHAINE

DEBUT
    LIRE( mot )
    REPETER
      ECRIRE( mot )
      SAUTER_LIGNE( 1 )
      LIRE( mot )
    TANTQUE( mot <> "fin" )
FIN
```

1.2.8. POUR

```
ALGORITHME Affichage d'un compteur à rebours
//BUT : affiche la valeur d'un compteur à rebours
//ENTREE : un entier saisi par l'utilisateur
//SORTIE : la valeur du compteur à rebours
VAR :
    nombre  : ENTIER

DEBUT
    LIRE( nombre )  //nombre de copie à photocopier
    POUR( compteur <-- 1 :ENTIER a nombre au pas de +1 )
    FAIRE
      IMPRIMER("C:\Rapport.txt")
      ECRIRE( "Photocopie numéro :", compteur )
    FINPOUR
FIN
```

2. Structure algorithmique avancée

2.1. RAPPEL

2.1.1. Séquences

Programme : Marathon ou test du potentiel d'un sportif (course à pieds)

Objectif du programme :

Le Test du potentiel a pour objectif d'établir une estimation de la durée d'un Marathon, pour une personne qui vient de se chronométrer sur 2000 mètres à la course à pieds.

Enoncé :

Ce calcul se fait à l'aide des trois équations suivantes :

Temps sur 10 kilomètres = Temps sur 2000 m x 5,5

Vitesse au semi-marathon = Vitesse au 10 km (en km/h) - 1km/h

Vitesse au marathon = Vitesse au 10 km (en km/h) - 2 km/h

Mais, la démarche complète nécessite une décomposition en étapes successives, comme par exemple :

-1- Temps sur 10 kilomètres = Temps sur 2000 m x 5,5
-2- Vitesse au 10 km (en km/h) = (10 x 60) / Temps sur 10 kilomètres
-3- Vitesse au semi-marathon = Vitesse au 10 km (en km/h) - 1km/h
-4- Vitesse au marathon = Vitesse au 10 km (en km/h) - 2 km/h
-5- Temps au semi-marathon = 21,1 / Vitesse au semi-marathon
-6- Temps au marathon = 42,195 / Vitesse au marathon

D'ores et déjà, à ce niveau d'analyse, il est possible d'envisager le regroupement des opérations 3 et 5, puis 4 et 6. Les étapes se trouvent être maintenant :

-1- Temps sur 10 kilomètres = Temps sur 2000 m x 5,5
-2- Vitesse au 10 km (en km/h) = (10 x 60) / Temps sur 10 kilomètres
-3- Temps au semi-marathon = 21,1 / Vitesse au 10 km (en km/h) - 1km/h
-4- Temps au marathon = 42,195 / Vitesse au 10 km (en km/h) - 2 km/h

L'objectif de ce regroupement est principalement de réduire le nombre d'étapes pour faire aboutir notre programme. A regarder de plus près, si deux opérations semblent être retirées, il en résulte surtout une réduction du nombre de variables, puisque les vitesses du semi-marathon et marathon ne sont plus stockées temporairement.

Dans un exemple comme celui-ci, cela ne semble pas avoir un intérêt primordial, en revanche, lors de la construction d'algorithmes plus complets et surtout plus complexes, ce principe de réduction aura un impact important non seulement sur la place occupée en mémoire par les données, mais aussi sur la lisibilité du programme.

```
//-------------------------------------------------------------------------------
//TITRE DU PROGRAMME:         Marathon (test du potentiel)
//-------------------------------------------------------------------------------
//-------------------------------------------------------------------------------
//                     PROGRAMME PRINCIPAL
//-------------------------------------------------------------------------------
//          Déclaration des constantes et des variables
//-------------------------------------------------------------------------------
//-------------------------------------------------------------------------------
//Déclaration des constantes
CONST:
    coefficient_10km<-- 5,5 :REEL
    constante_semi_marathon<-- 1 :ENTIER
    constante_marathon<-- 2 :ENTIER
//-------------------------------------------------------------------------------
//Déclaration des variables
VAR:
    course_10km :REEL
    semi_marathon :REEL
    marathon :REEL
    temps_decimal :REEL
    temps_minutes, temps_secondes :ENTIER
//-------------------------------------------------------------------------------
DEBUT
    ECRIRE("Donnez votre temps de course pour 2000 mètres: en minutes décimales")
    LIRE(temps_decimal)
    course_10km <-- temps_decimal * coefficient_10km
    semi_marathon <-- course_10km - constante_semi_marathon
    marathon <-- semi_marathon - constante_marathon
    ECRIRE("Pour un temps de parcours de 2000m:",temps_decimal,", votre temps de
course sur 10km est estimé à:", course_10km )
    ECRIRE("Pour un temps de parcours de 2000m:",temps_decimal,", votre temps de
course pour un semi marathon est estimé à:", semi_marathon )
    ECRIRE("Pour un temps de parcours de 2000m:",temps_decimal,", votre temps de
course pour un marathon est estimé à:", marathon )
FIN
//-------------------------------------------------------------------------------
```

2.1.2. Ruptures de sequences

Programme : Choix de la monnaie en euro

Objectif du programme :

Lors de l'achat d'un produit, dans la monnaie euro, l'ordinateur aura pour tâche d'ajuster le nombre de billets afin de régler la somme due.

Enoncé :

L'ordinateur dispose de 5 valeurs différentes de billets, soit 5 euro, 10, 20, 50 et 100 euro. En fonction du prix du produit, celui-ci déterminera le nombre de billets pour chacune des valeurs données ci-dessus.

```
//-------------------------------------------------------------------------------
//TITRE DU PROGRAMME:         Choix de la monnaie en euro
//-------------------------------------------------------------------------------
//-------------------------------------------------------------------------------
//                    PROGRAMME PRINCIPAL
//-------------------------------------------------------------------------------
//          Déclaration des constantes et des variables
//-------------------------------------------------------------------------------
//-------------------------------------------------------------------------------
//Déclaration des variables
VAR:
    prix                                        : ENTIER
    nb_billets_100 <-- 0, nb_billets_50 <-- 0 : ENTIER
    nb_billets_20 <-- 0, nb_billets_10 <-- 0  : ENTIER
    nb_billets_5 <-- 0                          : ENTIER
//-------------------------------------------------------------------------------
DEBUT
    ECRIRE ("Entrez un prix en euro (sans les centimes):")
    LIRE( prix )
    SI( (prix / 100) >= 1)
    ALORS
        nb_billets_100 <-- VALEUR_ENTIERE ( prix / 100 )
        prix <-- prix - (nb_billets_100 * 100)
    FINSI
    SI( (prix / 50) >= 1)
    ALORS
        nb_billets_50 <-- VALEUR_ENTIERE ( prix / 50 )
        prix <-- prix - (nb_billets_50 * 50)
    FINSI
    SI( (prix / 100) >= 1)
    ALORS
        nb_billets_20 <-- VALEUR_ENTIERE ( prix / 20 )
        prix <-- prix - (nb_billets_20 * 20)
    FINSI
    SI( (prix / 10) >= 1)
    ALORS
        nb_billets_10 <-- VALEUR_ENTIERE ( prix / 10 )
        prix <-- prix - (nb_billets_10 * 10)
    FINSI
    SI( (prix / 5) >= 1)
```

```
    ALORS
        nb_billets_5 <-- VALEUR_ENTIERE ( prix / 5 )
        prix <-- prix - (nb_billets_5 * 5)
    FINSI
    ECRIRE ("Billets de 100:", nb_billets_100)
    ECRIRE ("Billets de 50:", nb_billets_50)
    ECRIRE ("Billets de 20:", nb_billets_20)
    ECRIRE ("Billets de 10:", nb_billets_10)
    ECRIRE ("Billets de 5:", nb_billets_5)
    ECRIRE ("Monnaie en pièce(s):", prix)
FIN
//-------------------------------------------------------------------------------
```

2.2. Instructions conditionnelles : Si alors et Cas parmi

2.2.1. Instruction conditionnelle imbriquée : Si alors

```
//-------------------------------------------------------------------------------
//TITRE DU PROGRAMME:         Test de la filiation (booléen simple)
//-------------------------------------------------------------------------------
//-------------------------------------------------------------------------------
//                      PROGRAMME PRINCIPAL
//-------------------------------------------------------------------------------
//          Déclaration des constantes et des variables
//-------------------------------------------------------------------------------
//-------------------------------------------------------------------------------
//Déclaration des variables
VAR:
    mere, pere  :CHAINE
//-------------------------------------------------------------------------------
DEBUT
ECRIRE("Veuillez entrer la(les) lettre(s) relative(s) au groupe sanguin de la mère
[en majuscule]")
LIRE(mere)
ECRIRE("Veuillez entrer la(les) lettre(s) relative(s) au groupe sanguin du père [en
majuscule]")
LIRE(pere)
SI (mere = "O")
ALORS
{
    SI(pere = "O")
    ALORS
    {
        Afficher ("L'enfant sera de type O")
    }
    FINSI
}
FINSI
SI (mere = "A")
ALORS
{
    SI(pere = "A")
    ALORS
    {
        Afficher ("L'enfant ne sera ni B, ni AB")
        Afficher ("L'enfant peut être du groupe A ou O")
    }
    FINSI
}
FINSI
SI (mere = "A")
ALORS
{
    SI(pere = "O")
    ALORS
    {
        Afficher ("L'enfant ne sera ni B, ni AB")
        Afficher ("L'enfant peut être du groupe A ou O")
    }
    FINSI
}
```

```
FINSI
SI (mere = "O")
ALORS
{
    SI(pere = "A")
    ALORS
    {
        Afficher ("L'enfant ne sera ni B, ni AB")
        Afficher ("L'enfant peut être du groupe A ou O")
    }
    FINSI
}
FINSI
SI (mere = "B")
ALORS
{
    SI(pere = "B")
    ALORS
    {
        Afficher ("L'enfant ne sera ni A, ni AB")
        Afficher ("L'enfant peut être du groupe B ou O")
    }
    FINSI
}
FINSI
SI (mere = "B")
ALORS
{
    SI(pere = "O")
    ALORS
    {
        Afficher ("L'enfant ne sera ni A, ni AB")
        Afficher ("L'enfant peut être du groupe B ou O")
    }
    FINSI
}
FINSI
SI (mere = "O")
ALORS
{
    SI(pere = "B")
    ALORS
    {
        Afficher ("L'enfant ne sera ni A, ni AB")
        Afficher ("L'enfant peut être du groupe B ou O")
    }
    FINSI
}
FINSI
SI(mere = "AB")
ALORS
{
    Afficher ("L'enfant ne peut être O")
    Afficher ("L'enfant peut être du groupe A ou B ou AB")
}
FINSI
SI(pere = "AB")
ALORS
{
    Afficher ("L'enfant ne peut être O")
    Afficher ("L'enfant peut être du groupe A ou B ou AB")
```

```
}
FINSI
FIN
//-------------------------------------------------------------------------------
```

2.2.2. Instruction conditionnelle imbriquée : Si alors sinon

```
//-------------------------------------------------------------------------------
//TITRE DU PROGRAMME:         Indice de Masse Corporelle (IMC version simplifiée)
//-------------------------------------------------------------------------------
//-------------------------------------------------------------------------------
//                     PROGRAMME PRINCIPAL
//-------------------------------------------------------------------------------
//          Déclaration des constantes et des variables
//-------------------------------------------------------------------------------
//-------------------------------------------------------------------------------
//Déclaration des variables
VAR:
imc :REEL
poids :ENTIER
taille :REEL
//-------------------------------------------------------------------------------
DEBUT
    ECRIRE("Entrez votre poids en kilogramme, puis votre taille en mètre :")
    LIRE(poids, taille)
    imc <-- poids / (taille * taille)
    SI(imc <= 25)
    ALORS
    {
        ECRIRE("Poids normal")
    }
    SINON
    {
        SI(imc <= 30)
        ALORS
        {
            ECRIRE("Surpoids")
        }
        SINON
        {
            SI(imc <= 35)
            ALORS
            {
                ECRIRE("Obésité modérée")
            }
            SINON
            {
                SI(imc <= 40)
                ALORS
                {
                    ECRIRE("Obésité sévère")
                }
                SINON
                {
                    SI(imc > 40)
                    ALORS
```

```
                    {
                        ECRIRE("Obésité massive")
                    }
                }
                FINSI
            }
            FINSI
        }
        FINSI
    }
    FINSI
FIN
//--------------------------------------------------------------------------------
```

2.2.3. Instruction conditionnelle imbriquée : Cas parmi

```
//--------------------------------------------------------------------------------
//TITRE DU PROGRAMME:         Test de la filiation (booléen simple)
//--------------------------------------------------------------------------------
//--------------------------------------------------------------------------------
//                    PROGRAMME PRINCIPAL
//--------------------------------------------------------------------------------
//          Déclaration des constantes et des variables
//--------------------------------------------------------------------------------
//--------------------------------------------------------------------------------
//Déclaration des variables
Var :
    mere, pere  :CHAINE
//--------------------------------------------------------------------------------
DEBUT
ECRIRE("Veuillez entrer la(les) lettre(s) relative(s) au groupe sanguin de la mère
[en majuscule]")
LIRE(mere)
ECRIRE("Veuillez entrer la(les) lettre(s) relative(s) au groupe sanguin du père [en
majuscule]")
LIRE(pere)
CAS( mere )PARMI:
CAS1: "O"
    CAS( pere )PARMI:
    CAS1: "O"
        {
            Afficher ("L'enfant sera de type O")
        }
    CAS2: "A"
        {
            Afficher ("L'enfant ne sera ni B, ni AB")
            Afficher ("L'enfant peut être du groupe A ou O")
        }
    CAS3: "B"
        {
            Afficher ("L'enfant ne sera ni A, ni AB")
            Afficher ("L'enfant peut être du groupe B ou O")
        }
    CAS4: "AB"
        {
            Afficher ("L'enfant ne peut être O")
            Afficher ("L'enfant peut être du groupe A ou B ou AB")
```

```
        }
    FINCASPARMI
CAS2: "A"
    CAS( pere )PARMI:
    CAS1: "O"
        {
            Afficher ("L'enfant ne sera ni B, ni AB")
            Afficher ("L'enfant peut être du groupe A ou O")
        }
    CAS2: "A"
        {
            Afficher ("L'enfant ne sera ni B, ni AB")
            Afficher ("L'enfant peut être du groupe A ou O")
        }
    CAS3: "B"
        {
            Afficher ("Cas indéterminé")
        }
    CAS4: "AB"
        {
            Afficher ("L'enfant ne peut être O")
            Afficher ("L'enfant peut être du groupe A ou B ou AB")
        }
    FINCASPARMI
CAS3: "B"
    CAS( pere )PARMI:
    CAS1: "O"
        {
            Afficher ("L'enfant ne sera ni A, ni AB")
            Afficher ("L'enfant peut être du groupe B ou O")
        }
    CAS2: "A"
        {
            Afficher ("Cas indéterminé")
        }
    CAS3: "B"
        {
            Afficher ("L'enfant ne sera ni A, ni AB")
            Afficher ("L'enfant peut être du groupe B ou O")
        }
    CAS4: "AB"
        {
            Afficher ("L'enfant ne peut être O")
            Afficher ("L'enfant peut être du groupe A ou B ou AB")
        }
    FINCASPARMI
CAS4: "AB"
            Afficher ("L'enfant ne peut être O")
            Afficher ("L'enfant peut être du groupe A ou B ou AB")
FINCASPARMI
FIN
//-------------------------------------------------------------------------------
```

2.2.4. Instruction conditionnelle imbriquée : Cas parmi par défaut

```
//-------------------------------------------------------------------------------
//TITRE DU PROGRAMME:        Test de la filiation (booléen simple)
//-------------------------------------------------------------------------------
//-------------------------------------------------------------------------------
//                    PROGRAMME PRINCIPAL
//-------------------------------------------------------------------------------
//          Déclaration des constantes et des variables
//-------------------------------------------------------------------------------
//-------------------------------------------------------------------------------
//Déclaration des variables
Var :
    mere, pere  :CHAINE
//-------------------------------------------------------------------------------
DEBUT
ECRIRE("Veuillez entrer la(les) lettre(s) relative(s) au groupe sanguin de la mère
[en majuscule]")
LIRE(mere)
ECRIRE("Veuillez entrer la(les) lettre(s) relative(s) au groupe sanguin du père [en
majuscule]")
LIRE(pere)

CAS( mere )PARMI:
CAS1: "O"
    CAS( pere )PARMI:
    CAS1: "O"
        {
            Afficher ("L'enfant sera de type O")
        }
    CAS2: "A"
        {
            Afficher ("L'enfant ne sera ni B, ni AB")
            Afficher ("L'enfant peut être du groupe A ou O")
        }
    CAS3: "B"
        {
            Afficher ("L'enfant ne sera ni A, ni AB")
            Afficher ("L'enfant peut être du groupe B ou O")
        }
    CAS4: "AB"
        {
            Afficher ("L'enfant ne peut être O")
            Afficher ("L'enfant peut être du groupe A ou B ou AB")
        }
    PARDEFAUT:
        Afficher ("Veuillez vérifier les informations entrées pour le groupe
sanguin du père !")
    FINCASPARMI
CAS2: "A"
    CAS( pere )PARMI:
    CAS1: "O"
        {
            Afficher ("L'enfant ne sera ni B, ni AB")
            Afficher ("L'enfant peut être du groupe A ou O")
        }
    CAS2: "A"
        {
            Afficher ("L'enfant ne sera ni B, ni AB")
            Afficher ("L'enfant peut être du groupe A ou O")
        }
    CAS3: "B"
```

```
        {
            Afficher ("Cas indéterminé")
        }
    CAS4: "AB"
        {
            Afficher ("L'enfant ne peut être O")
            Afficher ("L'enfant peut être du groupe A ou B ou AB")
        }
    PARDEFAUT:
        Afficher ("Veuillez vérifier les informations entrées pour le groupe
sanguin du père !")
    FINCASPARMI
CAS3: "B"
    CAS( pere )PARMI:
    CAS1: "O"
        {
            Afficher ("L'enfant ne sera ni A, ni AB")
            Afficher ("L'enfant peut être du groupe B ou O")
        }
    CAS2: "A"
        {
            Afficher ("Cas indéterminé")
        }
    CAS3: "B"
        {
            Afficher ("L'enfant ne sera ni A, ni AB")
            Afficher ("L'enfant peut être du groupe B ou O")
        }
    CAS4: "AB"
        {
            Afficher ("L'enfant ne peut être O")
            Afficher ("L'enfant peut être du groupe A ou B ou AB")
        }
    PARDEFAUT:
        Afficher ("Veuillez vérifier les informations entrées pour le groupe
sanguin du père !")
    FINCASPARMI
CAS4: "AB"
            Afficher ("L'enfant ne peut être O")
            Afficher ("L'enfant peut être du groupe A ou B ou AB")
PARDEFAUT:
            Afficher ("Veuillez vérifier les informations entrées pour le groupe
sanguin de la mère !")
FINCASPARMI
FIN
//-------------------------------------------------------------------------------
```

2.2.5. Instructions conditionnelles diverses imbriquées

```
//-------------------------------------------------------------------------------
//-------------------------------------------------------------------------------
//TITRE DU PROGRAMME:         Tendance à l'obésité
//-------------------------------------------------------------------------------
//-------------------------------------------------------------------------------
//                    PROGRAMME PRINCIPAL
//-------------------------------------------------------------------------------
//          Déclaration des constantes et des variables
```

```
//-------------------------------------------------------------------------------
//-------------------------------------------------------------------------------
//Déclaration des constantes
CONST:
homme <-- 100 :ENTIER
femme <-- 90 :ENTIER
sexe_masculin<-- 'M' :CARACTERE
sexe_feminin<-- 'F' :CARACTERE
//-------------------------------------------------------------------------------
//Déclaration des variables
VAR:
mesure :ENTIER
sexe :BOOLEEN
//-------------------------------------------------------------------------------
DEBUT
    ECRIRE("Veuillez renseigner le programme sur votre sexe: M pour masculin, et F
pour féminin.")
    LIRE(sexe)
    ECRIRE("Veuillez entrer la valeur entière relative à votre tour de taille,
SVP.")
    LIRE(mesure)
    CAS(sexe)PARMI
        CAS1: sexe_masculin
            {
                  SI(mesure > homme)
                  ALORS
                  {
                      ECRIRE("Veuillez consulter votre médecin pour une possible
tendance à l'obésité")
                  }
                  SINON
                  {
                      ECRIRE("Vous n'avez pas, a priori, de tendance à l'obésité")
                  }
                  FINSI
            }
        CAS2: sexe_feminin
            {
                  SI(mesure > femme)
                  ALORS
                  {
                      ECRIRE("Veuillez consulter votre médecin pour une possible
tendance à l'obésité")
                  }
                  SINON
                  {
                      ECRIRE("Vous n'avez pas, a priori, de tendance à l'obésité")
                  }
                  FINSI
            }
    FINCASPARMI

FIN
//-------------------------------------------------------------------------------
```

2.3. Instructions répétitives : Tantque et Répéter

2.3.1. Présentation des problèmes liés aux instructions répétitives (ou boucles)

2.3.2. Instruction « Tantque »

```
//-------------------------------------------------------------------------------
//-------------------------------------------------------------------------------
//PROGRAMME PRINCIPAL:       Affichage d'un agenda journalier
//-------------------------------------------------------------------------------
//-------------------------------------------------------------------------------
//Déclaration des variables
VAR:
nom_jour :CHAINE
heure <-- 8 :ENTIER
//-------------------------------------------------------------------------------
DEBUT
ECRIRE("Veuillez entrer le nom du jour d'édition de l'agenda (p.ex.:MARDI).")
LIRE( nom_jour )
ECRIRE( nom_jour )
TANTQUE( heure <= 18 )
FAIRE
{
    ECRIRE( heure, "H00 : ")
    heure  <-- heure + 1
}
FINTANTQUE
FIN
//-------------------------------------------------------------------------------
```

2.3.3. Instruction « Répéter jusqu'à »

```
//-------------------------------------------------------------------------------
//-------------------------------------------------------------------------------
//PROGRAMME PRINCIPAL:       Affichage d'un agenda journalier
//-------------------------------------------------------------------------------
//-------------------------------------------------------------------------------
//Déclaration des variables
VAR:
nom_jour :CHAINE
heure <-- 8 :ENTIER
//-------------------------------------------------------------------------------
DEBUT
ECRIRE("Veuillez entrer le nom du jour d'édition de l'agenda (p.ex.:MARDI).")
LIRE( nom_jour )
ECRIRE( nom_jour )
    REPETER
    {
        ECRIRE( heure, "H00 : ")
        heure  <-- heure + 1
```

```
    }
    JUSQU'A( heure > 18 )
FIN
//-------------------------------------------------------------------------------
```

2.3.4. Instruction « Répéter tantque »

```
//-------------------------------------------------------------------------------
//-------------------------------------------------------------------------------
//PROGRAMME PRINCIPAL:       Affichage d'un agenda journalier
//-------------------------------------------------------------------------------
//-------------------------------------------------------------------------------
//Déclaration des variables
VAR:
nom_jour :CHAINE
heure <-- 8 :ENTIER
//-------------------------------------------------------------------------------
DEBUT
ECRIRE("Veuillez entrer le nom du jour d'édition de l'agenda (p.ex.:MARDI).")
LIRE( nom_jour )
ECRIRE( nom_jour )
    REPETER
    {
        ECRIRE( heure, "H00 : ")
        heure  <-- heure + 1
    }
    TANTQUE( heure <= 18 )
FIN
//-------------------------------------------------------------------------------
```

2.3.5. Boucle et sortie de boucle : sortie prématurée

```
//-------------------------------------------------------------------------------
//-------------------------------------------------------------------------------
//PROGRAMME PRINCIPAL:       Calendrier mensuel
//-------------------------------------------------------------------------------
//          Déclaration des constantes et des variables
//-------------------------------------------------------------------------------
//-------------------------------------------------------------------------------
//Déclaration des constantes
CONST:
janvier <-- 31 :ENTIER
fevrier <-- 28 :ENTIER
mars <-- 31 :ENTIER
avril <-- 30 :ENTIER
mai <-- 31 :ENTIER
juin <-- 30 :ENTIER
juillet <-- 31 :ENTIER
aout <-- 31 :ENTIER
```

```
septembre <-- 30 :ENTIER
octobre <-- 31 :ENTIER
novembre <-- 30 :ENTIER
decembre <-- 31 :ENTIER
//-------------------------------------------------------------------------------
//Déclaration des variables
VAR:
    mois, nombre_jours :ENTIER
    jour <-- 1 :ENTIER
//-------------------------------------------------------------------------------
DEBUT
ECRIRE("Veuillez entrer le nom du mois pour faire afficher le calendrier (en
majuscule):")
LIRE(mois)
ECRIRE("Mois de: ", mois)
TANTQUE( jour <= 31 )
FAIRE
{
    CAS(mois)PARMI
        CAS1: "JANVIER"
            {
                nombre_jours <-- janvier
                ECRIRE(jour)
            }
        CAS2: "FEVRIER"
            {
                nombre_jours <-- fevrier
                ECRIRE(jour)
            }
        CAS3: "MARS"
            {
                nombre_jours <-- mars
                ECRIRE(jour)
            }
        CAS4: "AVRIL"
            {
                nombre_jours <-- avril
                ECRIRE(jour)
            }
        CAS5: "MAI"
            {
                nombre_jours <-- mai
                ECRIRE(jour)
            }
        CAS6: "JUIN"
            {
                nombre_jours <-- juin
                ECRIRE(jour)
            }
        CAS7: "JUILLET"
            {
                nombre_jours <-- juillet
                ECRIRE(jour)
            }
        CAS8: "AOUT"
            {
                nombre_jours <-- aout
                ECRIRE(jour)
            }
        CAS9: "SEPTEMBRE"
            {
```

```
                nombre_jours <-- septembre
                ECRIRE(jour)
            }
        CAS10: "OCTOBRE"
            {
                nombre_jours <-- octobre
                ECRIRE(jour)
            }
        CAS11: "NOVEMBRE"
            {
                nombre_jours <-- novembre
                ECRIRE(jour)
            }
        CAS12: "DECEMBRE"
            {
                nombre_jours <-- decembre
                ECRIRE(jour)
            }
      PARDEFAUT:
            {
                ECRIRE("Erreur ! Veuillez contrôler les informations fournies.")
                jour <-- 32 //SORTIE PREMATUREE DU TANTQUE
            }
    FINCASPARMI
    SI( jour > nombre_jours )
    ALORS
    {
        jour <-- 32 //SORTIE PREMATUREE DU TANTQUE
    }
    FINSI
}
FINTANTQUE
FIN
//-------------------------------------------------------------------------------
```

2.3.6. Boucle et sortie de boucle : persistance dans une boucle

```
//-------------------------------------------------------------------------------
//-------------------------------------------------------------------------------
//PROGRAMME PRINCIPAL:       Machine à sous (bouclage aléatoire)
//-------------------------------------------------------------------------------
//          Déclaration des constantes et des variables
//-------------------------------------------------------------------------------
//-------------------------------------------------------------------------------
//Déclaration des constantes
CONST:
//-------------------------------------------------------------------------------
//Déclaration des variables
VAR:
duree :ENTIER
//-------------------------------------------------------------------------------
DEBUT
LIRE( nombre_jetons )
memoire <-- nombre_jetons
TANTQUE( nombre_jetons <> 0 )
```

```
FAIRE
{
    SI( RANDOM(memoire) = nombre_jetons ) //RANDOM(memoire) renvoie un entier
                                          //compris entre 0 et memoire
    ALORS
    {
        nombre_jetons <-- nombre_jetons + RANDOM(1) //RANDOM(1) renvoie 0 ou 1
aléatoirement
        memoire <-- nombre_jetons
    }
    FINSI
    nombre_jetons <-- nombre_jetons - 1
}
FINTANTQUE
FIN
//-------------------------------------------------------------------------------
```

2.3.7. Boucles imbriquées

```
//-------------------------------------------------------------------------------
//-------------------------------------------------------------------------------
//PROGRAMME PRINCIPAL: Affichage d'un agenda journalier évolué (TANTQUE/TANTQUE)
//-------------------------------------------------------------------------------
//-------------------------------------------------------------------------------
//Déclaration des constantes
CONST:
//-------------------------------------------------------------------------------
//Déclaration des variables
VAR:
nombre_jours :ENTIER
jour <-- 1 :ENTIER
heure <-- 8 :ENTIER
//-------------------------------------------------------------------------------
DEBUT
ECRIRE("Veuillez entrer le nombre de jours d'édition de l'agenda.")
LIRE( nombre_jours )
TANTQUE( jour <= nombre_jours )
FAIRE
{
    ECRIRE( "Jour numéro: ", jour )
    TANTQUE( heure <= 18 )
    FAIRE
    {
        ECRIRE( heure, "H00 : ")
        heure  <-- heure + 1
    }
    FINTANTQUE
    jour  <--  jour + 1
    heure <-- 8
}
FINTANTQUE
FIN
//-------------------------------------------------------------------------------
```

```
//-------------------------------------------------------------------------------
//-------------------------------------------------------------------------------
//PROGRAMME PRINCIPAL: Affichage d'un agenda journalier (REPETER/REPETER JUSQU'A)
//-------------------------------------------------------------------------------
//-------------------------------------------------------------------------------
//Déclaration des constantes
CONST:
//-------------------------------------------------------------------------------
//Déclaration des variables
VAR:
nombre_jours :ENTIER
jour <-- 1 :ENTIER
heure <-- 8 :ENTIER
//-------------------------------------------------------------------------------
DEBUT
ECRIRE("Veuillez entrer le nombre de jours d'édition de l'agenda.")
LIRE( nombre_jours )
REPETER
{
    ECRIRE( "Jour numéro: ", jour )
    REPETER
    {
        ECRIRE( heure, "H00 : ")
        heure  <-- heure + 1
    }
    JUSQU'A( heure > 18 )
    jour  <--  jour + 1
    heure <-- 8
}
JUSQU'A( jour > nombre_jours )
FIN
//-------------------------------------------------------------------------------
```

```
//-------------------------------------------------------------------------------
//-------------------------------------------------------------------------------
//PROGRAMME PRINCIPAL: Affichage d'un agenda journalier (REPETER/REPETER TANTQUE)
//-------------------------------------------------------------------------------
//-------------------------------------------------------------------------------
//Déclaration des constantes
CONST:
//-------------------------------------------------------------------------------
//Déclaration des variables
VAR:
nombre_jours :ENTIER
jour <-- 1 :ENTIER
heure <-- 8 :ENTIER
//-------------------------------------------------------------------------------
DEBUT
ECRIRE("Veuillez entrer le nombre de jours d'édition de l'agenda.")
LIRE( nombre_jours )
REPETER
{
    ECRIRE( "Jour numéro: ", jour )
    REPETER
    {
```

```
        ECRIRE( heure, "H00 : ")
        heure  <-- heure + 1
    }
    TANTQUE( heure <= 18 )
    jour  <--  jour + 1
    heure <-- 8
}
TANTQUE( jour <= nombre_jours )
FIN
//-------------------------------------------------------------------------------
```

2.4. Instructions répétitives : Pour

2.4.1. Présentation des problèmes liés aux instructions répétitives (ou boucles)

2.4.2. Instruction « Pour »

```
//-------------------------------------------------------------------------------
//-------------------------------------------------------------------------------
//PROGRAMME PRINCIPAL: Affichage d'un agenda journalier
//-------------------------------------------------------------------------------
//-------------------------------------------------------------------------------
//Déclaration des variables
VAR:
nom_jour :CHAINE
heure <-- 8 :ENTIER
//-------------------------------------------------------------------------------
DEBUT
ECRIRE("Veuillez entrer le nom du jour d'édition de l'agenda (p.ex.:MARDI).")
LIRE( nom_jour )
ECRIRE( nom_jour )
    POUR( heure <-- 8 :ENTIER a 18 au pas de +1 )
    FAIRE
    {
        ECRIRE( heure, "H00 : ")
    }
    FINPOUR
FIN
//-------------------------------------------------------------------------------
```

2.4.3. Boucle et sortie de boucle : sortie prématurée

```
//-------------------------------------------------------------------------------
//-------------------------------------------------------------------------------
```

```
//PROGRAMME PRINCIPAL:       Calendrier mensuel
//-------------------------------------------------------------------------------
//          Déclaration des constantes et des variables
//-------------------------------------------------------------------------------
//-------------------------------------------------------------------------------
//Déclaration des constantes
CONST:
janvier <-- 31 :ENTIER
fevrier <-- 28 :ENTIER
mars <-- 31 :ENTIER
avril <-- 30 :ENTIER
mai <-- 31 :ENTIER
juin <-- 30 :ENTIER
juillet <-- 31 :ENTIER
aout <-- 31 :ENTIER
septembre <-- 30 :ENTIER
octobre <-- 31 :ENTIER
novembre <-- 30 :ENTIER
decembre <-- 31 :ENTIER
//-------------------------------------------------------------------------------
//Déclaration des variables
VAR:
    mois, nombre_jours :ENTIER
//-------------------------------------------------------------------------------
DEBUT
ECRIRE("Veuillez entrer le nom du mois pour faire afficher le calendrier (en
majuscule):")
LIRE(mois)
ECRIRE("Mois de: ", mois)
POUR(jour <-- 1 :ENTIER a 31 au pas de +1 )
FAIRE
{
    CAS(mois)PARMI
        CAS1: "JANVIER"
            {
                nombre_jours <-- janvier
                ECRIRE(jour)
            }
        CAS2: "FEVRIER"
            {
                nombre_jours <-- fevrier
                ECRIRE(jour)
            }
        CAS3: "MARS"
            {
                nombre_jours <-- mars
                ECRIRE(jour)
            }
        CAS4: "AVRIL"
            {
                nombre_jours <-- avril
                ECRIRE(jour)
            }
        CAS5: "MAI"
            {
                nombre_jours <-- mai
                ECRIRE(jour)
            }
        CAS6: "JUIN"
            {
                nombre_jours <-- juin
```

```
                ECRIRE(jour)
            }
        CAS7: "JUILLET"
            {
                nombre_jours <-- juillet
                ECRIRE(jour)
            }
        CAS8: "AOUT"
            {
                nombre_jours <-- aout
                ECRIRE(jour)
            }
        CAS9: "SEPTEMBRE"
            {
                nombre_jours <-- septembre
                ECRIRE(jour)
            }
        CAS10: "OCTOBRE"
            {
                nombre_jours <-- octobre
                ECRIRE(jour)
            }
        CAS11: "NOVEMBRE"
            {
                nombre_jours <-- novembre
                ECRIRE(jour)
            }
        CAS12: "DECEMBRE"
            {
                nombre_jours <-- decembre
                ECRIRE(jour)
            }
      PARDEFAUT:
            {
                ECRIRE("Erreur ! Veuillez contrôler les informations fournies.")
                jour <-- 32 //SORTIE PREMATUREE DU TANTQUE
            }
    FINCASPARMI
    SI( jour > nombre_jours )
    ALORS
    {
        jour <-- 32 //SORTIE PREMATUREE DU TANTQUE
    }
    FINSI
}
FINPOUR
FIN
//-------------------------------------------------------------------------------
```

2.4.4. Boucle et sortie de boucle : persistance dans une boucle

```
//-------------------------------------------------------------------------------
//-------------------------------------------------------------------------------
//PROGRAMME PRINCIPAL:       Machine à sous (bouclage aléatoire)
//-------------------------------------------------------------------------------
```

```
//          Déclaration des constantes et des variables
//-------------------------------------------------------------------------------
//-------------------------------------------------------------------------------
//Déclaration des constantes
CONST:
//-------------------------------------------------------------------------------
//Déclaration des variables
VAR:
nombre_jetons, memoire :ENTIER

//-------------------------------------------------------------------------------
DEBUT
LIRE( nombre_jetons )
memoire <-- nombre_jetons
POUR( nombre_jetons <-- memoire :ENTIER a 0 au pas de -1 )
FAIRE
{
    SI( RANDOM(memoire) = nombre_jetons )
    ALORS
    {
        nombre_jetons <-- nombre_jetons + RANDOM(1) //RANDOM(1) renvoie 0 ou 1
aléatoirement
        memoire <-- nombre_jetons
    }
    FINSI
    nombre_jetons <-- nombre_jetons - 1
}
FINPOUR
FIN
//-------------------------------------------------------------------------------
```

2.4.5. Boucle et sortie de boucle : bornes variables

```
//-------------------------------------------------------------------------------
//-------------------------------------------------------------------------------
//PROGRAMME PRINCIPAL: Affichage d'un agenda journalier sur plusieurs jours
//-------------------------------------------------------------------------------
//-------------------------------------------------------------------------------
//Déclaration des constantes
CONST:
//-------------------------------------------------------------------------------
//Déclaration des variables
VAR:
nombre_jours :ENTIER
//-------------------------------------------------------------------------------
DEBUT
ECRIRE("Veuillez entrer le nombre de jours d'édition de l'agenda.")
LIRE( nombre_jours )
POUR( jour <-- 1 :ENTIER a nombre_jours au pas de +1 )
FAIRE
{
    ECRIRE( "Jour numéro: ", jour )
    POUR( heure <-- 8 :ENTIER a 18 au pas de +1 )
    FAIRE
    {
```

```
        ECRIRE( heure, "H00 : ")
    }
    FINPOUR
}
FINPOUR
FIN
//-------------------------------------------------------------------------------
```

2.4.6. Boucle et sortie de boucle : pas variable

```
//-------------------------------------------------------------------------------
//-------------------------------------------------------------------------------
//                            TITRE DU PROGRAMME:
//-------------------------------------------------------------------------------
//-------------------------------------------------------------------------------
//PROGRAMME PRINCIPAL:        Calcul de la monnaie en euro
//-------------------------------------------------------------------------------
//          Déclaration des constantes et des variables
//-------------------------------------------------------------------------------
//-------------------------------------------------------------------------------
//Déclaration des constantes
CONST:
//-------------------------------------------------------------------------------
//Déclaration des variables
VAR:
//-------------------------------------------------------------------------------
DEBUT
ALGORITHME Choix de la monnaie en euro
VAR :
    prix                                        : ENTIER
    nb_billets_100 <-- 0, nb_billets_50 <-- 0 : ENTIER
    nb_billets_20 <-- 0, nb_billets_10 <-- 0  : ENTIER
    nb_billets_5 <-- 0                          : ENTIER
DEBUT
    ECRIRE ("Entrez un prix en euro (sans les centimes):")
    LIRE( prix )
POUR( monnaie <-- prix :ENTIER a 0 au pas de - euro)
FAIRE
{
    SI( monnaie >= 100 )
    ALORS
        euro <-- 100
        nb_billets_100 <-- nb_billets_100 + 1
    SINON
        SI( monnaie >= 50 )
        ALORS
            euro <-- 50
            nb_billets_50 <-- nb_billets_50 + 1
        SINON
            SI( monnaie >= 20 )
            ALORS
                euro <-- 20
                nb_billets_20 <-- nb_billets_20 + 1
            SINON
                SI( monnaie >= 10 )
                ALORS
                    euro <-- 10
                    nb_billets_10 <-- nb_billets_10 + 1
                SINON
```

```
                    SI( monnaie >= 5 )
                    ALORS
                        euro <-- 5
                        nb_billets_5 <-- nb_billets_5 + 1
                    FINSI
                FINSI
            FINSI
        FINSI
    FINSI
}
FINPOUR
    ECRIRE ("Billets de 100:", nb_billets_100)
    ECRIRE ("Billets de 50:", nb_billets_50)
    ECRIRE ("Billets de 20:", nb_billets_20)
    ECRIRE ("Billets de 10:", nb_billets_10)
    ECRIRE ("Billets de 5:", nb_billets_5)
    ECRIRE ("Monnaie en pièce(s):", monnaie)
FIN
//-------------------------------------------------------------------------------
```

2.4.7. Boucles imbriquées : bornes fixes

```
//-------------------------------------------------------------------------------
//-------------------------------------------------------------------------------
//PROGRAMME PRINCIPAL:        Table de multiplication complète
//-------------------------------------------------------------------------------
//-------------------------------------------------------------------------------
DEBUT
POUR( chiffre1 <-- 0 :ENTIER a 10 au pas de +1 )
FAIRE
{
    ECRIRE( "Table de multiplication de: ", chiffre1 )
    POUR( chiffre2 <-- 0 :ENTIER a 10 au pas de +1)
    FAIRE
    {
        ECRIRE( chiffre1, " x ", chiffre2, " = ", chiffre1 * chiffre2 )
    }
    FINPOUR
}
FINPOUR
FIN
//-------------------------------------------------------------------------------
```

2.4.8. Boucles imbriquées : bornes imbriquées

```
//-------------------------------------------------------------------------------
//-------------------------------------------------------------------------------
//PROGRAMME PRINCIPAL:        Affichage d'une droite en diagonale
//-------------------------------------------------------------------------------
//-------------------------------------------------------------------------------
DEBUT
```

```
POUR( abscisse <-- 0 :ENTIER a 50 au pas de +1 )
FAIRE
{
    POUR( ordonnee <-- abscisse :ENTIER a 50 au pas de +1)
    FAIRE
    {
        ECRIRE_POSITION("x", abscisse, ordonnee )
    }
    FINPOUR
}
FINPOUR
FIN
//-------------------------------------------------------------------------------
```

2.5. Instructions de rupture de séquences imbriquées

2.5.1. Problème résolu 1

```
//-------------------------------------------------------------------------------
//-------------------------------------------------------------------------------
//PROGRAMME PRINCIPAL:       Nombre de jours par mois
//-------------------------------------------------------------------------------
//-------------------------------------------------------------------------------
//Déclaration des variables
VAR:
    mois, annee, jours  : ENTIER
    bissextile          : BOOLEEN
//-------------------------------------------------------------------------------
DEBUT
   ECRIRE( "Saisir le mois et l'année" )
   LIRE( mois, annee )
   CAS( mois ) PARMI
       CAS 2 :    jours  <--  28
       CAS 4 :    jours  <--  30
       CAS 6 :    jours  <--  30
       CAS 9 :    jours  <--  30
       CAS 11 :   jours  <--  30
       PAR DEFAUT :     jours  <--  31
   FINCASPARMI
SI ( jours = 28 )
ALORS
        SI ( ( annee MOD 400 ) = 0 )
        ALORS
                bissextile  <--  VRAI
        SINON
                SI ( ( annee MOD 4 ) = 0 )
                ALORS
                        SI ( ( annee MOD 100 ) = 0 )
                        ALORS
                                bissextile  <--  FAUX
                        SINON
                                bissextile  <--  VRAI
```

```
                        FINSI
                SINON
                        bissextile  <--  FAUX
                FINSI
        FINSI
        SI ( bissextile )
        ALORS
                jours  <--  jours + 1
        FINSI
FINSI
ECRIRE( "Ce mois comporte : ", jours, " jours" )
FIN
//-------------------------------------------------------------------------------
```

2.5.2. Problème résolu 2

```
//-------------------------------------------------------------------------------
//-------------------------------------------------------------------------------
//PROGRAMME PRINCIPAL:       Extraire la valeur entière
//-------------------------------------------------------------------------------
//-------------------------------------------------------------------------------
VAR:
    nombre              : REEL
    valeur_entiere <-- 0      : ENTIER
//-------------------------------------------------------------------------------
DEBUT
    LIRE( nombre )
    SI( nombre > 0 )
    ALORS
      TANTQUE( nombre <> 0 )
      FAIRE
            valeur_entiere <-- valeur_entiere + 1
            nombre <-- nombre - 1
      FINTANTQUE
    SINON
      TANTQUE( nombre <> 0)
      FAIRE
            valeur_entiere <-- valeur_entiere + 1
            nombre <-- nombre + 1
      FINTANTQUE
    FINSI
    ECRIRE ("Valeur entière de",nombre,"est",valeur_entiere)
FIN
//-------------------------------------------------------------------------------
```

2.5.3. Problème résolu 3

```
//-------------------------------------------------------------------------------
//-------------------------------------------------------------------------------
```

```
//PROGRAMME PRINCIPAL:       Calcul les points d'un mot (non posé) au SCRABBLE
//-------------------------------------------------------------------------------
//-------------------------------------------------------------------------------
CONST:
    k,w,x,y,z <-- 10          : ENTIER
    j,q <-- 8                 : ENTIER
    f,h,v <-- 4               : ENTIER
    b,c,p <-- 3               : ENTIER
    d,g,m <-- 2               : ENTIER
    a,e,i,l,n,o,r,s,t,u <-- 1 : ENTIER
//-------------------------------------------------------------------------------
VAR:
    mot           : CHAINE
    lettre        : CARACTERE
    points <-- 0  : ENTIER
//-------------------------------------------------------------------------------
DEBUT
    LIRE( mot )
    POUR(position <-- 1 : ENTIER a 7 au pas de +1)
    FAIRE
        lettre <-- EXTRAIRE( mot, position )
        CAS( lettre )PARMI:
            CAS1:     'a' points <-- points + a
            CAS...:   ...
            CAS26:    'z' points <-- points + z
        FINCASPARMI
    FINPOUR
    ECRIRE("Le mot:",mot,"vaut:",points,"points.")
FIN
//-------------------------------------------------------------------------------
```

2.5.4. Problème résolu 4

```
//-------------------------------------------------------------------------------
//-------------------------------------------------------------------------------
//TITRE DU PROGRAMME: Temps de remontée d'un plongeur
//-------------------------------------------------------------------------------
//-------------------------------------------------------------------------------
//problème relatif à une plongée dont la profondeur est comprise entre 0 et 100m
//-------------------------------------------------------------------------------
//                    PROGRAMME PRINCIPAL
//-------------------------------------------------------------------------------
//          Déclaration des constantes et des variables
//-------------------------------------------------------------------------------
//Déclaration des variables
VAR:
profondeur <-- 0 :ENTIER
dureetotale <-- 0 :ENTIER
palier <-- 3 :ENTIER                        // 3 metres
dureepalier <-- 0 :ENTIER                        // 10 secondes
duree <-- 10 :ENTIER
etape <-- 0 :ENTIER
//-------------------------------------------------------------------------------
DEBUT
profondeurplongee <-- SAISIR_PROFONDEUR_CLAVIER()    //saisir la valeur absolue
TANTQUE(profondeur < profondeurplongee)
{
```

```
      etape <-- etape + 1           //compteur pour suivre chaque incrémentation
      profondeur <-- profondeur + palier
      dureepalier <-- dureepalier + duree
      dureetotale <-- dureetotale + dureepalier
//<<<<<<<<<<<<<<<<<< TABLE DE DONNEES
      SI(profondeur = 30)
      {
            palier <-- 10           // 10 metres
            duree <-- 25            // 25 secondes
      }
}
dureetotale <-- dureetotale + (profondeur * 1)       // vitesse de remontée = 1m/s
ECRIRE( dureetotale )
FIN
//-------------------------------------------------------------------------------
```

3. Structure de données avancée : structures linéaires

3.1. Annexe 1

3.1.1. Sous Annexe 1

3.1.2. Sous Annexe 2

3.2. Annexe 2

3.2.1. Sous Annexe 1

3.2.2. Sous Annexe 2

3.2.3. Sous Annexe 3

3.2.4. Sous Annexe 4

3.2.5. Sous Annexe 5

3.3. Annexe 3

3.4. Annexe 4

4. Structure de programme avancée

4.1. Annexe 1

4.1.1. Sous Annexe 1

4.1.2. Sous Annexe 2

4.2. Annexe 2

4.2.1. Sous Annexe 1

4.2.2. Sous Annexe 2

4.2.3. Sous Annexe 3

4.2.4. Sous Annexe 4

4.2.5. Sous Annexe 5

4.3. Annexe 3

4.4. Annexe 4

5. Structure de données avancée : structures non linéaires et fichiers

5.1. Annexe 1

5.1.1. Sous Annexe 1

5.1.2. Sous Annexe 2

5.2. Annexe 2

5.2.1. Sous Annexe 1

5.2.2. Sous Annexe 2

5.2.3. Sous Annexe 3

5.2.4. Sous Annexe 4

5.2.5. Sous Annexe 5

5.3. Annexe 3

5.4. Annexe 4

6. Structure dynamique de données

6.1. Annexe 1

6.1.1. Sous Annexe 1

6.1.2. Sous Annexe 2

6.2. Annexe 2

6.2.1. Sous Annexe 1

6.2.2. Sous Annexe 2

6.2.3. Sous Annexe 3

6.2.4. Sous Annexe 4

6.2.5. Sous Annexe 5

6.3. Annexe 3

6.4. Annexe 4