**Notion de rupture de séquences.**

L’exécution des opérations élémentaires (ou ACTIONS 2a et 2b) n’est pas systématique . Elle dépend d’une « condition ».

La lecture des ruptures de séquences s’effectue en fonction des éléments transmis à « titre de condition »

**CATEGORIES :**

Séquences

Ruptures de séquences :

Rupture conditionnelle ( et non répétitive)

SI

CAS

Rupture répétitive :

POUR

TANTQUE

REPETER

Ruptures conditionnelles :

SI..ALORS..FINSI

SI..ALORS..SINON..FINSI

CAS..PARMI..

CAS..PARMI..PARDEFAUT

Exemple :

SI<condition>ALORS

<instructions>

[SINON

<instructions>

]

FINSI

Note : Les [] caractérisent des choses non obligatoires.

Ecrire un algo qui donne le maximum entre deux nombres saisis par l’utilisateur

ALGORITHME :

//BUT

//ENTREE

//SORTIE

VAR : n.nombre1,n.nombre2 :ENTIER

DEBUT

ECRIRE « programme max entre 2 entiers »

ECRIRE «  Veuillez saisir le 1er nombre »

LIRE n.nombre1

ECRIRE «  Veuillez saisir le 2eme nombre »

LIRE n.nombre2

SI(n.nombre1>n.nombre2) ALORS

ECRIRE « nombre 1 est plus > » + n.nombre1

SINON

ECRIRE «  nombre 2 est plus > » +n.nombre2

FINSI

FIN

ALGORITHME : max\_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

ECRIRE « veuillez entrer deux nombres »

LIRE a , b

SI (a>=b)

ALORS

Max <=a

DEBUT

ECRIRE «  Veuillez entrer deux nombres »

LIRE a ,b

SI (a>=b)

ALORS

ECRIRE « le maximum entre « +a+ » et « +b+ » est : »  +a

SINON

ECRIRE « Le maximum entre « +a+ » et « +b+ » est : » +b

FINSI

FIN

Ecrire un algo qui calcule la valeur absolue d’un entier.

ALGORITHME :valeur\_absolue

//BUT Calculer la valeur absolue d’un entier

//ENTREE Saisi d’un entier

//SORTIE Affichage de la valeur absolue d’un entier

VAR : x : ENTIER

DEBUT

LIRE x //affectation d’une valeur par l’utilisateur

SI (x<0) //si cette valeur est négative

ALORS //alors cette condition est vraie

X<=x\*(-1) // et la variable se trouve réaffectée

FINSI

ECRIRE «  La val.abs. du nombre saisi vaut : » +x

FIN

Ecrire un algo qui demande un nombre à l’utilisateur , et l’informe ensuite si ce nombre est positif ou négatif ( on laisse de côté le cas où le nombre vaut zéro)

ALGO : exo-1

VAR : n :REEL

DEBUT :

ECRIRE « Entrez un nombre »

LIRE n

SI (n>0)

ALORS //alors cette condition est vraie

ECRIRE « Ce nombre est positif »

SINON

ECRIRE « Ce nombre est négatif »

FINSI

FIN

Ecrire un algo qui demande deux nombres à l’utilisateur et l’informe ensuite si leur produit est négatif ou positif ( on laisse de côté le cas où le produit est nul).

ALGO :positif\_negatif

//BUT Pouvoir déterminer si le produit de deux nombres est positif ou négatif

//ENTREE Saisi de deux nombres

//SORTIE Résultat du produit de deux nombres

VAR : nb1,nb2 :REEL

produit : CHAINE

ECRIRE « Veuillez saisir deux nombres »

LIRE nb1,nb2

SI nb1\*nb2>0 ALORS

produit<= Positif

SINON

produit<=Négatif

ECRIRE « Le produit de « nb1 » et « nb2 » est : »+produit

FIN

Ecrire un algo qui demande deux nombres à l’utilisateur et l’informe ensuite si leur produit est négatif ou positif ( on laisse de côté le cas où le produit est nul). Attention toutefois : on ne doit pas calculer le produit des deux nombres.

ALGO :positif\_negatif\_nocalcul

//BUT Pouvoir déterminer si le produit de deux nombres est positif ou négatif

//ENTREE Saisi de deux nombres

//SORTIE Résultat du produit de deux nombres

VAR : nb1,nb2 :REEL

produit : CHAINE

ECRIRE « Veuillez saisir deux nombres »

LIRE nb1,nb2

SI ( (nb1>0 ET nb2 >0) OU (nb1<0 ET nb2 <0) )

ALORS

ECRIRE «  le produit est positif »

SINON

ECRIRE « Le produit est négatif »

FINSI

FIN

Ecrire un algo qui demande trois noms à l’utilisateur et l’informe ensuite s’ils sont rangés ou non dans l’ordre alphabétique.

ALGORITHME : trois\_noms

//BUT : Informer l’utilisateur si les 3 noms sont classés par ordre alphabétique

//ENTREE : Saisi des 3 noms

//SORTIE : Informer si ils sont rangés

VAR : m1,m2,m3 :CHAINE

DEBUT

ECRIRE « Entrez trois noms »

LIRE m1,m2,m3

SI ((m2<=m1) et (m2<=m3))

ALORS

ECRIRE « Ces noms sont classés »

SINON

ECRIRE « Ces noms ne sont pas classés »

FINSI

FIN

Variante :

VAR : n :REEL

DEBUT

ECRIRE « Entrez un nombre »

LIRE n

SI (n>0) ALORS

ECRIRE « Ce nombre est positif »

SINON

SI(n<0) ALORS

ECRIRE « Ce nombre est négatif »

SINON

ECRIRE «  Ce nombre est nul »

FINSI

FINSI

FIN

Ecrire un algo qui demande le numéro d’un mois de l’année et affiche le nom du mois .

ALGORITHME : mois

//BUT : Afficher le noms du mois selon un numéro

//ENTREE : Saisi du numéro d’un mois

//SORTIE : Affichage du nom du mois

VAR : mois :ENTIER

Mois\_chaine : CHAINE

DEBUT

LIRE mois

SI (mois=1)

ALORS

Mois\_chaine <- « Janvier »

FINSI

SI (mois =2)

ALORS

Mois\_chaine <-« Février »

FINSI

Etc…

ECRIRE « Le mois est : » + mois\_chaine

FIN

Variante , plus optimisée :

DEBUT

LIRE mois

SI ( mois=1) ALORS

Mois\_chaine<-« Janvier »

SINON SI (mois=2)

ALORS

Mois\_chaine<-« Février »

SINON SI(mois=3)

ALORS mois\_chaine <-« Mars »

SINON

. . .

FINSI FINSI FINSI

FINSI

ECRIRE «  Le mois est : » + mois\_chaine

FIN

**CAS PARMI :**

CAS<variable>PARMI

{CAS{value[,]} :<instructions>}

[PARDEFAUT :<instructions>]

FINCASPARMI

Ecrire un algo qui demande le numéro d’un mois de l’année et qui affiche le mois

ALGO : nom\_mois2

//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 :

CAS 1 : ECRIRE « Janvier »

CAS 2 : ECRIRE « Février »

. . .

CAS 12 : ECRIRE «  Décembre »

PARDEFAUT : ECRIRE « Erreur de saisie ! »

FINCASPARMI

FIN

**Ruptures de répétition :**

POUR..FAIRE..FINPOUR

TANTQUE..FAIRE..FINTANTQUE

REPETER...JUSQU’A..

REPETER..TANTQUE..

**POUR :**

POUR<variable entière> DE <variable ou constante entière> A < variable ou constante entière> [PAS <variable ou constante entière>]

FAIRE

[<instructions>]

FINPOUR

Exemple :

ALGO : compteur\_rebours1

//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,compteur : ENTIER

DEBUT

LIRE nombre

POUR compteur DE nombre A 1 pas -1

FAIRE

ECRIRE « Valeur du compteur à rebours : » + compteur

FINPOUR

FIN

**TANTQUE :**

TANTQUE <expression booléenne>FAIRE

{<instructions>}

FINTANTQUE

Exemple :

ALGO :compteur\_rebours3

//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)

Nombre <- nombre -1

FINTANTQUE

FIN

Instruction répétitive « Répéter »

ALGO : compteur\_rebours2

//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)

Nombre <- nombre -1

JUSQU’A (nombre<0)

FIN

**REPETER..TANTQUE..**

ALGO : compteur\_rebours3

//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)

Nombre <- nombre -1

TANTQUE (nombre >0)

FIN