**Algorithmie Module 1 parti 2**

**Rappel notion de variable :**

* Les entiers
* Les réels
* Caractères
* Chaines de caractères
* Booléens

Les entiers :

* Entier
  + Nombres entiers
  + Négatifs
  + Positifs
  + Nuls
* Exemples :
  + 33
  + -13

[Pas de limite de taille]

Un entier signé est un entier avec un signe (+/-)

Un entier signé est un entier tjr positif

|  |  |
| --- | --- |
| Opération possible | Opération associés, symboles, mot clés correspondant |
| Addition | + |
| Soustraction | - |
| Multiplication | \* |
| Division | DIV (Pour l’entier) |
| Exposant | ^ |
| Modulo | MOD (pour le reste de la div) |
| Comparaison | </<=/>=/>/=/<> |

Algo : reel2entier

//but : cet algo renvoie l partie entière d’un réel

//entrée : un réel saisi par l’utilisateur

//sortis : l’entier

VAE : reel2ent : entier

Nb1 : réel

Début

Lire nb1

Reell2ent←

**Réel :**

* Nombres entiers
* Fractionnés
* Négatif
* Positifs
* Nuls

Exemple :

* 3.26
* -10.5

Pas de Limite de taille

|  |  |
| --- | --- |
| Adition | + |
| Soustraction | - |
| Multiplication | \* |
| Division | / |
| Exposant | ^ |
| Comparaison | </<=/>/>=/=/<> |

**Caractère :**

* Caractère unique :

Exemple :

‘a’

|  |  |
| --- | --- |
| Opération possible | Opérateurs associés, symbole, mot clés correspondant |
| Comparaison | </<=/>/>=/=/<> |
| Extraction | CAR1←EXTRACTION (chaine, position) |

**CHAINE :**

* Caractère unique
* Suite de caractère

Exemple :

« Bonjours »

« 64 »

« Super mario world »

[Référence : table des codes ASCII, Unicode]

Apostrophe double

|  |  |
| --- | --- |
| Opération possible | Opérateur symbole mots clés |
| Comparaison | </<=/>/>=/=/<> |
| Extraction | CH1←Extracion (CH2, position, longueur) |
| Longueur | Ent←LONGUEUR (ch.) |
| Concaténation | Ch←concatener (ch1, ch2) ou ch←ch1+ch2 |

BOOLEEN:

* Vrai
* Faux

|  |  |
| --- | --- |
| Opération possible | Opérateur symbole mots clés |
| Comparaison | = <> |
| Négation | Non |
| Conjonction | Et |
| Disjonction | Ou |

**Structure de données autres structures de données :**

* Structures linéaires :
  + Tableaux
  + Piles
  + Files
  + Listes
* Structures non linéaires :
  + Enregistrements
  + Graphes
  + Arbres

EXERCICE :

Ecrire un algo qui demande à un utilisateur son nom et prénom et affiche ses initiales

Algo : initiales

//BUT : extrait les initiales d’un perso

//ENTREE : le nom et le prénom saisis

//SORTIS : Nom, prénom, initiales

VAR : nom, prénom, initiales : CHAINE

DEBUT

ECRIRE « Veuillez entrer votre nom »

LIRE nom

ECRIRE « Veuillez entrer votre prénom »

LIRE prénom

Initiale EXTRACTION (nom,1,1) +EXTRACTION (prenom,1,1)

ECRIRE INITIALES

FIN

Seconde solution

Algo : initiales

//BUT : extrait les initiales d’un perso

//ENTREE : le nom et le prénom saisis

//SORTIS : initiales

VAR : nom, prénom, initiales : CHAINE

DEBUT

ECRIRE « Veuillez entrer votre nom »

LIRE nom

ECRIRE « Veuillez entrer votre prénom »

LIRE prénom

Initiale EXTRACTION (nom+’ ’,1,1) +EXTRACTION (prenom+’ ‘,1,1)

ECRIRE INITIALES

FIN

Ecrire un algo qui demande à un utilisateur son nom et son prénom et qui affiche son adresse mail Ludus :

JEAN/Martin/[j.martin@ludus-academie.com](mailto:j.martin@ludus-academie.com)

Algorithme : Adresse mail

//BUT : Afficher l’adresse mail ludus

//ENTREE : Saisir Prénom et nom

//SORTIS : Adresse mail ludus

VAR : NOM, PRENOM, ADRESSE : CHAINE

DEBUT

ECRIRE : « VEUILLEZ SAISIR VOTRE PRENOM

LIRE PRENOM

ECRIRE : « VEUILLEZ SAIS VOTRE NOM »

LIRE : NOM

ADRESSE←EXTRACTION (PRENOM+’ ’,1,1) +’.’+non+ « @ludus-academie.com »

ECRIRE ADRESSE

FIN

Ecrire un algo qui demande un mot de 3 lettres a un utilisateur et affiche son miroir.

Ex :abc devient cba

Algorithme : miroir

//BUT : Afficher le miroir d’un mot

//ENTREE : Saisir un mot de 3 lettres

//SORTIS : mot de 3 lettres inverser

VAR : mot, motmiroir : chaine

DEBUT :

ECRIRE : « SAISIR UN MOT »

Lire mot

Mot ←EXTRACTION (mot+ « \_\_\_ »,1,3)

Motmiroir←EXTRACTION (mot,3,1) +EXTRACTION (mot,2,1)+ EXTRACTION(mot1,1)

Ecrire MOTMIROIRE

FIN

Ecrire un algo qui demande un mot de 5 lettres et qui affiche si c’est un palindrome ou non.

Ex : « radar » affiche vrai/ « Mario » affiche faux

Algo : palindrome

//But : dire si le mot est un palindrome ou non

//entrée : saisis d’un mot de 5 lettres

//sortis : Vrai si palindrome ou faux

VAR : MOT, motmiroir : CHAINE

Début

ECRIRE « Veuillez entrer votre mot »

LIRE mot

mot ←EXTRACTION(mot+ « \_\_\_\_\_ »,1,5)

motmirroir←EXTRACITON(mot,5,1)+EXTRACTION(mot,4,1)+ EXTRACTION(mot,3,1)+EXTRACTION(mot,2,1)+EXTRACTION(mot,1,1)

ECRIRE motmirroir=mot

FIN

Seconde solution

Algo : palindrome

//But : dire si le mot est un palindrome ou non

//entrée : saisis d’un mot de 5 lettres

//sortis : Vrai si palindrome ou faux

VAR : MOT, motmiroir : CHAINE

motegal :BOOLEEN

Début

ECRIRE « Veuillez entrer votre mot »

LIRE mot

mot ←EXTRACTION(mot+ « \_\_\_\_\_ »,1,5)

motmirroir←EXTRACITON(mot,5,1)+EXTRACTION(mot,4,1)+ EXTRACTION(mot,3,1)+EXTRACTION(mot,2,1)+EXTRACTION(mot,1,1)

motegal←(motmirroire=mot)

Ecrire motegale

FIN

3ème Version

Algo : palindrome

//But : dire si le mot est un palindrome ou non

//entrée : saisis d’un mot de 5 lettres

//sortis : Vrai si palindrome ou faux

VAR : MOT : CHAINE

Estpalin : BOOLEEN

Début

ECRIRE « Veuillez entrer votre mot »

LIRE mot

mot ←EXTRACTION (mot+ « \_\_\_\_\_ »,1,5)

estpalin←EXTRACTION(mot,5,1)= EXTRACTION(mot,1,1) et EXTRACTION(mot,4,1)=EXTRACTION(mot,2,1)

ECRIRE estpalin

FIN

4ème solution :

Algo : palindrome

//But : dire si le mot est un palindrome ou non

//entrée : saisis d’un mot de 5 lettres

//sortis : Vrai si palindrome ou faux

VAR : MOT : CHAINE

Début

ECRIRE « Veuillez entrer votre mot »

LIRE mot

mot←EXTRACITON(mot+ « \_\_\_\_\_ »,5,1)= EXTRACTION (mot+ « \_ »,1,1)et EXTRACTION( mot+ « \_\_\_\_ »,4,1)= EXTRACTION (mot+ « \_\_ »,2,1)

FIN

**Notion de rupture des 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 transmit à titre de condition

**Catégorie :**

* Séquences
  + Ruptures des 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

**Si :**

SI <condition> ALORS

<instruction>

[SINON

<instruction>

]

FINSI

Exercice :

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

**Solution 1**

ALGO : …

//BUT : Donner le maximum entre deux nombre saisis par l’utilisateur

//ENTREES : Saisis de deux nombres

//SORTIES : Maximum entre deux nombres

VAR : nb1, nb2 : ENTIER

DEBUT

ECRIRE « Programme max entre 2 nombre »

ECRIRE « Veuillez saisir le 1er nombre »

LIRE nb1

ECRIRE « Veuillez saisir le 2ème nombre »

LIRE nb2

SI (nb1>nb2) ALORS

ECRIRE « Nombre 1 est plus > » +nb1

SINON

ECRIRE « Nombre 2 est plus > » +nb2

FINSI

FIN

**SOLUTION 2 :**

Algo :max\_deux\_nombres

//but : cherche lla valeur max.parmi 2 valuer saisies

//ENTREE : deux réels saisis par l’utilisateur

//SORTIE : le maximum des deux valeurs

VAR : a, b, max :REEL

DEBUT

ECRIRE « veuillez entrer deux nombre »

LIRE a, b

Max←b

SI (a>=b)

ALORS

Max←a

FINSI

ECRIRE « LE Maximum entre » +a+ « et » +b+ « est : » +max

FIN

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

ALGO : valeur absolue

//BUT : obtenir la valeur absolue d’un nombre

//ENTREE : saisir un nombre

//SORTIS : donner la valeur absolue du nombre

VAR :x : ENTIER

DEBUT

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

SI « x<0) //Si cette valeur est négative

ALORS //alors cette condiiton est vraie

x←x\*-1 //et l variable se trouve réaffecter

FINSI

ECRIRE « la valeur absolue du nombre saisis vaut : » +x

FIN

Ecrire un algo qui indique à l’utilisateur si le nombre saisis est positif ou négatif (on laisse de côté le cas vaut 0)

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 2 nombres a l’utilisateur et l’informe ensuite si leur produit est négatif ou positif (on laisse de coté les cas ou le prds est nul)

ALGO : + ou –

//BUT Savoir si le produit des deux nb saisis est positif ou négatif

//ENTREE saisis de 2 nbr

//SORTIS : dire si le résultat est positif ou négatif

VAR : nb1, nb2, produit : REEL

DEBUT

ECRIRE « Saisir 2 nombre »

LIRE nb1, nb2

produit←nb1\*nb2

SI (produit>0) ALORS

ECRIRE « LE résultat est » +produit+ « et est donc positif »

SINON

ECRIRE « Le résultat est » +produit+ « et est donc négatif »

Ecrire un algo qui demande 2 nombres à l’utilisateur et l’informe ensuite si leur produit est négatif ou positif (on laisse de côté les cas où le prds est nul) Attention toutefois : on ne doit pas calculer le produit des deux nombres

ALGO : + ou –

//BUT Savoir si le produit des deux nb saisis est positif ou négatif

//ENTREE saisis de 2 nbr

//SORTIS : dire si le résultat est positif ou négatif

VAR : nb1, nb2, : REEL

DEBUT

ECRIRE « Saisir 2 nombre »

LIRE nb1, nb2

SI (nb1>0 et nb2>0) OU (nb1<0 et nb<0) ALORS

ECRIRE « le produit est positif »

SINON

ECRIRE « Le résultat 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.

ALGO : Alpha

//BUT classer 3 noms par ordre alphabétique

//ENTREE : SAISIR 3 noms

//SORTIS : Les nom sont classé par ordre alphabétique

VAR : mot1, mot2, mot3 : CHAINE

DEBUT

ECRIRE « Entrez trois noms »

LIRE mot1, mot2, mot3

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

ALORS

ECRIRE « ces noms sont classés »

SINON

ECRIRE « Ces noms ne sont pas classés »

FINSI

FIN

Ecrire un algo qui demande 2 nombres a l’utilisateur et l’informe ensuite si leur produit est négatif ou positif (on inclut les cas où le prds est nul) Attention toutefois : on ne doit pas calculer le produit des deux nombres

ALGO :exo-1

Var m,n : ENTIER

DEBUT

ECRIRE « ENTREZ deux nombre »

LIRE m,n

SI ((m>0 et n>0) ou (m<0 et n<0)) ALORS

Ecrire « le produit est positif »

Si(m=0 ou n=0) ALORS

ECRIRE «  le produit est nul »

SINON

ECRIRE « le produit est négatif »

FINSI

FINSI

FIN

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

ALGO : mois

//BUT : donner le nom d’un mois par rapport a son chiffre

//ENTREE : Saisis d’un nombre entre 1 et 12 inclus

//SORTIS : NOM du mois de l’année

VAR : mois :ENTIER

mois\_chaine :CHAINE

DEBUT

LIRE mois

SI (mois =1)

ALORS

mois\_chaine← « janvier »

FINSI

ETC…

VERSIon la plus opti

VAR : mois :ENTIER

mois\_chaine :CHAINE

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

etc…………………………………….

FINSI FINSI FINSI

**CAS PARMI :**

CAS<variable>PARMI

{CAS {value[,]} :<instructions>}

[PARDEFAUT : <instructions>]

FINCASPARMI

ALGO :nom\_mois2

//BUT : affiche le nom du mois en fonction du numéro

//ENTREE :un entier saisi par l’utilisateur

//SORTIS :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 »

Etc…

CAS 12 : ECRIRE « Décembre »

PARDEFAUT : ECRIRE « Erreur de saisie ! »

FINCASPARMI

FIN

Rupture 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

[<instruction>]

FINPOUR

ALGO :compteur\_rebours1

//BUT :affiche la valeur d’un compteur à rebours

//ENTREE : un entier saisis par l’utilisateur

//SORTIE :LA valeur du compteur à rebours

VAR : nombre,compteur  :ENTIER

DEBUT

LIRE nombre

POUR compteur DE nombre A 1pas -1

FAIRE

ECRIRE « Valeur du compteur a rebours : »+compteur

FINPOUR

FIN

ALGO :compteur\_rebours2

//BUT :affiche la valeur d’un compteur à rebours

//ENTREE : un entier saisis par l’utilisateur

//SORTIE :LA valeur du compteur à rebours

VAR : nombre,compteur  :ENTIER

DEBUT

LIRE nombre

POUR compteur DE 0 A nombre -1

FAIRE

ECRIRE « Valeur du compteurà rebours : » +nombre-compteur

FINPOUR

FIN

TANTQUE< exp booléenne>FAIRE

{<instruction>}

FINTANTQUE

ALGO :compteur\_rebours1

//BUT :affiche la valeur d’un compteur à rebours

//ENTREE : un entier saisis 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 »**

**REPETER…JUSQU’A…**

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 : ENTIER

DEBUT

LIRE (nombre)

REPETER

ECRIRE (« Valeur du compteur à rebours » +nombre)

Nombre <-nombre – 1

JUSQU’A (nombre<0)

FIN

**Instruction répétitive : 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