PARTIE 3 – LES TESTS

**ENONCE DES EXERCICES**

# Exercice 3.1

Ecrire un algorithme 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).

A EST\_DU\_TYPE NOMBRE

LIRE A

SI (A > 0) ALORS

DEBUT\_SI

AFFICHER A

AFFICHER " est positif"

FIN\_SI

SI (A < 0) ALORS

DEBUT\_SI

AFFICHER A

AFFICHER " est négatif"

FIN\_SI

# Exercice 3.2

Ecrire un algorithme 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.

A EST\_DU\_TYPE NOMBRE

B EST\_DU\_TYPE NOMBRE

LIRE A

LIRE B

AFFICHER "le produit de "

AFFICHER A

AFFICHER " et "

AFFICHER B

SI (A>0) ALORS

DEBUT\_SI

SI (B>0) ALORS

DEBUT\_SI

AFFICHER " est positif"

FIN\_SI

SINON

DEBUT\_SINON

AFFICHER " est négatif"

FIN\_SINON

FIN\_SI

SINON

DEBUT\_SINON

SI (B<0) ALORS

DEBUT\_SI

AFFICHER " est positif"

FIN\_SI

SINON

DEBUT\_SINON

AFFICHER " est négatif"

FIN\_SINON

FIN\_SINON

FIN\_ALGORITHME

# Exercice 3.3

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

A EST\_DU\_TYPE CHAINE

B EST\_DU\_TYPE CHAINE

C EST\_DU\_TYPE CHAINE

LIRE A

LIRE B

LIRE C

AFFICHER A

AFFICHER ", "

AFFICHER B

AFFICHER " et "

AFFICHER C

SI (A < B) ALORS

DEBUT\_SI

SI (B < C) ALORS

DEBUT\_SI

AFFICHER " sont rangés dans l’ordre alphabétique."

FIN\_SI

SINON

DEBUT\_SINON

AFFICHER " ne sont pas rangés dans l’ordre alphabétique."

FIN\_SINON

FIN\_SI

SINON

DEBUT\_SINON

AFFICHER " ne sont pas rangés dans l’ordre alphabétique."

FIN\_SINON

FIN\_ALGORITHME

# Exercice 3.4

Ecrire un algorithme qui demande un nombre à l’utilisateur, et l’informe ensuite si ce nombre est positif ou négatif (on inclut cette fois le traitement du cas où le nombre vaut zéro).

A EST\_DU\_TYPE NOMBRE

LIRE A

SI (A > 0) ALORS

DEBUT\_SI

AFFICHER A

AFFICHER " est positif"

FIN\_SI

SINON

DEBUT\_SINON

SI (A < 0) ALORS

DEBUT\_SI

AFFICHER A

AFFICHER " est négatif"

FIN\_SI

SINON

DEBUT\_SINON

AFFICHER A

AFFICHER " est nul"

FIN\_SINON

FIN\_SINON

# Exercice 3.5

Ecrire un algorithme qui demande deux nombres à l’utilisateur et l’informe ensuite si le produit est négatif ou positif (on inclut cette fois le traitement du cas où le produit peut-être nul). Attention toutefois, on ne doit pas calculer le produit !

A EST\_DU\_TYPE NOMBRE

B EST\_DU\_TYPE NOMBRE

LIRE A

LIRE B

AFFICHER "le produit de "

AFFICHER A

AFFICHER " et "

AFFICHER B

SI (A != 0 ET B != 0) ALORS

DEBUT\_SI

SI (A>0) ALORS

DEBUT\_SI

SI (B>0) ALORS

DEBUT\_SI

AFFICHER " est positif"

FIN\_SI

SINON

DEBUT\_SINON

AFFICHER " est négatif"

FIN\_SINON

FIN\_SI

SINON

DEBUT\_SINON

SI (B<0) ALORS

DEBUT\_SI

AFFICHER " est positif"

FIN\_SI

SINON

DEBUT\_SINON

AFFICHER " est négatif"

FIN\_SINON

FIN\_SINON

FIN\_SI

SINON

DEBUT\_SINON

AFFICHER " est nul"

FIN\_SINON

FIN\_ALGORITHME

# Exercice 3.6

Ecrire un algorithme qui demande l’âge d’un enfant à l’utilisateur. Ensuite, il l’informe de sa catégorie :

* "Poussin" de 6 à 7 ans
* "Pupille" de 8 à 9 ans
* "Minime" de 10 à 11 ans
* "Cadet" après 12 ans

A EST\_DU\_TYPE NOMBRE

LIRE A

AFFICHER "Catégorie: "

SI ( A >= 12 ) ALORS

DEBUT\_SI

AFFICHER "cadet"

FIN\_SI

SINON

DEBUT\_SINON

SI (A >= 10) ALORS

DEBUT\_SI

AFFICHER "minimes"

FIN\_SI

SINON

DEBUT\_SINON

SI (A >= 8) ALORS

DEBUT\_SI

AFFICHER "pupilles"

FIN\_SI

SINON

DEBUT\_SINON

SI (A >= 6) ALORS

DEBUT\_SI

AFFICHER "poussin"

FIN\_SI

SINON

DEBUT\_SINON

AFFICHER "aucune (trop jeune)"

FIN\_SINON

FIN\_SINON

FIN\_SINON

FIN\_SINON

Peut-on concevoir plusieurs algorithmes équivalents menant à ce résultat ?

oui