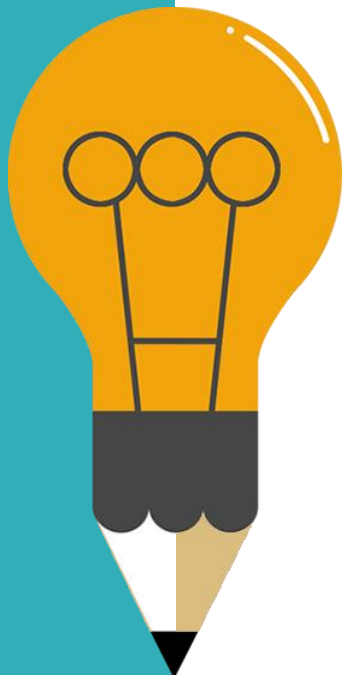


EXERCICES EN ALGORITHMIQUE

Introduction
Aux algorithmes de base

SOMMAIRE



01

LES VARIABLES

UTILISATION DES VARIABLES

02

LIRE & ECRIRE

UTILISATION DES FONCTION DE **LECTURE** ET **AFFICHAGE**

03

LES STRUCTURES DE TEST

UTILISATION DES **SI** ET DE **SELON...**

04

LES STRUCTURES ITERATIVES

UTILISATION DE : **TANT QUE** – **REPETER...JUSQU'A** - **POUR**



Les variables

Utilisation des operateurs arithmétiques

EXERCICE 1

LES VARIABLES & LES OPérateURS



Calcul simple:

Écrivez un algorithme qui définit deux variables a et b avec des valeurs numériques, puis calcule et stocke dans des variables :

- *La somme de a et b. *La différence entre a et b.
- *Le produit de a et b. *Le quotient de a par b.



SOLUTION

.....

SOLUTION : 1



Algorithme CalculSimple

Variables

a, b, somme, difference, produit, quotient : Réel

Début

a \leftarrow 10 ; **b** \leftarrow 5 ;

somme \leftarrow a + b; **difference** \leftarrow a - b; **produit** \leftarrow a * b; **quotient** \leftarrow a / b;

Fin



EXERCICE 2

LES VARIABLES & LES OPPERATEURS



**Écrivez un algorithme qui définit trois variables note1 , note2 et note3 avec des valeurs numériques.
puis calcule et stocke dans une variable moyenne la
moyenne de ces trois notes.**



SOLUTION

.....

SOLUTION : 2



Algorithme CalculMoyenne

Variables

note1, note2, note3, moyenne : Réel;

Début

note1 \leftarrow 14; note2 \leftarrow 12; note3 \leftarrow 10;

moyenne \leftarrow (note1 + note2 + note3) / 3;

Fin



EXERCICE 3

LES VARIABLES & LES OPérateURS



Écrivez un algorithme qui définit **deux variables** longueur et largeur avec des valeurs numériques, puis **calcule** et **stocke** dans des variables :
aire : l'aire du rectangle.
perimetre : le périmètre du rectangle.



SOLUTION

.....

SOLUTION : 3



Algorithme AirePerimetre

Variables

longueur, largeur, aire, perimetre : Réel

Début

longueur \leftarrow 10; largeur \leftarrow 5;

aire \leftarrow longueur * largeur;

perimetre \leftarrow 2 * (longueur + largeur);

Fin





Lire & Ecrire

Utilisation des fonctions : lire() & Ecrire()

EXERCICE 1

Les fonction de lecture et d'affichage



Écrire un algorithme qui demande à l'utilisateur de saisir deux nombres, calcule leur somme et affiche le résultat.



SOLUTION

.....

SOLUTION : 1



Algorithme SommeDeuxNombres

Variable **nombre1, nombre2, somme** : Réel;

DEBUT

```
Écrire("Entrez le premier nombre : "); Lire(nombre1);  
Écrire("Entrez le deuxième nombre : "); Lire(nombre2);  
somme ← nombre1 + nombre2;  
Écrire("La somme des deux nombres est : ", somme);
```

FIN



EXERCICE 2

Les fonction de lecture et d'affichage



Écrire un algorithme qui demande à l'utilisateur de saisir trois notes, calcule leur moyenne et affiche le résultat.



SOLUTION

.....

SOLUTION : 2



Algorithme CalculMoyenne

Variable **note1, note2, note3, moyenne** : Réel

DEBUT

Écrire("Entrez la première note : "); **Lire**(note1);

Écrire("Entrez la deuxième note : "); **Lire**(note2);

Écrire("Entrez la troisième note : "); **Lire**(note3);

moyenne \leftarrow (note1 + note2 + note3) / 3 ;

Écrire("La moyenne des trois notes est : ", moyenne);

FIN



EXERCICE 3

Les fonction de lecture et d'affichage



Écrire un algorithme qui demande à l'utilisateur de saisir une température en degrés Celsius, la convertit en degrés Fahrenheit et affiche le résultat. La formule de conversion est :

$$\text{Fahrenheit} = (\text{Celsius} \times 9/5) + 32$$



SOLUTION

.....

SOLUTION : 3



Algorithme ConversionTemperature

DEBUT

VARIABLE **celsius**, **fahrenheit** : Réel ;

Écrire("Entrez la température en degrés Celsius : ");

Lire(**celsius**);

fahrenheit \leftarrow (**celsius** * 9/5) + 32;

Écrire("La température en degrés Fahrenheit est : ", **fahrenheit**);

FIN





Les structures

Utilisation des : si , si(....)alors(...)

EXERCICE 1

Les structures de test



**Écrivez un algorithme qui demande à l'utilisateur de saisir un nombre entier.
Si le nombre est positif, affichez "Le nombre est positif".
Si le nombre est négatif, affichez "Le nombre est négatif".
Si le nombre est nul, affichez "Le nombre est nul".**

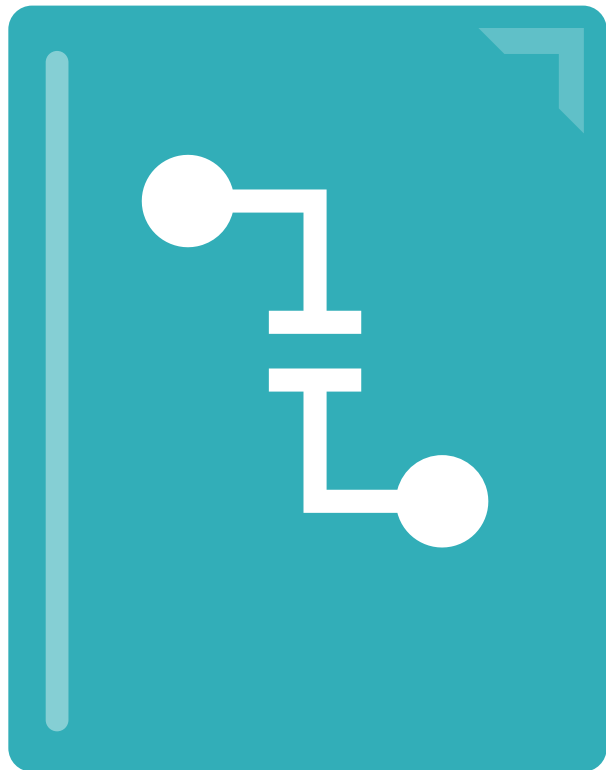


SOLUTION

.....

SOLUTION : 1

“



Algorithme **TestSi**

Variables: **nombre** : entier;

Début

Afficher ("Entrez un nombre entier :");

Lire (nombre);

Si (nombre > 0) **Alors**

Afficher ("Le nombre est positif");

Sinon Si (nombre < 0) **Alors**

Afficher ("Le nombre est négatif");

Sinon

Afficher("Le nombre est nul ");

Fin Si

Fin

”

EXERCICE 2

Les structures de test



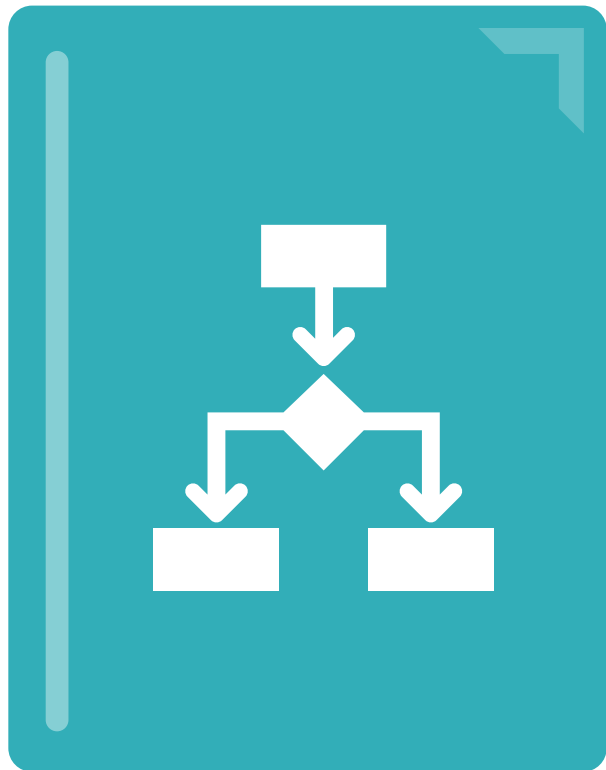
Écrivez un algorithme qui demande à l'utilisateur de saisir trois nombres entiers. Ensuite, déterminez et affichez le plus grand des trois nombres. Utilisez des structures de test si pour comparer les nombres.



SOLUTION

.....

SOLUTION : 2



Algorithme PlusGrandNombre

variable : a, b, c, max : entier;

Début

Afficher("Entrez le premier nombre : "); Lire(a);

Afficher("Entrez le deuxième nombre : "); Lire(b);

Afficher("Entrez le troisième nombre : "); Lire(c);

Si (a > b) Alors

Si (a > c) Alors

max ← a;

Sinon

max ← c;

Fin Si

Sinon

Si (b > c) Alors

max ← b;

Sinon

max ← c;

Fin Si

Fin Si

Afficher("Le plus grand nombre est : ", max);

Fin



EXERCICE 3

Les structures de test



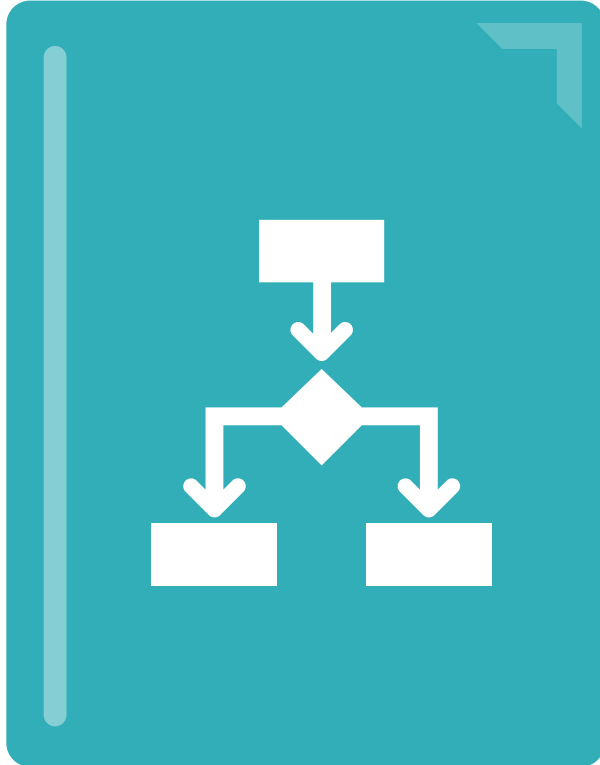
Écrivez un algorithme qui demande à l'utilisateur de saisir un jour de la semaine (sous forme de nombre entre 1 et 7). Affichez le nom du jour correspondant en utilisant une structure selon (switch). Par exemple, si l'utilisateur saisit 1, affichez "Lundi".



SOLUTION

.....

SOLUTION : 3



Algorithme TestSelon

variable **jour** : entier ;

Début

Afficher ("Entrez un nombre entre 1 et 7 : ");

Lire(**jour**);

Selon **jour**

Cas 1 : **Afficher** ("Lundi");

Cas 2 : **Afficher** ("Mardi");

Cas 3 : **Afficher** ("Mercredi");

Cas 4 : **Afficher** ("Jeud");

Cas 5 : **Afficher** ("Vendredi");

Cas 6 : **Afficher** ("Samed");

Cas 7 : **Afficher** ("Dimanche");

Sinon : **Afficher** ("Jour invalide");

Fin Selon

Fin





Les structures

Utilisation des : tant que – répéter...jusqu'à-pour

EXERCICE 1

Tant que



Écrire un algorithme qui demande à l'utilisateur de saisir un nombre entier positif. Si l'utilisateur saisit un nombre négatif, l'algorithme doit redemander la saisie jusqu'à ce qu'un nombre positif soit entré.



SOLUTION

.....

SOLUTION : 1



Algorithme SaisiePositive

Variables

nombre : entier ;

Début

nombre \leftarrow -1 ;

Tant que (**nombre** < 0) **faire**

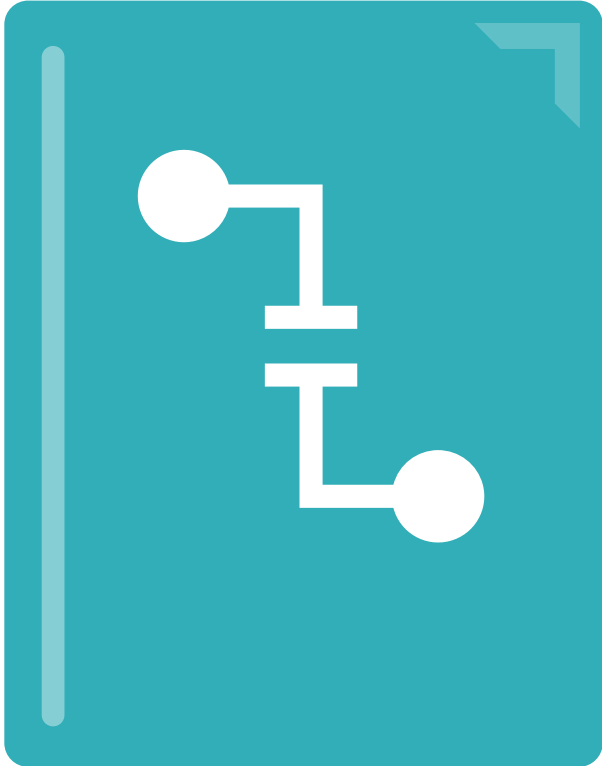
Afficher ("Entrez un nombre positif :");

Lire (**nombre**);

Fin Tant que

Afficher ("Vous avez saisi un nombre positif : ", **nombre**);

Fin



EXERCICE 2

Répéter jusqu'à



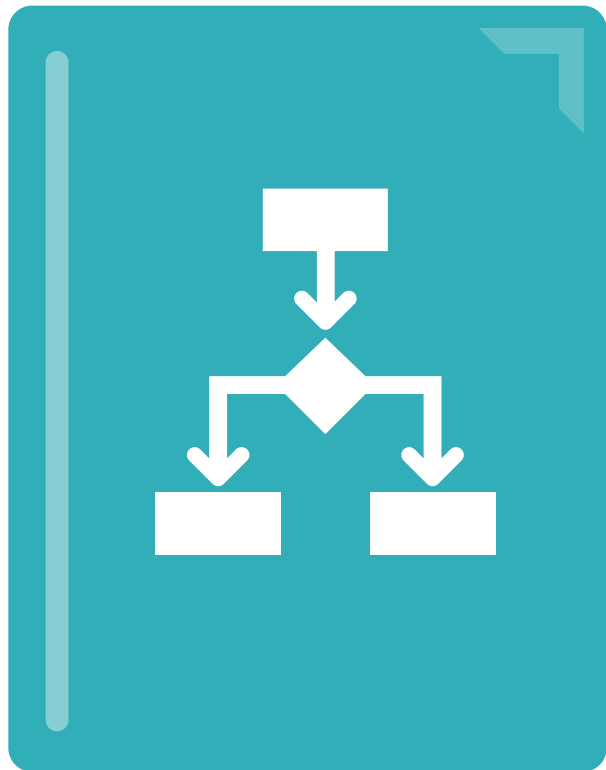
Écrire un algorithme qui demande à l'utilisateur de deviner un nombre secret entre 1 et 100. L'algorithme doit répéter la demande jusqu'à ce que l'utilisateur trouve le nombre secret.



SOLUTION

.....

SOLUTION : 2



Algorithme DevinerNombre

Variables

nombreSecret, tentative : entier

Début

nombreSecret \leftarrow 42 ;

Répéter

Afficher ("Devinez le nombre secret (entre 1 et 100) :");

Lire (**tentative**);

Jusqu'à (**tentative** = **nombreSecret**);

Afficher ("Bravo ! Vous avez trouvé le nombre secret");

Fin



EXERCICE 3

Pour(.....)



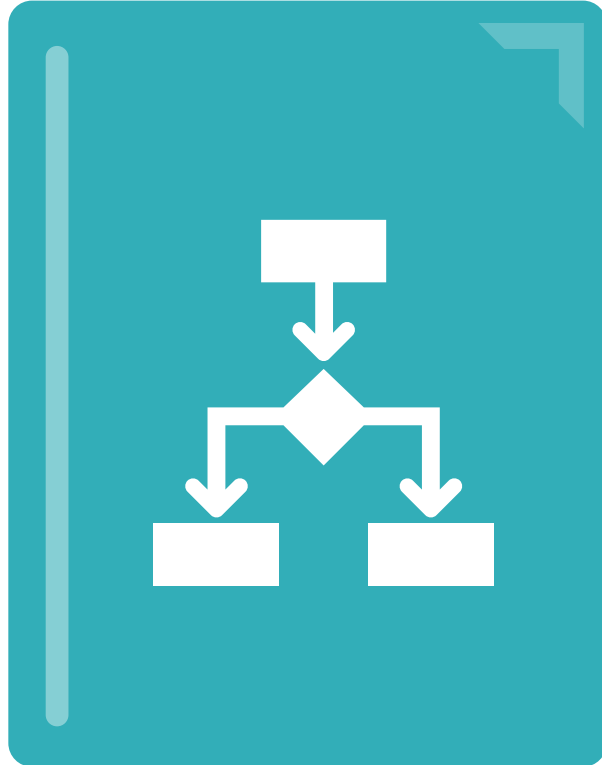
Écrire un algorithme qui calcule et affiche la somme des 10 premiers entiers naturels ($1 + 2 + 3 + \dots + 10$) en utilisant une boucle "pour".



SOLUTION

.....

SOLUTION : 3



Algorithme Somme10Entiers

Variables

somme, i : entier;

Début

somme \leftarrow 0;

Pour i de 1 à 10 faire

somme \leftarrow **somme** + i ;

Fin Pour

Afficher (" somme des 10 premiers entiers est : ", **somme**);

Fin

