**EXERCICE 1 & 3 :**

Algorithme Afficher\_le\_plus\_grand\_nombre ;

Var

m, n : réels ;

Début

Afficher ‟ Entrer deux nombres. ” ;

Saisir m, n ;

SI m > n alors :

Afficher ‟ Le plus grand nombre est ” m ;

SINON SI m < n alors :

Afficher ‟ Le plus grand nombre est ” n ;

SINON

Afficher ‟ Les nombres sont égaux. ” ;

FIN SI

Fin

**EXERCICE 2 :**

Algorithme Afficher\_message ;

Var

annee\_naissance : entiers ;

Cons

annee\_actu = 2022

Début

Afficher ‟ Entrer votre date de naissance. ” ;

Saisir annee\_naissance ;

age ← annee\_actu - annee\_naissance ;

SI age < 10 alors :

Afficher ‟ Vous êtes un enfant. ” ;

FIN SI

Fin

**EXERCICE 4 :**

Algorithme Afficher\_valeur\_absolu\_d\_un\_nombre ;

Var

m, n, o : réels ;

Début

Afficher ‟ Entrer deux nombres. ” ;

Saisir m, n ;

o ← m - n ;

SI o > 0 alors :

Afficher ‟ VA = ” o ;

SINON SI o = 0 alors :

Afficher ‟ VA = 0 ” ;

SINON

Afficher ‟ VA = ” -o ;

FIN SI

Fin

**EXERCICE 5 :**

Algorithme Afficher\_les\_actions ;

Var

feu\_tricolore : chaine de caractère ;

Début

Afficher ‟ Entrer une couleur de feu tricolore. ” ;

Saisir feu\_tricolore ;

SI feu\_tricolore = rouge alors :

Afficher ‟ STOP ” ;

SINON SI feu\_tricolore = orange alors :

Afficher ‟ Veuillez RALENTIR” ;

SINON SI feu\_tricolore = VERT alors :

Afficher ‟ PASSER” ;

SINON

Afficher ‟ ERRUR” ;

FIN SI

Fin

**EXERCICE 6 :**

Algorithme Résolution\_d\_équation ;

Var

a, b, x : réels ;

Début

Afficher ‟ Entrer deux nombres réels. ” ;

Saisir a, b ;

x ← -b / a ;

Afficher ‟ La solution à l’équation ax + b = 0 est ” x ;

Fin

**Exercice 7 :**

Algorithme Le\_plus\_grand\_nombre ;

Var

A, B, C, Max, égal 1, égal 2 : entiers ;

Début

Afficher ‟Entrer trois nombres.” ;

Saisir A, B, C ;

SI A ˃ B alors ;

Max ← A ;

SI A ˂ B alors ;

Max ← B ;

SINON

égal 1 ← A ;

Fin SI

SI C ˃ Max alors ;

Afficher ‟ Le plus grand nombre est ” C ;

SINON SI Max ˃ C alors ;

Afficher ‟ Le plus grand nombre est ” Max ;

SINON

égal 2 ← C ;

Fin SI

SI égal 1 ˃ égal 2  alors ;

Afficher ‟ Le plus grand nombre est ” égal 1 ;

SINON SI égal 1 ˂ égal 2  alors ;

Afficher ‟ Le plus grand nombre est ” égal 2 ;

SINON

Afficher ‟ Les nombres sont égaux ” ;

FIN SI

Fin

**EXERCICE 8 :**

Algorithme Afficher\_les\_actions ;

Var

feu\_tricolore : chaine de caractère ;

Début

Afficher ‟ Entrer une couleur de feu tricolore. ” ;

Saisir feu\_tricolore ;

Suivant les valeurs de feu tricolore faire :

‟ Rouge ”

Afficher ‟ STOP ” ;

‟ Orange ”

Afficher ‟RALENTIR” ;

‟ VERT ”

Afficher ‟ PASSER” ;

SINON

Afficher ‟ ERRUR” ;

FIN Suivant

Fin

**Exercice 9 :**

Algorithme Opération ;

Var

a, b, c : entiers ;

x : caractère ;

Début

Afficher ‟ Entrer un nombre.” ;

Saisir a ;

Afficher ‟ Entrer un nombre.” ;

Saisir b ;

Afficher ‟ Entrer un nombre.” ;

Saisir c ;

Afficher ‟ 1 – somme ” ;

‟ 2 – produit ” ;

‟ 3 – moyenne ” ;

Afficher ‟ Choisissez le numéro de l’opération à faire ” ;

Saisir x ;

Suivant les valeurs de x faire :

‟ 1 ” ;

Afficher ‟ La somme des nombres est : ” a + b + c ;

‟ 2 ” ;

Afficher ‟ Le produit des nombres est : ” a \* b \* c ;

‟ 3 ” ;

Afficher ‟ La moyenne des nombres est : ”  a + b + c / 3 ;

Fin Suivant

Fin

**Exercice 10 :**

Algorithme Somme\_des\_10\_premiers\_nombre ;

Var

n, somme : entiers ;

Début

n ← 1 ;

somme ← 0 ;

Tant que n < = 10 faire :

somme ← somme + 1 ;

n ← n + 1 ;

Fin Tant que

Afficher ‟ La somme des 10 premiers nombre est ” somme;

Fin

**Exercice 11 :**

Algorithme Afficher\_nombre\_de\_tentative\_ZZZZ ;

Var

nom, fois : chaîne de caractère ;

Début

fois ← 0 ;

Répéter

Afficher ‟ Entrer un nom.” ;

Saisir nom ;

fois ← fois+1 ;

Jusqu’à ce que nom = ‟ZZZZ”;

Fin Répéter

Afficher ‟ Vous avez fait”, fois, ‟ tentative.” ;

Fin

**Exercice 12 :**

Algorithme Afficher\_message ;

Var

reponse : chaîne de caractère ;

Début

Afficher ‟ Aimez-vous l’informatique ?” ;

Saisir reponse ;

reponse ← oui ;

Tant que n < > oui faire :

Afficher ‟ Veillez bien repondre par oui ou non.” ;

Fin Tant que

Fin

**Exercice 13 :**

Algorithme Afficher\_message ;

Var

reponse : chaîne de caractère ;

Début

Répeter

Afficher ‟ Aimez-vous l’informatique ?” ;

Saisir reponse ;

SI reponse < > ‟oui” ou reponse < > ‟non” alors :

Afficher ‟ Mauvaise reponse ” ;

Fin SI

Jusqu’à ce que reponse = ‟oui” ou reponse = ‟non”

Afficher ‟oui ou non.” ;

Fin Répeter

Fin

**Exercice 14 :**

Algorithme Somme\_des\_nombres\_supérieur\_ou\_égale\_à\_100 ;

Var

nombre, somme : entiers ;

Début

somme ← 0 ;

Répéter

somme ← somme + n ;

Afficher ‟ Entrer un nombre.” ;

Saisir nombre ;

Jusqu’à ce que somme > = 100 ;

Fin Répéter

Fin

**Exercice 15 :**

Algorithme Somme\_des\_10\_premiers\_nombre ;

Var

n, somme : entiers ;

Début

Afficher ‟ Entrer un nombre.” ;

Saisir nombre ;

n ← 1 ;

somme ← 0 ;

Pour somme allant de 1 à 10 à pas de 1 faire ;

somme ← somme + n ;

Afficher ‟ La somme des 10 premiers nombres est” somme ;

Fin Pour

Fin

**EXERCICE 16 :**

Algorithme Table\_de\_8 ;

Var

I : entiers ;

Début

Pour i allant de 0 à 10 à pas de 1 faire ;

Afficher ‟ 8 \*”, i, ‟=”8 \*i ;

Fin Pour

Fin

**Exercice 17 :**

Algorithme Table\_de\_10 ;

Var

I, j : entiers ;

Début

Afficher ‟ Voici les 10 tables de 10” ;

Répéter

i ← 1 ;

Répéter

j ← 0 ;

Afficher i, ‟\*”, j, ‟=” i \*j ;

j ← j+1 ;

jusqu’à j ˃= 12 ;

i ← i+1 ;

jusqu’à i ˃= 10 ;

Fin Répéter

Fin

**Exercice 18 :**

Algorithme Factoriel ;

Var

n, i, f : entiers ;

Début

f ← 1 ;

Afficher ‟Veuillez entrer un nombre” ;

Saisir n ;

SI n ˂ 0 alors ;

Afficher ‟Entrer un nombre positive” ;

Pour i allant de 1 à n à pas de 1 faire ;

f ← f \* i ;

Fin Pour

Afficher ‟Le factoriel de ”, n, ‟est”, f ;

Fin

**Exercice 18 :**

Algorithme Factoriel ;

Var

x, i, n:entier;

Début

Afficher ‟Veuillez entrer un nombre positive” ;

Saisir n ;

x ← 1 ;

Pour i allant de 2 à n faire :

x ← x \* i ;

Fin pour

Afficher ‟Le factoriel de ”, n, ‟est”, x ;

Fin

**Exercice 19 :**

Algorithme PGCD ;

Var

a, b, PGCD:entier

Début

Afficher ‟ Entrer deux nombres” ;

Saisir a, b ;

Tant que a \* b < > 0 faire :

Si a > b Alors

a ← a – b ;

Sinon

b ←b - a ;

Fin Si

Fin Tant que ;

Si a = 0 alors

PGCD ← b

Sinon

PGCD ← a

Fin Si

Afficher ‟ le plus grand commun diviseur est : ” PGCD ;

Fin

**LES TABLEAUX**

**Exercice 1**

Algorithme Creation\_de\_tableau ;

Var

Tab (1 ; 10) : tableau de type entiers ;

i : entier ;

Début

Pour i allant de 1 à 10 à pas 1 faire

Afficher ‘’ Entrez un nombre de la case ‘’, i ;

Saisir tab [i] ;

Fin pour

Fin.

**Exercice 2**

Algorithme Editer\_tableau ;

Var

tab (1 ; 10) : tableau de type entiers ;

i : entier ;

Début

Pour i allant de 1 à 10 à pas 1 faire

Afficher ‘’ Le contenu de la case ‘’, i, ‘’ est ‘’, tab[i] ;

Fin pour

Fin.

**Exercice 3**

Algorithme Calcul\_somme ;

Var

tab (1 ; 10) : tableau de type entiers ;

i, som : réels ;

Début

som ← 0

Pour i allant de 1 à 10 à pas 1 faire

som ← som + tab[i] ;

Fin pour

Afficher ‘’ La somme des réels dans le tableau est : ‘’, som ;

Fin.

**Exercice 4**

Algorithme Afficher\_minimum ;

Var

tab (1 ; 10) : tableau de type entiers ;

i, min : réels ;

Début

i ← 1

min ← tab[i]

Pour i allant de 1 à 10 à pas 10 faire

Si tab[i] < min Alors

min ← tab [¡] ;

Fin si

Fin pour

Fin.

**Exercice 5**

Algorithme Rechercher\_nom ;

Var

tab (1 ; 10) : tableau de type entiers ;

i : réels ;

trouver : booléen ;

nom : chaîne de caractère ;

Début

Afficher ‘’ Veillez entrer un nom ‘’ ;

Saisir nom ;

trouver ← 0

Pour i allant de 1 à 10 à pas 1 faire

Si nom = tab[i] Alors

trover ← 1 ;

Fin si

Fin pour

Si trover = 1 Alors

Afficher ‘’ Le nom a été trouvé ‘’ ;

Sinon

Aficher ‘’ Le n’a pas été trouvé ‘’ ;

Fin si

Fin.

**Exercice 6**

Algorithme Occurrence ;

Var

tab (1 ; 50) : tableau de type entiers ;

i, occurrence : réels ;

nom : chaîne de caractère ;

Début

Afficher ‘’ Veillez entrer un nom ‘’ ;

Saisir nom ;

occurrence ← 0

Pour i allant de 1 à 50 à pas 1 faire

Si nom = tab[i] Alors

occurrence ← occurrence + 1 ;

Fin si

Fin pour

Afficher ‘’ Le nom a été trouvé ‘’, occurrence, ‘’ fois ‘’ ;

Fin.

**Exercice 7**

Algorithme Suppression ;

Var

tab(1,10) : tableau de type entier ;

i, nombre, position\_trouver : entiers ;

trouver : booléen ;

Début

Afficher ‘’ Veillez entrez un nombre ‘’ ;

Saisir nombre ;

trouver ← 0 ; i ← 1 ;

Pour i allant de 1 à 10 à pas 1 faire

Si tab[i] = nombre Alors

trouver ← 1 ;

position\_trouver ← i ;

Fin si

Fin pour

Si trouver = 0 Alors

Afficher ‘’ l’élément est introuvable ‘’ ;

Sinon

Pour i allant de position\_trouver à 9 à pas 1 faire

tab[i] ← tab[i + 1]

Fin pour

Fin si

tab[10] ← 0

Fin.

**Exercice 8**

Algorithme Somme\_tableau ;

Var

tab1(1,5), tab2(1,5), tab3(1,5) : tableau de type entiers ;

i : entier ;

Début

Pour i allant de 1 à 5 à pas 1 faire

tab3[i] ← tab1[i] + tab2[i]

Fin pour

Fin.

**Exercice 9**

Algorithme Tri\_croissant ;

Var

tab(1, 5) : tableau de type entiers ;

min, i, permute, j, position : entiers ;

Début

Pour i allant de 1 à 5 à pas 1 faire

min ← tab[i] ;

position ← i ;

Pour j allant de i à 5 à pas 1 faire

si tab[j] < min Alors

min ← tab[j] ;

position ← j ;

Fin si

Fin pour

permute ← tab[i] ;

tab[i] ← min ;

tab[position] ← permute ;

Fin pour

Fin.

**Exercice 10**

Algorithme Tri\_croissant ;

Var

tab(1, 5) : tableau de type entiers ;

max, i, permute, j, position : entiers ;

Début

Pour i allant de 1 à 5 à pas 1 faire

max ← tab[i] ;

position ← i ;

Pour j allant de i à 5 à pas 1 faire

si tab[j] > max Alors

max ← tab[j] ;

position ← j ;

Fin si

Fin pour

permute ← tab[i] ;

tab[i] ← max ;

tab[position] ← permute ;

Fin pour

Fin.