

Exercice 1

Algorithme Bissextile;

Var

Annee: Entier;

Début

Ecrire("Entrez l'année");

Lire(Annee);

Si (Annee % 4 = 0) alors

Si (Annee % 100 = 0) alors

Si (Annee % 400 = 0) alors

Ecrire("pas Bissextile");

Sinon

Ecrire("Bissextile");

Fin Si

Fin Si

Sinon

Ecrire("pas Bissextile");

Fin Si

Fin.

Exercice 2

Algorithme Somme;

Var

i, n: Entier;

sgm: Entier;

début

Ecrire("Entrez un nombre");

Lire(n);

sgm ← 0;

Pour i ← 0 à n faire

sgm ← sgm + i;

Fin Pour;

Ecrire("la somme est: ", sgm);

Fin

Exercice 3

Algorithme Somme-Carrés;

Var

i, n: Entier;

sgm: Entier;

début

Ecrire("Entrez un nombre");

Lire(n);

sgm ← 0;

Pour i ← 0 à n faire

sgm ← sgm + (i + i);

Fin Pour;

Ecrire("la somme est: ", sgm);

Fin

exercice 1;

1) Algorithme Factorielle;

Var

n: Entier;

Fonction factorielle: Entier;

début

Si (n=0 || n=1) alors

retourner 1;

Sinon

retourner (n * factorielle(n-1));

Fin Si

Fin Fonction

Début

Ecrire ("Entrez un nombre");

Lire(n);

Si (n >= 0)

Ecrire ("n! = ", factorielle(n));

Sinon

Ecrire (n, "est négatif");

Fin Si

Fin

2)

Algorithme Factorielle;

Var

i, n: Entier.

Fonction factorielle: Entier;

début

Si (n=0 || n=1) alors

retourner 1;

Sinon

retourner (n * factorielle(n-1));

Fin Si;

Fin Fonction.

début

pour i ← 0 à 10 faire

Ecrire(i, "!" = ", factorielle(i));

Fin Pour;

Fin.

Exercices

Algorithme Pythagore;

Var

x, y, z, max : Entier;

Fonction : gcd : entier;

Var

r : Entier;

debut

Tout One (1) faire

$r \leftarrow x \% y$;

if ($r = 0$)

retourner y ;

$x \leftarrow y$;

$y \leftarrow r$;

Fin Tout One;

Fin Fonction.

debut

Ecrire ("Enter une valeur maximale à ne pas dépasser");

Lire(max);

Pour $x \leftarrow 1$ à max faire

Pour $y \leftarrow 1$ à max faire

Pour $z \leftarrow 1$ à max faire

Si ($(x^2 + y^2 = z^2) \&$

$(x < y) \& (y < z) \& \text{gcd}(x, y) = 1$

$\& \text{gcd}(x, y) = 1 \& \text{gcd}(z, y) = 1$)

Ecrire (" ", x , " ", y , " ", z , " ");

Fin Si

Fin Pour

Fin Pour

Fin Bar

Fin.

Exercice

Algorithme Armstrong;

Var

$\text{nbr}, d, \text{temp}, \text{somme}$: Entier;

debut

Ecrire ("Enter un nombre: ");

Lire(nbr);

$\text{temp} \leftarrow \text{nbr}$;

$\text{somme} \leftarrow 0$

Tout One (nbr > 0) faire

$d = \text{nbr} \% 10$;

$\text{somme} \leftarrow \text{somme} + (d * d * d)$;

$\text{nbr} = \text{nbr} / 10$;

Fin Tout One

Si ($\text{temp} = \text{somme}$) alors

Ecrire ("nbr Armstrong");

Sinon

Ecrire ("n'est pas un nbr Armstrong");

Fin Si

Fin

Exercice 4

1) Algorithme Diviseurs;

Var

i, n: Entier;

début

Ecrire("Entrez un nombre");

lire(n);

Pour i ← 1 à n/2 faire

Si (n % i = 0) faire

Ecrire(n % i);

Fin Si

Fin Pour

Fin

2) Algorithme Classification;

Var

sem, i, n: Entier;

Tableau T[100]: Entier;

Fonction IsTDiviseurs: Entier;

Var

i, n: Entier;

Tableau T[100]: Entier;

début

Pour i ← 1 à n/2 faire

Si (n % i = 0) faire

T[i-1] ← n % i;

Fin Si

Fin

début

Répondre

Ecrire("Entrez un nbr positif:");

lire(n);

Jusqu'à (n >= 0);

T ← IsTDiviseurs(n);

sem ← 0;

Pour i ← 0 à 100 faire

sem ← sem + T[i];

Fin Pour

Si (sem = n) alors

Ecrire("Le nbr est parfait");

Fin Si

Si (sem < n) alors

Ecrire("Le nbr est déficient");

Fin Si

Si (sem > n) alors

Ecrire("Le nbr est abondant");

Fin Si

Fin

exercice 7

Algorithme Fibonacci;

Var

nbr: Entier;

Fonction Fibonacci: Entier;

Var

nbr: Entier;

début

Si (nbr = 0 || nbr = 1) alors
retourner 1;

Sinon

retourner (Fibonacci(n-1) + Fibonacci(n-2));

FinSi

Fin Fonction.

début

Répete

Ecrire ("enter un nbr");

Lire(nbr);

Jusqu'à (n >= 0);

Ecrire ("Le terme n de la suite Fibonacci
est", Fibonacci(nbr));

Fin.

exercice 8

Algorithme Min-Max-Moy;

Var

i, nbr: Entier; Entier;

min, max, moy, Som: Double;

Tableau Suite[nbr-Entier];

début

Répete

Ecrire ("Entrez nbr des éléments de votre suite
à condition qu'il n'est pas nulle ou
négative");

Lire(nbr-Entier);

Jusqu'à (nbr > 0);

Ecrire ("Maintenant veuillez saisir votre
suite de nombre");

Pour i ← 0 à nbr-Entier faire

Répete

Ecrire ("Veuillez numéro: ", i+1);

Lire(suite[i]);

Jusqu'à (suite >= 0);

Fin Pour;

Som ← suite[0];

min ← suite[0];

max ← suite[0];

Pour i ← 1 à nbr-Entier faire

Som ← Som + suite[i];


```

Si (suite[i] < min) faire
    min ← suite[i];
Sinon Si (suite[i] > max)
    max ← suite[i];

```

Fin Si;

Fin Pour;

max ← Som (x);

Ecrire ("l'élément minimum de la suite est:", min);

Ecrire ("l'élément maximum de la suite est:", max);

Ecrire ("la moyenne de la suite est:", moy);

Fin.

Exercice 10

Algorithme End-Itérative;

Pour

degre: Entier;

x: Réel;

Tableau A[10]: Réel;

resultPoly: Réel;

Fonction Pui: Entier;

Pour

temp, i, p, B: Entier;

début

Temp ← B;

Pour i ← 0 à p faire

B ← B * temp;

Fin Pour;

retourner B;

Fin Fonction;

Fonction eval-iterative: Entier;

Pour

i, degre: Entier;

x: Réel;

Tableau A[10]: Réel;

resultPoly: Réel;

début

Pour i ← 0 à degre faire

Ecrire ("Quelle valeur aura votre

coefficient A", i);

Lire(A[i]);

Fin Pour;

resultPoly ← 0;

Pour i ← 0 à degre faire

resultPoly ← resultPoly + A[i] * Pui(x,

degre - i);

Fin Pour;

retourner resultPoly;

Fin Fonction;

début

Répeté

lire (degre);

Jusqu'à (degre ≤ 10);

Ecrire ("Entre la valeur de x de votre réel");

Lire(x);