

1) Résolution d'une équation du 1^{er} degré réelle à une inconnue

Rédiger l'algorithme « Equation1erDegre » qui :

- demande à un utilisateur de saisir trois nombres réels a, b et c ;
- calcule la solution s de l'équation $ax+b=c$;
- affiche la solution (ou plus exactement, afficher le message : « La solution est : [solution] »)

Algorithme Equation1erDegre

Variables

a : réel
b : réel
c : réel
x : réel

Début

```
// récupération des saisies utilisateur
afficher("Il va vous être demandé de saisir a, b et c, les coefs de l'équation ax+b=0")
a ← saisir("Nombre a :")
b ← saisir("Nombre b :")
c ← saisir("Nombre c :")
// gestion du cas particulier a = 0, aucune solution
SI a = 0 ALORS
    afficher("Aucune solution")
SINON
    x ← (c - b) / a
FIN SI
// on affiche la solution
afficher("La solution de l'équation est : " + x)
```

Fin

2) Résolution d'une équation du 2nd degré réelle à une inconnue

Rédiger l'algorithme « Equation2ndDegre » qui :

- demande à un utilisateur de saisir trois nombres réels a, b et c ;
- calcule, si elle(s) existe(nt), la ou les solutions réelle(s) de l'équation $ax^2+bx+c=0$;
- affiche, s'il y en a, la ou les solutions.

Corrigé déjà fourni dans le devoir n°1

3) Calcul de la puissance d'un nombre

Rédiger la fonction « CalculePuissance(x : réel, n : entier naturel) : réel » qui retourne la puissance du nombre x à la puissance d'exposant n. Question subsidiaire : pour un nombre n donné, combien de tour de boucle avez-vous fait ? (autrement dit, combien de multiplication vous faut-il faire ?)

Fonction CalculePuissance(X : réel, n : entier naturel) : réel

Variables

resultat : réel

i : entier

Début

resultat \leftarrow 1

// on multiplie 1 par X exactement n fois, ce qui vaut bien X^n

POUR i **DE** 1 **A** n **PAS DE** 1

resultat \leftarrow resultat x X

FIN POUR

RETOURNER resultat

Fin Fonction

Deux variantes (hors programmes, une fonction peut s'appeler elle-même, c'est ce qu'on appelle la récursivité). Si vous ne comprenez pas, inutile de vous attarder.

Fonction CalculePuissance(X : réel, n : entier naturel) : réel

Début

SI N = 0 **ALORS**

RETOURNER 1

SINON

RETOURNER X x CalculePuissance(X, n-1)

FIN SI

Fin Fonction

Plus efficace (performance)

Fonction CalculePuissance(X : réel, n : entier naturel) : réel

Début

// Si n est pair

SI n % 2 = 0 **ALORS**

SI n = 0 **ALORS**

RETOURNER 1

SINON

// Pourquoi faire ça ? Parce que $X^n = (X^2)^{n/2}$, autrement dit, on a besoin de bien

// moins de multiplications pour obtenir le résultat ($\log_2(n)$ multiplications au

// lieu de n)

RETOURNER CalculePuissance(X x X, n / 2)

FIN SI

// Sinon n est impair

SINON

RETOURNER X x CalculePuissance(X, n - 1)

FIN SI

Fin Fonction

4) Calcul d'une moyenne pondérée

Rédiger l'algorithme « MoyennePonderee » qui :

- demande à l'utilisateur de saisir « une note » note et « un coefficient » coefficient tant qu'il ne saisit pas « FIN » ;
- calcule au fur et à mesure la somme des note x coefficient ;
- finit de calculer la moyenne des notes ;
- affiche la moyenne.

Algorithme MoyennePonderee (X : réel, n : entier naturel) : réel

Variables

estFin : chaîne de caractères

note : réel

coef : entier naturel

moyenne : réel

sommeCoef : entier naturel

Début

// on initialise les variables

nombreNotes \leftarrow 0

moyenne \leftarrow 0

sommeCoef \leftarrow 0

// tant que l'utilisateur n'a pas saisi FIN

TANTQUE estFin \neq "FIN" **FAIRE**

 // on demande une note et un coefficient à l'utilisateur

 note \leftarrow saisir("Saisir une note : ")

 coef \leftarrow saisir("Saisir un coefficient :")

 // on calcule la somme des note x coef

 moyenne \leftarrow moyenne + note x coef

 // on calcule le somme des coefficients

 sommeCoef \leftarrow sommeCoef + coef

 // on continue si l'utilisateur saisit tout sauf FIN

 estFin \leftarrow saisir("Saisir FIN pour arrêter :")

FIN TANTQUE

 // on finit le calcul de la moyenne

 moyenne \leftarrow moyenne / sommeCoef

 // on affiche la moyenne calculée

 afficher("La moyenne est de : " + moyenne)

Fin