

révisions générales

I - Révisions générales

II - les alternatives

III - les boucles

IV - les tableaux

V - Les paramètres de la fonction

VI - les fonctions

les notions générales

1 - Syntaxe générale

- Algo : nom algo Declaration | Objets de données Début | traitement Fin nom algo

Les Objets de données :

entier(int), réel(float, double) caractères(char), booléen

Opérateurs : - | Négation | -\$a Opposé de \$a

+ | Addition | \$a + \$b Somme de \$a et \$b

* | Multiplication | \$a * \$b Produit de \$a et \$b

- | Soustraction | \$a - \$b Différence de \$a et \$b

/ | Division | \$a / \$b Quotient de \$a et \$b

% | Modulo | \$a % \$b Reste de \$a / \$b

++ Pré-Incrémentation ++\$a Incrémente \$a, puis retourne \$a

++ Post-Incrémentation \$a++ Retourne \$a, puis incrémente \$a

-- Pré-Décrémentation --\$a Décrémente \$a, puis retourne \$a

-- Post-Décrémentation \$a-- Retourne \$a, puis décrémente \$a

= Assignment \$a = 3 Affecte la valeur 3 à \$a

== Egalité en valeur \$a == \$b Vérifie que les valeurs de \$a et \$b sont identiques

=== Egalité en valeur et type \$a === \$b Vérifie que les valeurs et types de \$a et \$b sont identiques

!= Différence en valeur \$a != \$b Vérifie que les valeurs de \$a et \$b sont différentes

!= Différence en valeur et type \$a != \$b Vérifie que les valeurs et types de \$a et \$b sont différents

<> Différence en valeur \$a <> \$b Alias de !=

< Infériorité stricte \$a < \$b Vérifie que \$a est strictement inférieur \$b

<= Infériorité ou égalité \$a <= \$b Vérifie que \$a est strictement inférieur ou égal à \$b

> Supériorité stricte \$a > \$b Vérifie que \$a est strictement supérieur \$b

>= Supériorité ou égalité \$a >= \$b Vérifie que \$a est strictement supérieur ou égal à \$b

Exemples

```
include <stdio.h>
<p>
int main(){<br>
float L,l,a,p

printf("donner la largeur");<br>
scanf("%f",&L);
printf("donner la longueur");
scanf("%f",&l);

a= lL;<br>
printf("la aire est de %f",a);

p=(L2) + (L2);<br>
printf("le perimetre est de %f",p);
}
```

```

<form method="post" action="">
Longueur:<input type="text" name="L"><br>
Largeur:<input type="text" name="l"><br>
<input type="submit" name="Valider" value="Valider">
<input type="reset" name="Annuler" value="Annuler"><br>
<?PHP

if(isset($_POST['Valider'])){
$l = $_POST['l'];
$L = $_POST['L'];

$a = $l$L;
$p = $l2 +$l2;
printf("le perimetre est de %f",$p);
printf("l'aire est de %f",$a);
}
?>

```

Ecrire un Algo un prog C un dev PHP qui permet de calculer le prix TTC, la valeur tva d'un produit un hors taxe, une tva, une quantité

II - les alternatives

Ex1 : résoudre dans $ax + b = 0$

Ex2 : résoudre equation second degré $ax^2 + bx + c = 0$

Algo:

```

Algo: équation
Déclaration :
| a, x, b reel
Début:
| Afficher("saisir 1er coef")
| Saisir(a)
| Afficher("saisir 2eme coef")
| Saisir(b)
| Si A = 0
| | Alors si b = 0
| | | Alors Afficher("ENS des R")
| | | Sinon Afficher("ENS Vide")
| | Fin Si
| | Sinon x <- -b/a
| | Afficher("La solution est : x")
| Fin Si
Fin équation

```

C:

```

#include <stdio.h>
int main() {
float a, x, b;
printf("saisir a");
scanf("%f", &a);
printf("saisir b");
scanf("%f", &b);
if(a == 0) {
    if(b == 0) {
        printf("Ensemble des R");
    } else {
        printf("ensemble vide");
    }
} else {
    x = -b/a;
    printf("La solution est %f", x);
}
return 0;
}

```

PHP :

```

<html>
<head>
<title>equation</title>
<meta charset="utf-8" />

```

```

</head>
<body>
  <form method="POST" action="equation_1_php">
    coef1
    <input type='text' name="coef1" />
    coef2
    <input type='text' name="coef2" />
    <input type='submit' name="valider" value="valider" />
    <input type='reset' name="annuler" value="annuler" />
  </form>
  <?php
    if (isset($_POST['valider'])) {
      $a = $_POST['coef1'];
      $b = $_POST['coef2'];
      if($a == 0) {
        if($b == 0) {
          echo "Ens des r";
        } else {
          echo "ensemble vide";
        }
      } else {
        $x = -$b/$a;
        echo 'la solution est : ' + $x;
      }
    }
  ?>
</body>
</html>

```

III - les boucles

La boucle tant que

Algo:

```

tantque 'condition' faire
| instruction
fin tantque

```

C et PHP :

```

while(condition) {
  instruction;
}

```

La boucle faire - tantque

Algo:

```

faire
| action
tantque
| condition

```

PHP et C :

```

do {
  instruction;
}
while (condition) {
}

```

La boucle pour :

Algo :

```

Algo: pour 'indice' allant de 'valeur debut' à 'valeur fin' faire
      | instruction
      fin pour

```

C et PHP :

```

for(initialisation, condition, evolution) {
    instruction;
}

```

Exercice 1 :

Ecrire un algo, un prog en C, dev PHP qui permet de saisir un nombre entier et affiche ses diviseurs.

Exercice 2 :

Ecrire un algo, un prog en C, dev PHP qui détermine les nombres parfait entre deux bornes entières saisies.

Un nombreest parfait si il est égale à la somme de ses diviseurs sauf a lui meme.

PHP:

```

<html>
  <head>
    <title>Diviseurs</title>
    <meta charset="utf-8" />
  </head>
  <body>
    <form method="POST" action="diviseur.php">
      nb
      <input type="text" name="nb" />
      Valider
      <input type="submit" value="valider" name="valider">
      <input type="reset" value="annuler" name="valider">
    </form>
    <?php
      if(isset($_POST['valider'])) {
        $nb = $_POST['nb'];
        $div = 1;
        while($div <= $nb) {
          if( $nb % $div == 0) {
            echo ('diviseur : '.$div);
          }
          $div ++;
        }
      }
    ?>
  </body>
</html>

```

C :

```

#include<stdio.h>
int main() {
    int nb, div;
    printf("Donner un nombre");
    scanf('%d', &nb);
    div = 1;
    while(div <= nb) {
        if( nb % div == 0) {
            printf("diviseur: %d", div)
        }
        div ++;
    }
    return 0;
}

```