

Progression ultra basique

Pour chaque paragraphe, copier le programme dans Execalgo, l'exécuter et réfléchir au résultat obtenu.
Puis faire les exercices demandés.

1/ Afficher, opérations, arrondi, racine carrée, π

```
Afficher( "Bonjour" )
Afficher( " A = " ; 17 )
Afficher( "17 + 2" )
Afficher( 17 + 2 )
Afficher("17 + 2 = " ; 17 + 2 )
Afficher("17 + 2 = " ; 17 - 2)
Afficher( 2*(5-8) )
Afficher( 100-2^3 )
Afficher(2/3)
Afficher("pi arrondi au centième : " ; arrondi(pi , 2) )
Afficher("racine carrée de 2 arrondie au millième : " ; arrondi ( rac(2) , 3 ) )
```

Exercice 1.1 : écrire un programme qui affiche $7^2 + \frac{1}{3}$ arrondi au centième, précédé du texte : "Résultat : ".

Exercice 1.2 : écrire un programme qui affiche $\frac{\pi+1}{\pi-1}$ arrondi au millième, précédé du texte : "A = " et suivi du texte " environ".

2/ Variable, Demander, commentaire

```
Variable(entier ; n )
Demander(n)
# Le signe multiplier ne doit pas être sous-entendu
#Le logiciel exécutant ne lit pas les commentaires
Afficher("Le double du nombre entré est " ; 2*n )
```

Exercice 2.1 : écrire un programme qui demande un entier n à l'utilisateur et qui affiche le triple de son suivant.

Exercice 2. 2 : écrire un programme qui demande un réel x à l'utilisateur et qui affiche la valeur de $\frac{2x+3}{5}$.

3/ FaireSi, reste

```
Variable(entier ; n )
Demander(n)
# reste(n , 2) est le reste de la division euclidienne de n par 2
FaireSi ( reste(n , 2) = 0 )
    Afficher("Le nombre est pair")
Sinon
    Afficher("Le nombre est impair")
FinFaire
```

Exercice 3.1. : écrire un programme qui demande un entier n à l'utilisateur et qui affiche son carré s'il est supérieur à 10 et son cube dans le cas contraire.

Exercice 3.2. : écrire un programme qui demande un entier n à l'utilisateur et qui indique si ce nombre est ou non multiple de 3.

Remarque : la fonction prédéfinie divise permet aussi de savoir si un nombre entier est divisible par un autre. Elle renvoie donc un booléen (Vrai ou Faux) :
 $\text{reste}(n , 2) = 0$ est synonyme de $\text{divise}(2 , n)$ ($2 \text{ divise } n$)

4/ Affecter

Variable(entier ; n ; p)
Demander (n)
Affecter(p ; n + 100)
Afficher(n)
Afficher(p)
Afficher(n+p)

Exercice 4.1. : écrire un programme qui demande un réel x à l'utilisateur et qui calcule puis affiche :
 $a=(x+3)^2$; $b=(x-3)^2$; $c=(x+3)(x-3)$; $d=a+b-2c$.

Exercice 4.2. : écrire un programme qui demande un réel x à l'utilisateur et qui calcule puis affiche :
 $a=(x+4)(x-6)$; $b=(x-2)^2+3$; $c=a-b$

5/ Affecter une variable en utilisant sa valeur

Variable(entier ; n)
Demander (n)
Affecter(n ; n + 100)
Afficher(n)

Exercice 5.1. : écrire un programme qui demande deux réels x et y à l'utilisateur, qui double l'un, qui triple l'autre et qui calcule puis affiche : $z=x^2-y^2$.

Exercice 5.2. : écrire un programme qui demande un entier n à l'utilisateur, qui le remplace par le suivant de son double, puis qui affiche son carré.

6/ Variable de type texte

Variable(texte ; t)
Afficher("Donner un nom")
Demander(t)
Affecter(t ; "Bonjour " + t)
Afficher(t)

Exercice 6.1. : écrire un programme qui demande un texte t à l'utilisateur, qui lui ajoute trois points d'exclamation, et qui l'affiche en deux exemplaires séparés par un espace.

Exercice 6.2. : écrire un programme qui demande deux textes t1 et t2 à l'utilisateur, et qui affiche le premier des deux dans l'ordre alphabétique (utiliser l'opération < valable entre deux textes).

7/ FaireFois, notion d'intérieur d'une boucle

Variable(entier ; n)
Demander (n)
Afficher("avant la boucle")
FaireFois (8)
 Affecter(n ; n+1)
 Afficher(n)
 Afficher("dans la boucle")
FinFaire
Afficher("après la boucle")

Exercice 7.1. : écrire un programme qui demande un entier n et un texte t à l'utilisateur, qui ajoute n points d'exclamations à la fin du texte t puis qui affiche la valeur de t.

Exercice 7.2. : écrire un programme qui demande un entier n à l'utilisateur et affiche un texte composé de n fois la lettre l.

8/ FaireTantQue

```
Variable(entier ; n )  
Demander ( n )  
Afficher("avant la boucle")  
FaireTantQue ( n > 10 )  
    Affecter(n ; n-10)  
    Afficher(n)  
    Afficher("dans la boucle")  
FinFaire  
Afficher("après la boucle")
```

Exercice 8.1. : écrire un programme qui demande un entier n à l'utilisateur, qui lui ajoute 2 tant qu'il n'est pas multiple de 7, qui affiche sa valeur à chaque passage dans la boucle. puis son carré après la boucle.

Exercice 8.2. : écrire un programme qui demande un entier n à l'utilisateur, puis qui, tant que n est positif, lui retire 2 et affiche son carré.