

TP n° 3 : Algo & Arithmétique

Traduire en C# chacun des algorithmes suivants
Compléter les formulaires associés sur padlet.

**POUR LES ENTIERS, ON UTILISERA LE TYPE
ULONG**

On rappelle que si a et b sont de type entier, a/b
est le quotient entier et a%b est le reste

L'instruction POUR I variant de A à B on écrira
for(int I=A ; I<=B ; I=I+1)

Exercice 1: BOUCLES POUR SIMPLES

1)

```
Demander x (ULONG)
POUR I VARIANT DE 1 à 5
  x ← 3*x+2*I
FIN POUR
Afficher x
```

2)

```
Le test I%3==0 est VRAI si I est divisible par 3
S1 ← 0 (ULONG)
Demander N
POUR I VARIANT DE 1 à N
  SI I est divisible par 3 ALORS
    S1←S1+I
  FIN SI
FIN POUR
Afficher S1
```

3)

```
S1 ← 0 (ULONG)
Demander N et P (int)
POUR I VARIANT DE 1 à N
  SI I est divisible par P ALORS
    S1←S1+I
  FIN SI
FIN POUR
Afficher S1
```

Exercice 2: TANT QUE

A, B, C sont des entiers
Traduire les algorithmes suivants en C#. tester
puis répondre aux questions du padlet.

```
A ← 0
TANT QUE A<=5 OU A > 100
  DEMANDER A
FIN TANT QUE
Afficher A

B ← 0
TANT QUE B % 2 = 0 OU B > 100
  DEMANDER B
FIN TANT QUE
Afficher B
```

```
C ← 10 ; B←10
TANT QUE C>9 ET B % 5 = 0
  DEMANDER C
  DEMANDER B
FIN TANT QUE
Afficher B et C
```

**PAR LA SUITE, TOUTES LES
DEMANDES FAITES A
L'UTILISATEUR, DU TYPE
'demander un entier positif'
DOIVENT ETRE VALIDEES PAR
UNE BOUCLE « TANT QUE »
DE CE TYPE**

Exercice 3:

Ecrire sur papier un algorithme qui saisit un
nombre entier p, puis qui affiche la somme des
carrés successifs jusqu'au p-ième carré obtenu,
c'est-à-dire $1^2+2^2+3^2+...+p^2$

Voir le TP sur padlet si vous êtes bloqués

Traduisez-le en C#

Exercice 4: division euclidienne (reste)

Ecrire un programme C# qui :

- ✓ Au moyen d'une boucle TANT QUE, demande un entier N positif (**ULONG**)
- ✓ Au moyen d'une boucle TANT QUE, demande un nombre D supérieur à 1 et inférieur à N
- ✓ Fait varier I de 1 à D (avec une boucle POUR)
- ✓ Affiche le quotient et la division de N par I sous la forme « $N=I \times q + r$ »

Exemple : si vous entrez N=13 et D=5 le programme doit afficher 13=1x13+0 ; 13=2x6+1 ; 13=3x4+1 ; 13=4x3+1 ; 13=5x2+3.

Exercice 5: Somme des multiples

Ecrire un programme en C# qui :

- ✓ demande **au moyen d'une boucle** un entier supérieur à 1 : P
- ✓ Demande **au moyen d'une boucle** un nombre M plus grande que P
- ✓ Affiche les multiples de P inférieurs à M
- ✓ Calcule leur somme S et l'affiche à la fin

Exemple : si P=2 et M=15, les multiples sont 0 ; 2 ; 4 ; 6 ; 8 ; 10 ; 12 ; 14 et leur somme fait 56

Exercice 6: Nombres parfaits

Un nombre est parfait s'il est égal à la moitié de la somme de ses diviseurs

Ecrire un programme en C# qui :

- ✓ demande **au moyen d'une boucle** un entier supérieur à 1 : P
- ✓ Calcule S : la somme de ses diviseurs et l'affiche à la fin
- ✓ Affiche « le nombre est parfait » ou « le nombre n'est pas parfait » suivant les cas

Exemple : si P=28, les diviseurs sont 1 ; 2 ; 4 ; 7 ; 14 et 28, leur somme fait 56, il est donc parfait

Exercice 7: Tous les nombres parfaits

Ecrire un programme qui demande **au moyen d'une boucle** une borne (variable) B>1 et affiche tous les nombres parfaits entre 2 et B

On utilisera DEUX boucle POUR imbriquées. Lisez le TP sur padlet si vous êtes bloqué.

Exemple : entre 1 et 10000 il y en a 4 : 6 ; 28 ; 496 et 8128

Exercice 8: Nombres premiers

Ecrire en C# un programme qui

- ✓ demande **au moyen d'une boucle** un nombre entier P supérieur à 1
- ✓ Détermine s'il est premier et affiche une phrase de conclusion

L'algorithme est donné dans le cours

Exemple : si on rentre 19 il doit afficher « 19 est un nombre premier ». Si on rentre 123, il affiche « 123 n'est pas premier »

Exercice 9: Amélioration

Modifiez le programme précédent pour qu'il affiche, par exemple pour 123 : « 123 n'est pas premier, il est divisible par 3 »

Exercice 10: Tous les nombres premiers

Ecrire un programme qui :

- ✓ demande **au moyen d'une boucle** une variable M > 2
- ✓ Fait varier n de 2 à M (**boucle POUR**)
- ✓ Affiche n seulement si n est premier
- ✓ Compte et affiche combien il y a de nombres premiers entre 2 et M (inclus)

Exercice 11: Modif 1

Modifier le programme précédent pour qu'il calcule la somme S des nombres premiers inférieurs à M.

vérification : Pour M=10, on doit trouver S=17

Exercice 12: Modif 2

Reprendre le programme de l'exercice précédent ainsi :

- ✓ demander **au moyen d'une boucle** une borne SMax > 1
- ✓ Faire varier n de 1 en 1 à partir de 2
- ✓ Si n est premier, l'ajouter à S
- ✓ Tant que S < Smax
Afficher le résultat en le présentant clairement

vérification : Pour Smax=1000, on doit trouver S=1060

Exercice 13: Décomposition entière

Ecrire en C# le programme correspondant à l'algorithme de la diapo 7 qui :

- ✓ demande **au moyen d'une boucle** un nombre **positif** n
- ✓ détermine la chaîne de caractère résultat égale à sa décomposition en facteurs premiers
- ✓ Soigner l'affichage final

Exercice 14: PGCD et PPCM

Ecrire en C# un programme qui

- ✓ demande **au moyen d'une boucle** deux entiers a et b supérieurs à 1
- ✓ affiche leur pgcd et leur ppcm
- ✓ redemande a et b et recommence tant qu'ils ne sont pas nuls

Exercice 15: Premiers Entre Eux

Ecrire un programme qui

- ✓ Demande **au moyen d'une boucle** un entier N supérieur à 1
- ✓ Au moyen d'une boucle POUR sur X variant de 1 à N, affiche TOUS les nombres qui sont premiers avec N
C'est-à-dire tels que PGCD(N ;X)=1

Exercice 16: Challenge

Déterminez exactement combien il y a de nombres premiers entre 2 et 100 000 000

Utilisez les fonctions de timing (DateTime) pour déterminer le temps du calcul

Ayez le meilleurs temps pour gagner le bonus du semestre !