Nom : …..... Prénom : …..... Classe : …..

2nde – Algorithmique – TP 1 – Programmons l'algorithme d'Euclide

I- Rappels sur l'algorithme d'Euclide.

L'algorithme d'Euclide, étudié en classe de 3ème, permet de calculer le PGCD de deux entiers positifs a et b avec $a>b$ en se basant sur la propriété : $\text{PGCD}(a,b)=\text{PGCD}(b,r)$, où r est le reste de la division euclidienne de a par b. Essayons-le pour $a=4608$ et $b=3528$.

Où trouver la commande qui donne le reste de la division euclidienne sur votre calculatrice ?
CASIO : OPTN – NUM – MOD taper : MOD(4608,3528) pour avoir le reste de la division euclidienne de 4608 par 3528.
TEXAS INSTRUMENT : MATH – NBRE – reste : reste (4608,3528) ou *remainder* si la calculatrice est en anglais.

À chaque étape, tant que $r\neq 0$, remplacer a par b, b par r, et r par le reste de la division euclidienne de a par b.

Le PGCD des deux nombres de départ sera la dernière valeur non nulle de r. Entourez-la dans la colonne r et dans la colonne b.

a	b	r
4608	3528	

II- À quoi sert l'algorithmique ?

La programmation sert à demander à une machine de faire des tâches répétitives à notre place.
Nous allons créer un algorithme (série de directives) et le programmer dans plusieurs langages de programmation, sur ordinateurs et sur calculatrices, pour que, quand l'utilisateur entre deux entiers, la machine fasse les calculs toute seule à sa place et fournisse le PGCD des deux entiers.

Pour cet algorithme : nous aurons besoin de 3 cases-mémoires (*variables*) : A, B et R pour stocker des nombres.

On demandera à l'utilisateur d'entrer les valeurs de A (le plus grand des deux entiers) et de B (le plus petit).

L'ordinateur calculera R, le reste de la division euclidienne de A par B.

Tant que R ne vaudra pas zéro, il remplacera A par B et B par R, et calculera de nouveau R, le reste de la division euclidienne de A par B.

Et quand R vaudra 0, il affichera la valeur de B (voir la dernière ligne de votre tableau), qui sera le PGCD des nombres A et B du départ.

Vous remarquerez que les valeurs stockées dans les cases-mémoires changent au cours du développement du programme : à la fin, les valeurs stockées dans les cases A, B et R ne sont plus les mêmes que celles du début.

III- Programmons l'algorithme d'Euclide sur Algobox (sur ordinateur).

Ouvrir Algobox.

Cliquer sur : 

Déclarer trois variables : A, B et R du type NOMBRE.

Cliquer ensuite sur : 

Nom : …..... Prénom : …..... Classe : …....

Puis sur : 

.

Faites afficher les deux messages : « Ce programme va calculer le PGCD de deux entiers entrés par l'utilisateur » , puis « Entrez le plus grand des deux entiers », avec retours à la lignes.

Pensez à créer une nouvelle ligne pour chaque nouvelle instruction.

Ensuite, cliquer sur : 

, et faites lire la variable A.

Puis faire afficher : « Entrez le plus petit des deux entiers », et lire la variable B.

Après avoir créé une nouvelle ligne (toujours), cliquez sur : 

Et faites affecter à **R** la valeur **A%B** (c'est la commande pour le reste de la division euclidienne de A par B sur Algobox)

Nouvelle ligne : cliquer sur 

, entrer la condition : **R!=0** (pour dire : R≠0).

Dans la boucle « tant que », entrer les instructions (en créant les lignes nécessaires) :

- A prend la valeur B.
- B prend la valeur R.
- R prend la valeur A%B.

Créer deux nouvelles ligne après la boucle « tant que » :
- Afficher message : « Le PGCD des deux nombres que vous avez entrés est : »
- Afficher la variable B.

Ensuite, lancez l'algorithme et testez-le avec les nombres du paragraphe I ainsi que d'autres entiers.

Essayez aussi le mode « pas à pas » qui détaillera toute l'exécution du programme en vous montrant les valeurs prises par les variables à chaque étape. Ce mode est formidable pour comprendre le principe de la programmation.

Pour aller plus loin, si vous vous sentez à l'aise, améliorez ce programme :

- Pour qu'on puisse entrer les deux entiers dans l'ordre qu'on veut, et que le programme les replace lui-même dans le bon ordre avant de calculer le PGCD. Indice : on aura besoin de l'instruction : SI...ALORS.
- Pour que le programme calcule, en plus du PGCD, le PPCM de deux entiers. Il nous faudra deux nouvelles variables, par exemple AA et BB, qui garderont jusqu'à la fin du programme les valeurs entrées par l'utilisateur. On remarquera que le PPCM de deux entiers est égal à leur produit divisé par leur PGCD[1]. À la fin, changez le message pour qu'il affiche non plus : « Le PGCD des deux entiers entrés est : » mais « Le PGCD de » AA « et de » BB « est : ».

  Pour afficher le PPCM, utiliser : 

Les corrigés des programmes améliorés sont sur le site www.letableaunoir.net , pages des 2ndes, chapitre 0.

1 Pour 12 et 20 par exemple : $12=4\times3$, $20=4\times5$. Leur PGCD est 4. Leur PPCM est $4\times3\times5$ (partie commune que multiplie chacune des parties spécifiques), c'est aussi $\frac{12\times20}{4}$ soit $\frac{4\times3\times4\times4}{5}$.

Nom : …..... Prénom : …..... Classe : …....

IV- Programmons l'algorithme d'Euclide sur nos calculatrices.

CASIO

MENU : PRGM → NEW

Entrer le nom du programme : PGCD, puis EXE

ALPHA permet de taper les lettres en rouge au-dessus des touches. Faire SHIFT + ALPHA pour pouvoir taper plusieurs lettres à la suite.

Entrer les instructions :

```
======PGCD     ======
"ENTRER LE PLUS GRAND
 DES DEUX ENTIERS :":
?→A:"ENTRER LE PLUS P
ETIT DES DEUX ENTIERS
 :":?→B↵
```

: et ? sont accessibles par SHIFT → PRGM VARS, quant à →, elle est sur le clavier.

Remarquer que : sert de séparateur entre les instructions.

On peut aussi utiliser le retour à la ligne : ↵ avec la touche EXE.

Ensuite : `Mod(A,B)→R↵` stocke le reste de la division de A par B dans la variable R.

SHIFT → PRGM VARS → COM → While pour While

SHIFT → PRGM VARS → REL → ≠ pour ≠

SHIFT → PRGM VARS → COM → WEnd pour WhileEnd

Presser EXIT pour revenir en arrière dans les menus et ▷ pour les faire défiler.

```
While R≠0:B→A:R→B:Mod
(A,B)→R:WhileEnd↵
```

(Tant que R est non-nul, stocker B dans A, R dans B et le reste de la division euclidienne de A par B dans R)

```
"LE PGCD EST :":B
```

Pour essayer le programme, taper EXIT (éventuellement plusieurs fois) puis EXE sur le nom du programme.

Bon à savoir : SHIFT + CATALOG 4 permet d'accéder à toutes les commandes par ordre alphabétique.

TEXAS INSTRUMENT

prgm → NOUV

Entrer le nom du programme : PGCD, puis entrer

alpha permet de taper les lettres en vert au-dessus des touches. Faire 2nde → alpha pour pouvoir taper plusieurs lettres à la suite.

Entrer les instructions :

```
PROGRAM:PGCD
:Input "ENTRER LE PLUS GRA
ND DES DEUX ENTIERS :",A
:Input "ENTRER LE PLUS PET
IT DES DEUX ENTIERS :",B
:reste(A,B)→R
```

Input s'obtient par prgm → › → 1. C'est la commande « lire une variable », qui a l'avantage de permettre d'afficher un texte avant la saisie.

→ s'obtient par la touche sto→

À la fin d'une ligne, on presse entrer et on passe à l'instruction suivante.

Si votre calculatrice s'éteint, pour continuer à taper votre programme, faites prgm ÉDIT et revenez sur PGCD.

```
:While R≠0
:B→A
:R→B
:reste(A,B)→R
:End ⟵ Indique la fin des instructions de la boucle While
:Disp "LE PGCD EST :",B
     ↖ commande "afficher".
```

On remarque, dans *Input* comme dans *Disp*, que les guillemets permettent d'afficher un texte « tel quel », et qu'on peut insérer des variables à afficher sur la même ligne en utilisant une virgule comme séparateur.

While s'obtient par : prgm → 5

≠ s'obtient par : 2nde → tests A math → 2

End s'obtient par : prgm → 7

Disp s'obtient par : prgm → › → 3

Pour lancer le programme, faire 2nde → quitter mode, puis prgm et PGCD dans le mode EXE.

Nom : …..... Prénom : …..... Classe : …....

V- Programmons l'algorithme d'Euclide en langage Python.

(depuis le menu démarrer car il n'est pas sur le bureau)

Sur ordinateur, ouvrir le logiciel : IDLE (Python GUI), et ouvrir un nouveau fichier (FILE→NEW FILE) pour taper un programme. Entrer le programme suivant, sans les lignes qui commencent par #, puisque ce sont des lignes de commentaires qui ne font pas partie des instructions du programme.

```
print("ce programme va calculer le PGCD de deux entiers que vous allez nous fournir.")
a=input("entrez le plus grand des deux entiers : ")
b=input("entrez le plus petit des deux entiers : ")
# par défaut, Python considère les variables a et b que nous avons saisies avec l'instruction "input"
# comme des chaînes de caractères, du type "string"
# il ne peut donc pas faire du calcul avec et nous enverra un message d'erreur si on passe directement
# à l'instruction r=a%b
# il faut donc convertir a et b en "entiers" à l'aide des deux instructions suivantes :
a=int(a)
b=int(b)
# a et b sont maintenant du type "int" (entier)
r=a%b
while r!=0 :
    a=b
    b=r
    r=a%b
print ("Le PGCD des deux entiers est :",b)
```

Pour lancer le programme : taper F5
Accepter que Python l'enregistre : choisir un nom et un emplacement.
Effacer ce fichier à la fin du TP, sauf si vous l'avez enregistré sur une clé personnelle et souhaitez le garder.

Quelques remarques (en plus de celle des commentaires dans le programme) :

- En Python, on n'a pas besoin de commencer par déclarer les variables qui vont être utilisées (alors que c'est le cas dans d'autres langages de programmation, comme sur algobox ou en turbo-pascal)
- Pour saisir la valeur d'une variable, on utilise l'instruction INPUT(), mais on commence par le nom de la variable suivi de =. On peut ensuite ajouter un message dans des parenthèses, entre guillemets pour les parties « textuelles ».
- Pour afficher un texte, on utilise l'instruction PRINT(). À la dernière ligne, on remarque dans les parenthèses une partie textuelle entre guillemets, séparée par une virgule de la variable *b* à afficher.
- Le signe = sert à l'affectation. « a=2 » signifie qu'on affecte la valeur 2 à la variable *a* . Sur une calculatrice, on taperait : « 2→a ». Dans d'autres langages, l'instruction peut être : « a:=2 ».
- En Python, on n'emploiera pas le signe = pour tester si deux valeurs sont égales. Il faudra utiliser ==.
  Exemple : On peut ajouter ce test à la fin du programme :

```
if b==1:
    print("Les deux entiers sont premiers entre eux !")
else : print("Les deux entiers ne sont pas premiers entre eux.")
```

Si (if) le PGCD est 1, le programme affichera la phrase : « Les deux entiers sont premiers entre eux. », dans le cas contraire (else = sinon), il affichera : « Les deux entiers ne sont pas premiers entre eux. »

- Dans la boucle while comme dans la condition if, les « : » et la marge de retrait à gauche sont très importants.

Après ces ":" vont suivre les instructions à effectuer dans la boucle while.

```
while r!=0 :
    a=b
    b=r
    r=a%b
print ("Le PGCD des deux entiers est :",b)
```

pendant ces instructions on reste en retrait.

une fois qu'elles sont terminées, on écrit à nouveau au niveau du while.

Remarquer ici encore les ":" puis le retrait.

```
if b==1:
    print("Les deux entiers sont premiers entre eux !")
else : print("Les deux entiers ne sont pas premiers entre eux.")
```

Quand il y a une seule instruction, on peut l'écrire à la suite des ":".

Si vous êtes à l'aise : améliorez le programme Python et le programme sur calculatrice, en faisant en sorte que les deux entiers entrés puissent l'être dans n'importe quel ordre, et en faisant calculer leur PPCM en plus de leur PGCD.

Symbole de la multiplication en Python : *
Symbole de la division : /

Par exemple : $4\times\frac{2}{3}$ s'écrit : 4*2/3

Corrigé du programme amélioré en Python sur www.letableaunoir.net , page des 2ndes, chapitre 0.