Algorithmes et suites avec LARP

Stéphan MANGANELLI, LEGTA « Louis Giraud » de CARPENTRAS-SERRES (84)

Bonjour.
Je m'appelle Stéphan MANGANELLI, et j'enseigne les mathématiques au lycée agricole "Louis Giraud" de Carpentras-Serres (84).
Responsable de la filière S, j'ai bien été contraint de me mettre à l'algorithmique...

Quand on lit les objectifs des programmes scolaires, il s'agit essentiellement pour un premier apprentissage dirons-nous, de (re)mettre en lumière le procédé qu'est l'algorithme, en comprendre les rouages et se familiariser avec, pour une utilisation raisonnée et raisonnable dans la démarche scientifique, à ce niveau d'étude.
Il est recommandé de présenter les algorithmes en "langage naturel" et de les mettre en œuvre avec un langage de programmation ; dès lors, la calculatrice étant pour l'instant (épreuve pratique en sommeil) le seul outil de programmation à disposition des élèves en épreuve, il paraît incontournable de l'utiliser comme tel.
Rien n'empêche, bien au contraire, de sensibiliser les élèves à d'autres outils de programmation (avec langage propre) ; je citerai ici ***R***, très complet et performant, et je vous renvoie aux articles de mon collègue Hubert RAYMONDAUD qui le présente.

Pour ma part, ma modeste contribution ici, consiste à vous présenter en quelques lignes un logiciel - certes, direz vous, un de plus - mais j'allais plutôt dire **LE logiciel pour l'algorithmique** comme le conçoit nos programmes...
Il s'agit de **LARP,** logiciel gratuit mais pas libre, que l'on peut charger ici
http://www.marcolavoie.ca/larp/fr/default.htm
Je voudrais ici de suite rendre hommage à son créateur Marco LAVOIE, mais aussi à Bernard EGGER collègue de Marseille, spécialiste en son genre, et qui me l'a fait découvrir.
Des collègues de l'ENFA, responsables de la revue Py-Math, ont aussi produit cette présentation
http://r2math.enfa.fr/wp-content/uploads/2012/09/21-6-larp.pdf

Moi qui suis loin, mais alors très loin, d'être un passionné ou fana d'informatique, il ne m'a fallu que très peu de temps pour comprendre, et déjà organiser et faire tourner ne serait-ce que les algorithmes des sujets de Bac (boucle et test au programme...) ; car en effet,

1. non seulement on écrit à la main (disons "à la souris et clavier") l'algorithme en français sous la forme directe d'un ***organigramme*** classique (bulles entrée/sortie, bulles boucles de calcul, etc.)

2. mais on peut l'**exécuter** en direct-live !

3. et même mieux, voir évoluer le traitement "**pas à pas**" si on le souhaite ! (régler la vitesse et jouer avec les élèves à anticiper la procédure, les chemins pris, ...)

AUCUNE PERTURBATION (prise de tête) INUTILE, DUE À UN QUELCONQUE CHOIX ET APPRENTISSAGE D'UN LANGAGE MACHINE PARTICULIER

4. mais si l'on veut, en plus, il propose une traduction en ***pseudo-code*** qui se rapproche d'un "langage naturel" qui pourra être utilisé éventuellement dans un deuxième temps (sur calculatrice par exemple ;

Bref, j'ai trouvé cela génial pour une première initiation.
La manipulation est à la fois **ludique**, **pédagogique** et **facile à prendre en main**.
Pour l'instant donc il ne m'a pas fallu faire grande recherche...
Essayez, vous verrez !

Je suis donc pour l'instant à peu près persuadé que cela suffit amplement pour le travail en algorithmique demandé dans nos programmes aujourd'hui.
Que les profs de Seconde (entre autres) s'y attardent quelques instants, cela devrait vraiment capter leur intérêt !!!
Quelques limites sont à noter pour l'instant à première vue...si l'on veut faire plus que de l'algorithmique simple, et par exemple

- Pour les calculs de Statistique et de probabilités, quelles que soient les séries et les lois, ce n'est pas pratique ; plus précisément : aucun outil de description des séries statistiques. Il faut programmer tous les descripteurs même élémentaires (moyennes, variance, quantiles ...). Je n'ai pas encore exploré la possibilité (existe t-elle ?) de faire des sous programmes (modules auxiliaires) enregistrés séparément, de façon à monter une bibliothèque.
  Ce qui fait que pour explorer les séries résultats de simulations, c'est assez pénible.
  Pas de commande de tri d'une liste, ce qui est très gênant pour la détermination des statistiques de rang, mais très intéressant pour la recherche, la mise en œuvre et la comparaison de l'efficacité de différents algorithmes de tri... (voir *TriBulle1.larp*).
- Faire des graphiques : y a pas !
- Pas de factorielle, ni de combinaisons, ... (voir *ProbaBinoAB3Modules.larp*)
- Programmation séquentielle basique : ni objet, ni fonctionnelle.
- Le logiciel n'est plus développé.

C'est alors là que le prolongement avec **R** devient intéressant voire indispensable !

Mais, encore une fois, je prétends que l'on peut atteindre facilement l'objectif du programme lycée en algorithmique, avec LARP.

Mes premiers essais sur les SUITES...

ValeurDedn.larp : pour calculer le terme d'indice *n* d'une suite définie par récurrence (1re S)

Listeded.larp : pour générer une liste de termes d'une suite définie par récurrence (1re S)

Syracuse0.larp : fournit la durée de vol et l'altitude maximale d'un vol Syracuse (1re S)
→ Ici, le mode RECUR d'une calculatrice ne suffit plus (à cause du test conditionnel) : l'algorithme et le programme s'imposent ! C'est pour moi, l'exercice phare d'algorithmique

TDC.larp : donne le temps de doublement, triplement, ... d'un capital placé à Intérêts Composés (1re S)

Tous ces fichiers larp (extension .larp) contenant les "organigrammes exécutables" sont consignés dans le dossier **FichiersLarpManganelli.zip**, attaché.
Les 4 autres applications traitées (3 sujets de Bac et un exercice de livre), sont consignées dans **LarpSuiteManga.zip** (*CEJuin2013.larp*, *AGJuin2013.larp*, *AsieJuin2013.larp* et *Exo153p170.larp*, avec une version augmentée *Exo153p170.larp*).

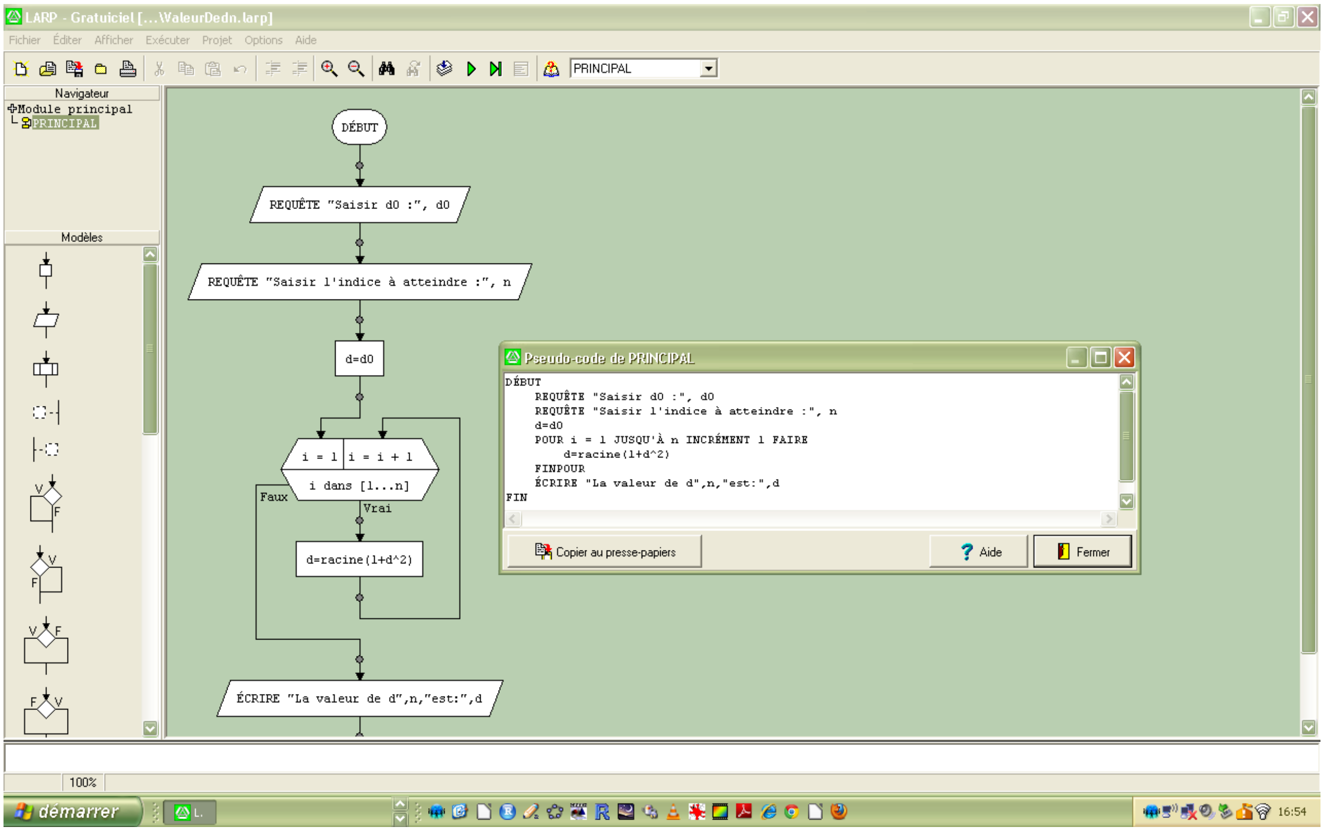
Dans le document **MathTice_3ImpEc.odt**, attaché, vous trouverez des copies d'écran d'exemples d'utilisation des "organigrammes exécutables" et de résultats obtenus. Pour des raisons pratiques il est possible que certains organigrammes ne figurent pas complètement sur les copies d'écran.

Pour construire ou modifier un "organigramme exécutable" avec LARP, il suffit de faire cliquer-glisser les modèles de gauche que l'on veut, de les double-cliquer pour les rendre actifs et de les compléter à souhait…

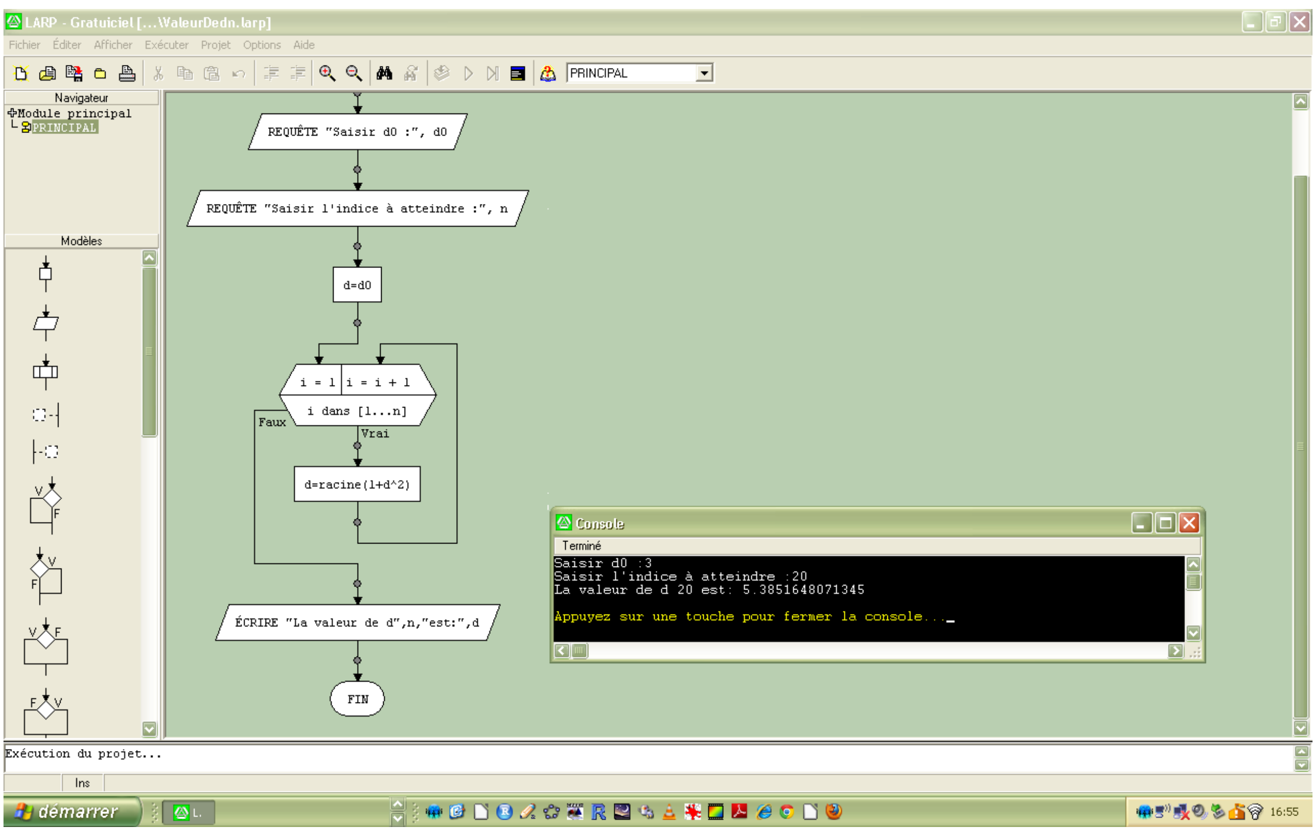
Un jeu d'enfant, c'est le cas de dire...

ValeurDedn.larp : pour calculer le terme d'indice _n_ d'une suite définie par récurrence (1re S)

Ici, il s'agit de la suite _d_, définie par d_0 et $d_{n+1}=\sqrt{1+d_n{}^2}$.

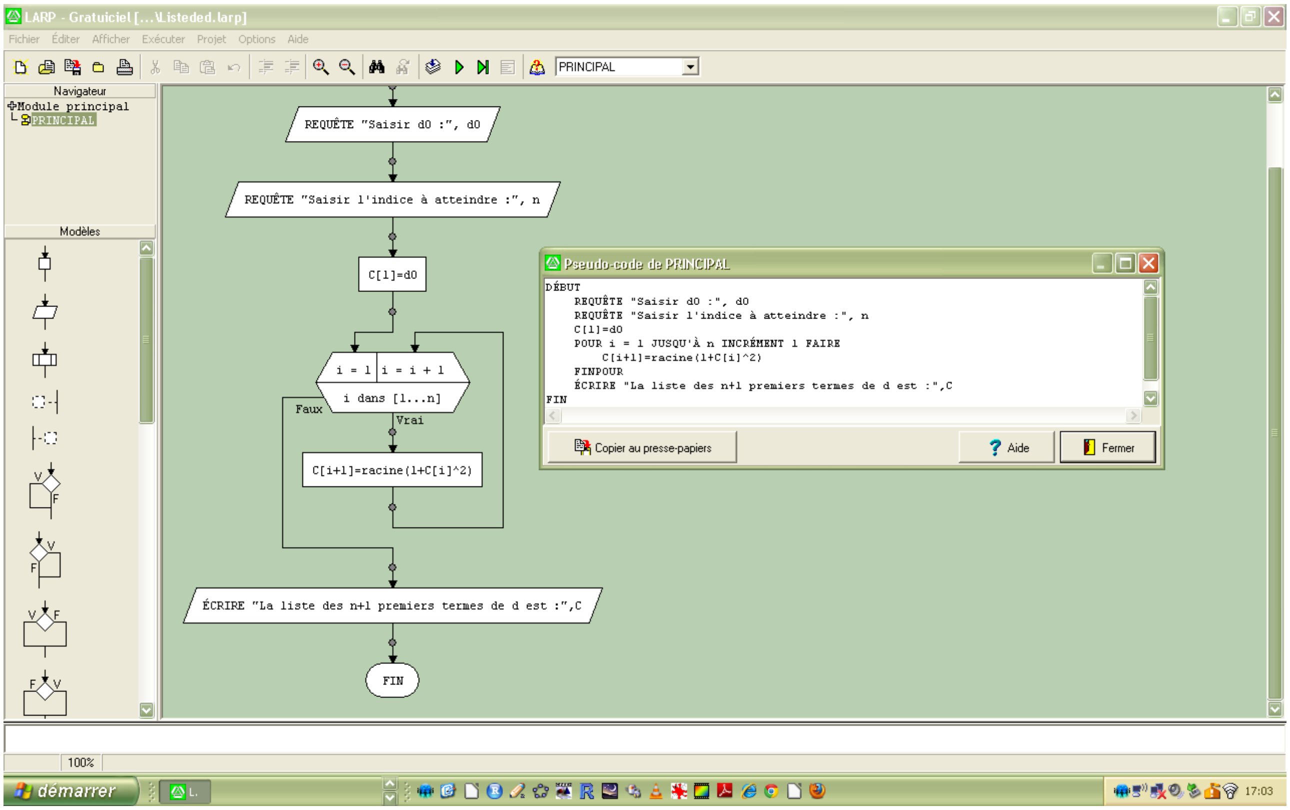


En rentrant $d_0=3$ et $n=20$, l'algorithme renvoie la valeur $d_{20}\approx 5{,}385$!

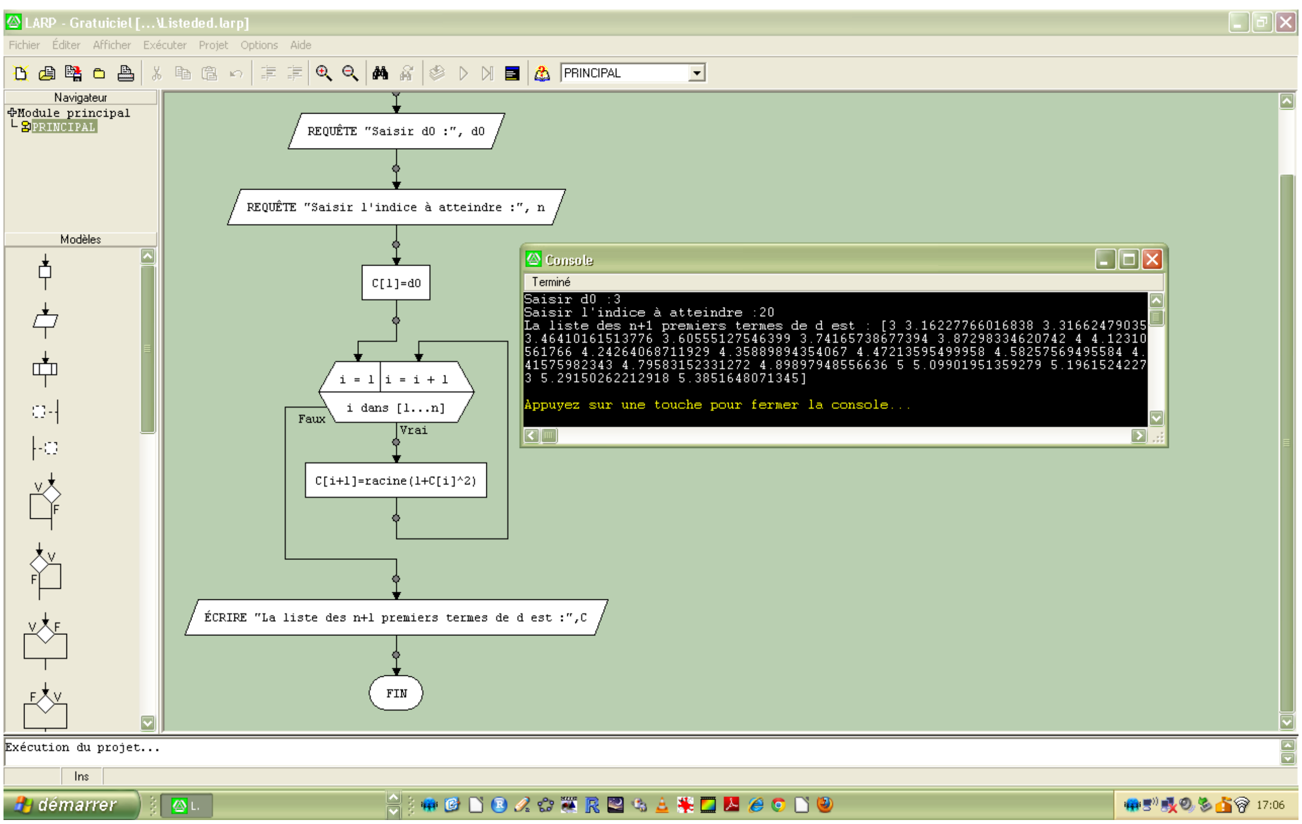


Listeded.larp **: pour générer une liste de termes d'une suite définie par récurrence (1re S)**

il s'agit encore de la suite *d*, définie par d_0 et $d_{n+1} = \sqrt{1+d_n{}^2}$.



En rentrant $d_0 = 3$ et $n = 20$, l'algorithme renvoie les 21 premières valeurs de *d* !



Syracuse0.larp : fournit la durée de vol et l'altitude maximale d'un vol Syracuse (1[re] S)

__La suite de Syracuse__

C'est Lothar Collatz, professeur à l'université de Hambourg, qui semble être le premier à s'être intéressé au problème dit de Syracuse[1] dans les années 30.

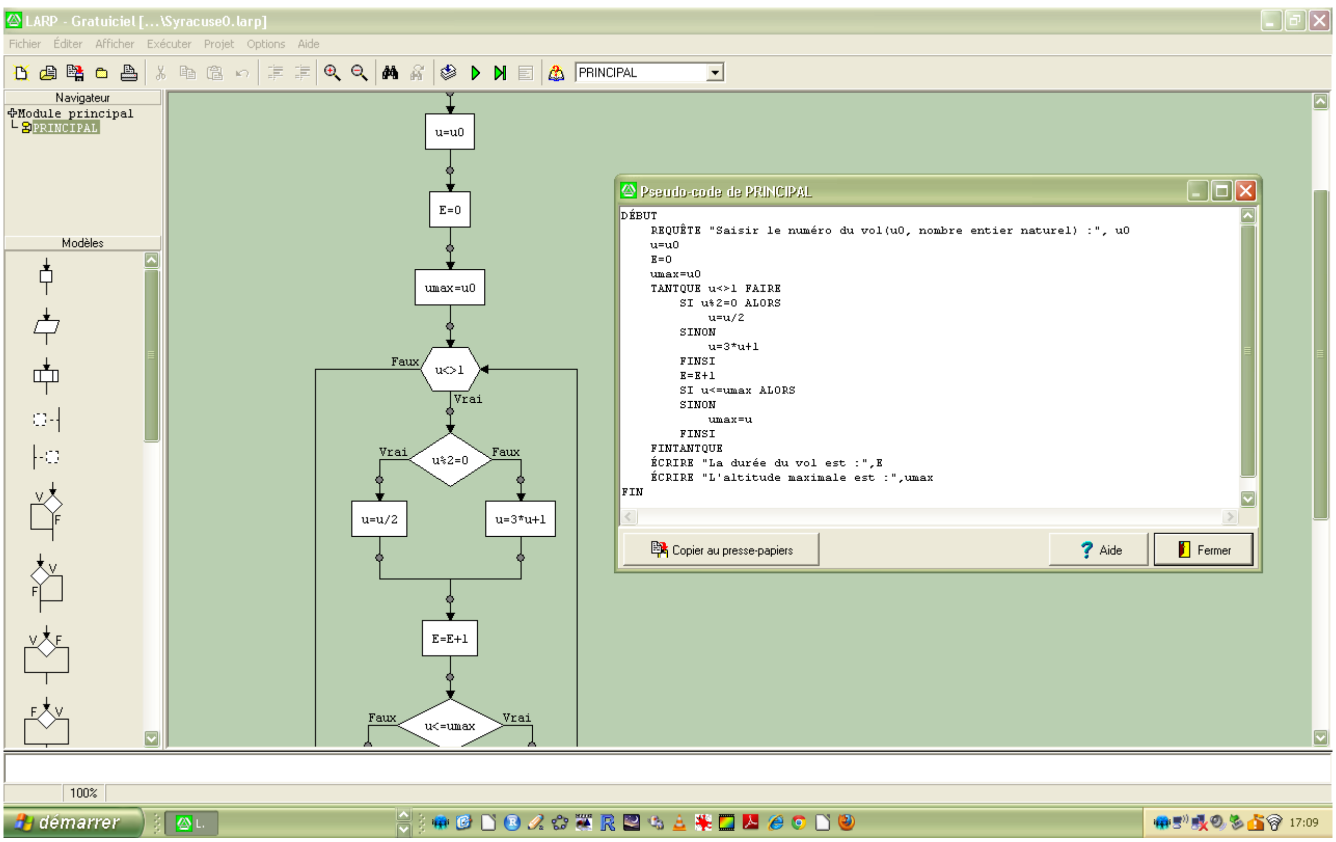
Il s'agit d'étudier le comportement (très spécial !) de cette suite définie par récurrence de la façon suivante :

u_0 **est un entier naturel non nul, et** $u_{n+1} = \begin{cases} \dfrac{u_n}{2} \text{ si } u_n \text{ est pair} \\ 3u_n + 1 \text{ si } u_n \text{ est impair} \end{cases}$.

Les mathématiciens **conjecturent** que pour n'importe quel entier naturel u_0 choisi, la suite de Syracuse finit par prendre la valeur 1 et tourner sur la boucle 4-2-1.Cette conjecture a été vérifiée pour les entiers naturels de 1 à 3.10^{12} (trois mille milliards !). Mais, à ce jour, **personne ne l'a encore démontrée dans le cas général**.

Un exercice intéressant peut consister alors à déterminer pour un u_1 ("numéro" de vol) donné,

- la "durée" du vol : l'indice d'arrivée (du premier terme valant 1)
- l'"altitude" maximale durant le vol : u_n maximal



Comme je l'ai dit plus haut, alors qu'un mode RECUR de calculatrice permet de résoudre les deux problèmes précédents, on est contraint ici, à cause du test conditionnel, d'écrire un algorithme de programmation de la suite. Voilà pourquoi je dis souvent que cette suite est pleinement pertinente en algorithmique.

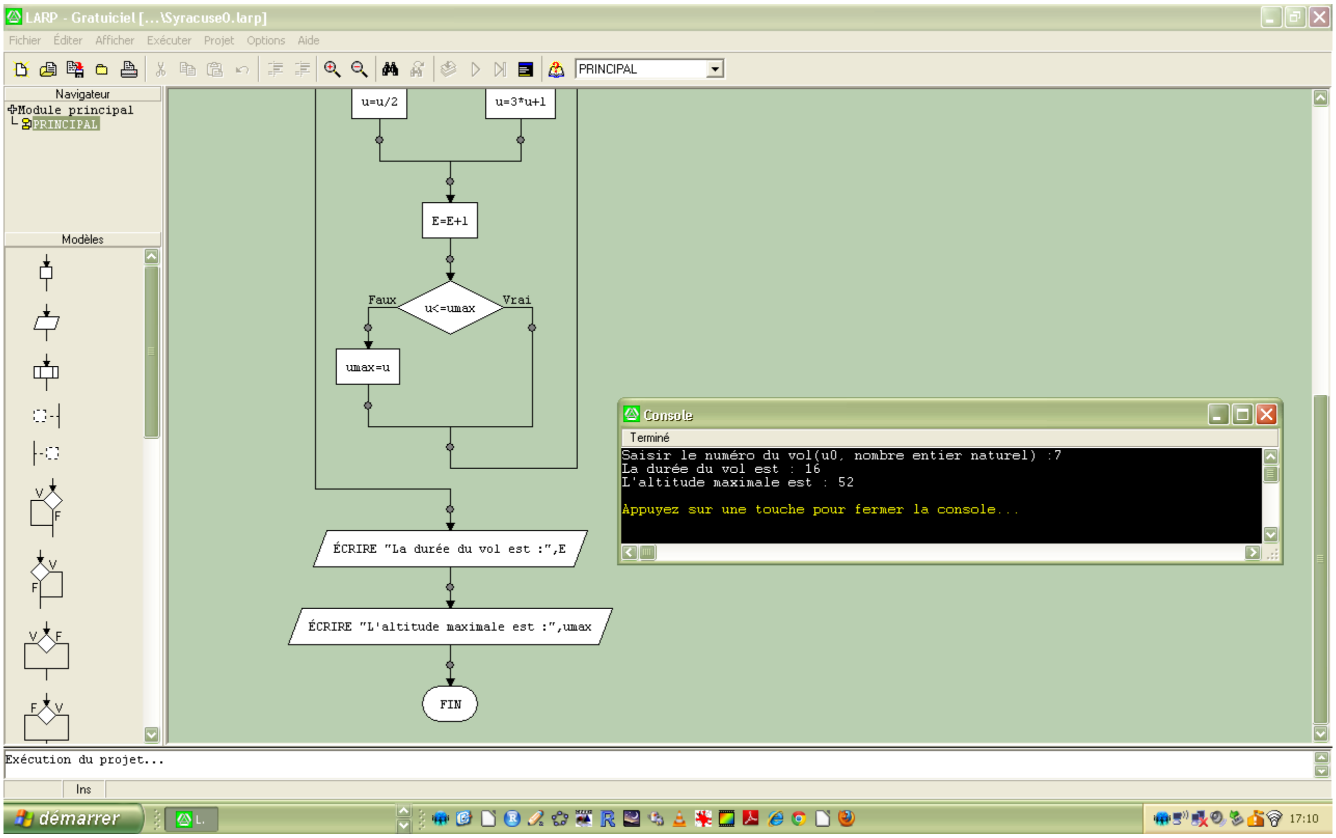
[1] Syracuse : nom de l'université de l'état de New York où Collatz présenta ce problème.

En rentrant $u_0 = 7$, l'algorithme renvoie :

- "durée" du vol : 16
- "altitude" maximale durant le vol : 52

On peut vérifier en effet la suite obtenue :

$$u_0 = 7 - 22 - 11 - 34 - 17 - 52 - 26 - 13 - 40 - 20 - 10 - 5 - 16 - 8 - 4 - 2 - u_{16} = 1$$



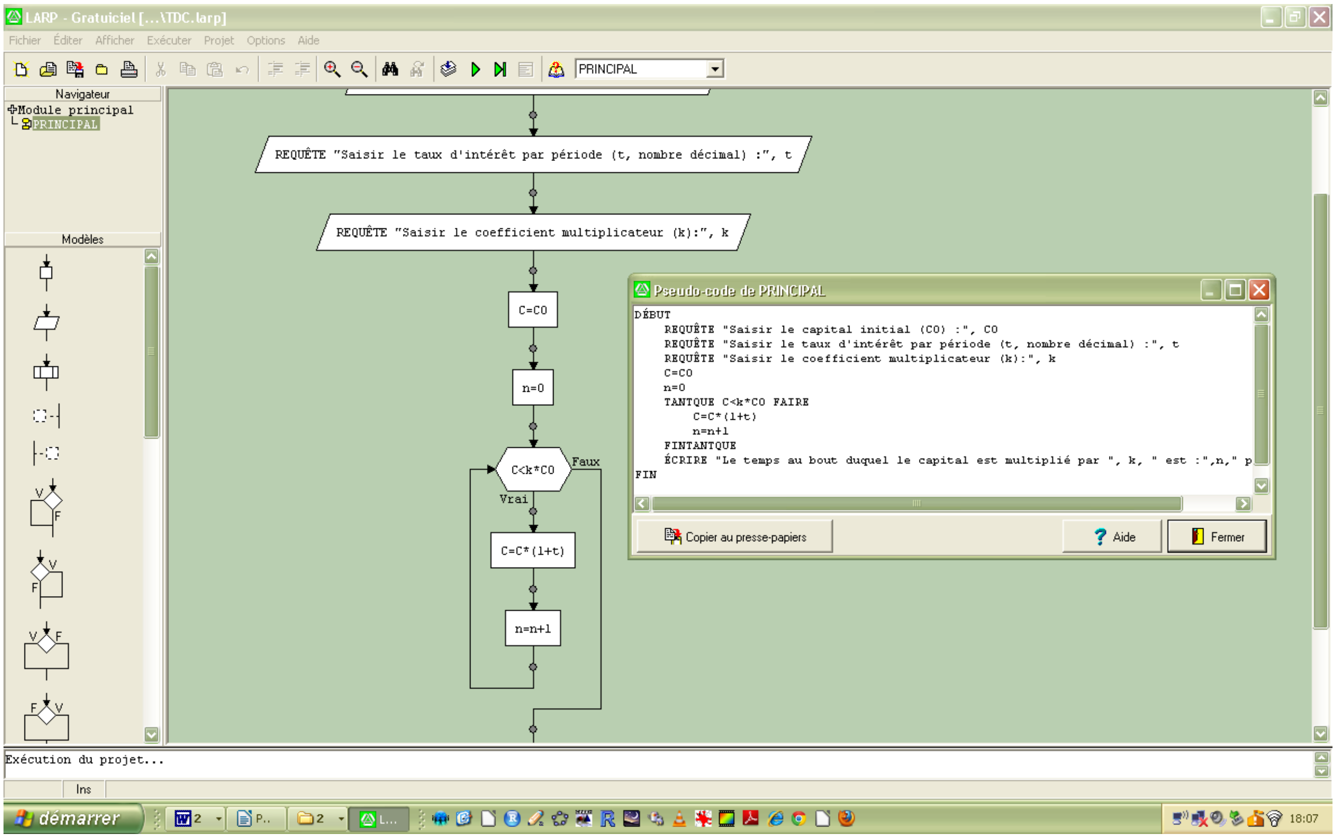
TDC.larp : donne le temps de doublement, triplement, ... d'un capital placé à Intérêts Composés (1re S)

Mise en place d'une boucle TANTQUE…

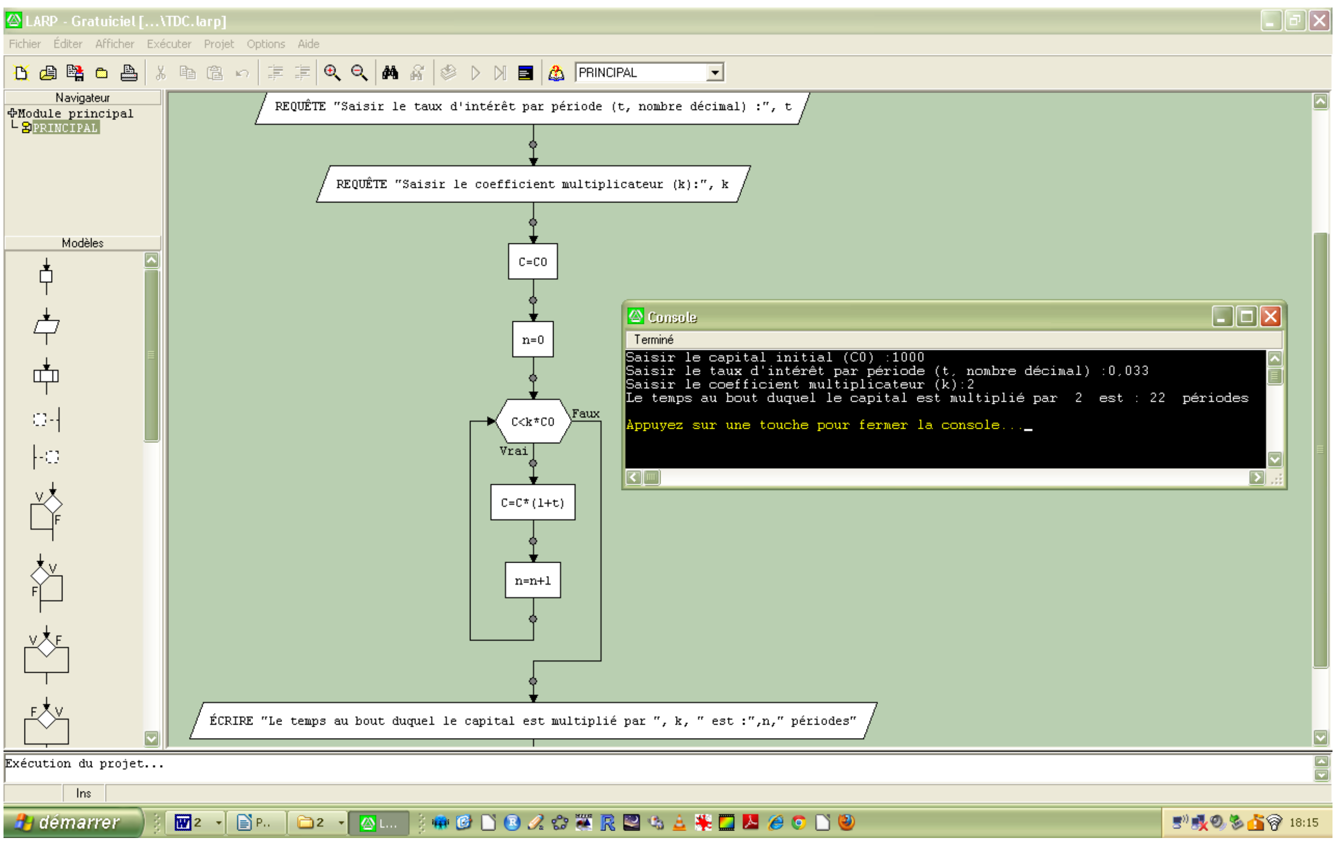
Bien sûr, on montrera aux élèves que le résultat ne dépend pas du Capital placé (C_0)…

et qu'ils peuvent trouver « empiriquement » le résultat en faisant tourner la suite géométrique définie par $(1+t)^n$ en jeu.

On pourra aussi montrer aux élèves que le problème se ramène à une inéquation dont l'inconnue figure en exposant et que la fonction logarithme népérien permettra de résoudre en Tle.



En rentrant $C_0=1000$, $t=0{,}033$ et $k=2$, l'algorithme renvoie 22 (périodes pour le doublement) !



Et maintenant quelques sujets de BAC que je me suis amusé à traiter à leur sortie…

Centres étrangers, Juin 2013

L'objet de cet exercice est l'étude de la suite (u_n) définie par son premier terme $u_1 = \frac{3}{2}$ et la relation de récurrence : $u_{n+1} = \frac{nu_n + 1}{2(n+1)}$.

Partie A – Algorithmique et conjectures

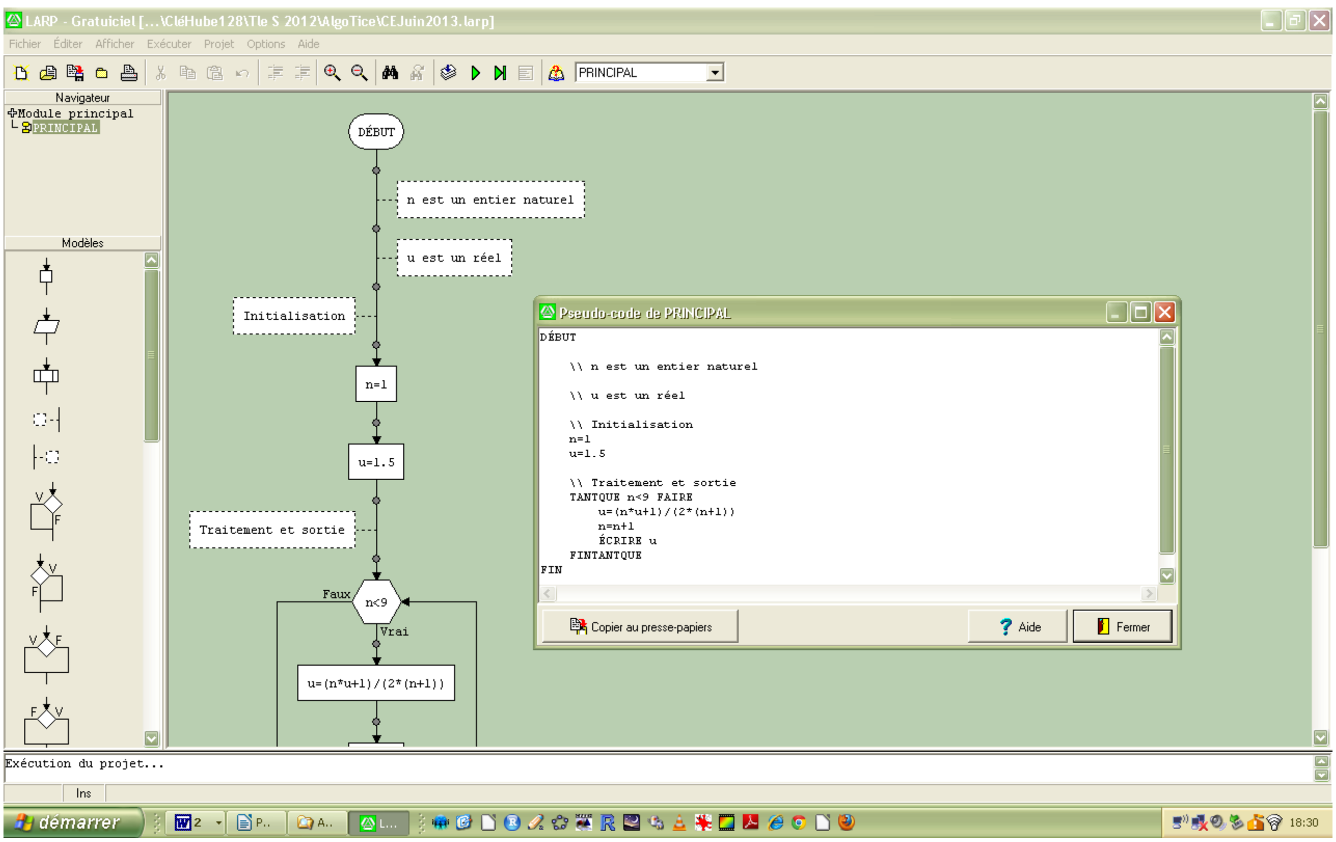
Pour calculer et afficher le terme u_9 de la suite, un élève propose l'algorithme ci-contre.

Il a oublié de compléter deux lignes.

Variables	n est un entier naturel u est un réel
Initialisation	Affecter à n la valeur 1 Affecter à u la valeur 1,5
Traitement	Tant que $n < 9$ Affecter à u la valeur ….. Affecter à n la valeur ….. Fin Tant que
Sortie	Afficher la variable u

1. Recopier et compléter les deux lignes de l'algorithme où figurent des points de suspension.
2. Comment faudrait-il modifier cet algorithme pour qu'il calcule et affiche tous les termes de la suite de u_2 jusqu'à u_9 ?

L'algorithme sous LARP… (Boucle TANTQUE)



et la console d'exécution qui fournit le résultat escompté !

Asie, Juin 2013

On considère la suite (u_n) définie par : $u_0 = 2$ et, pour tout entier nature n :

$$u_{n+1} = \frac{1+0,5u_n}{0,5+u_n}.$$

On admet que tous les termes de cette suite sont définis et strictement positifs.

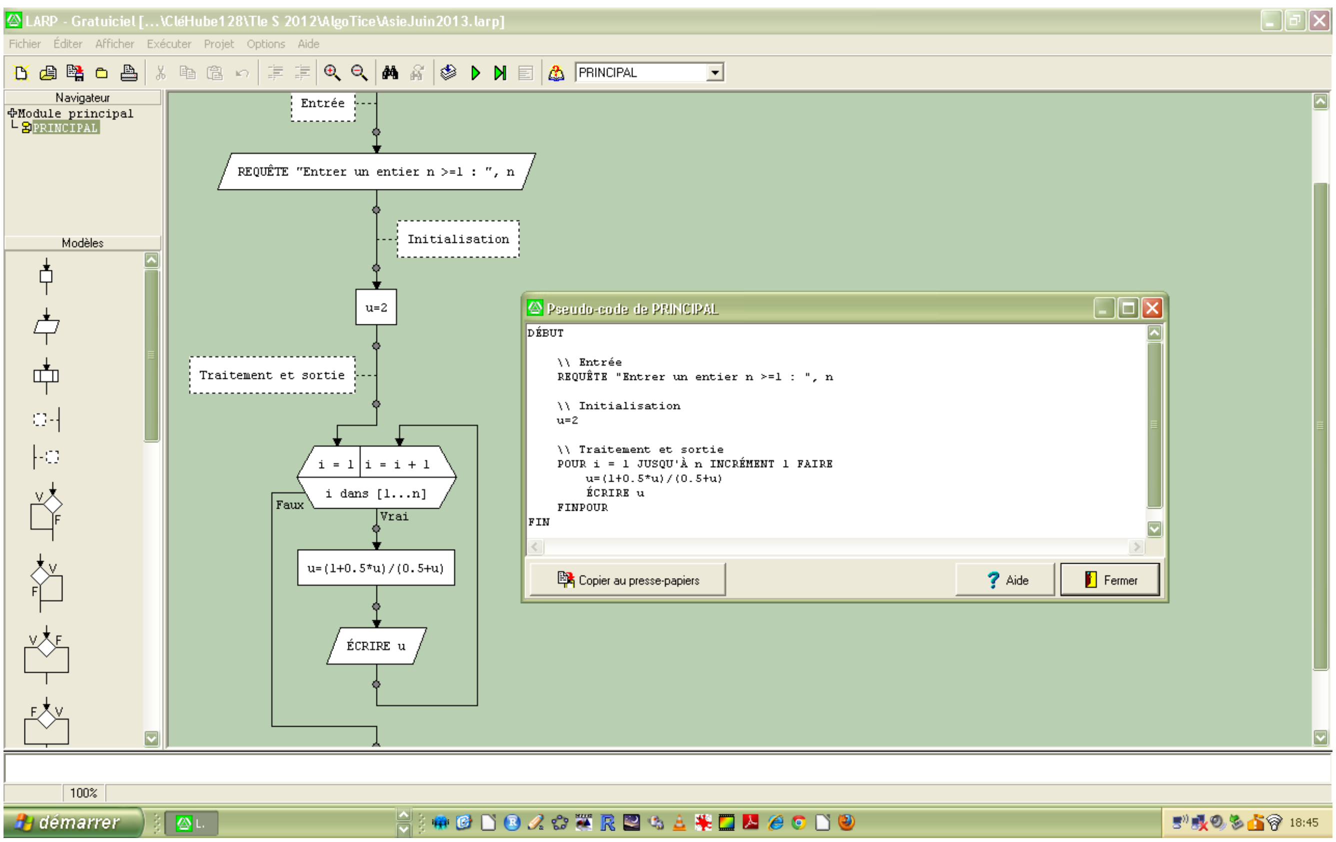
1. On considère l'algorithme suivant :

Entrée	Soit un entier naturel non nul n
Initialisation	Affecter à u la valeur 2
Traitement et sortie	POUR i allant de 1 à n Affecter à u la valeur $\dfrac{1+0,5u}{0,5+u}$ Afficher u
	FIN POUR

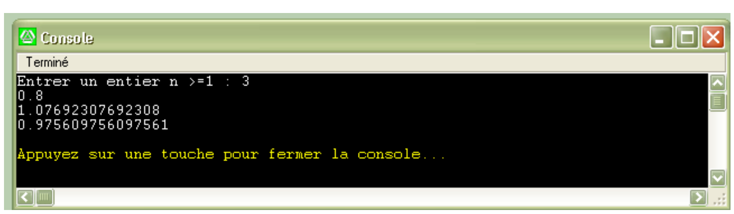
Reproduire et compléter le tableau suivant, en faisant fonctionner cet algorithme pour $n = 3$. Les valeurs de u seront arrondies au millième.

i	1	2	3
u			

L'algorithme sous LARP… (Boucle POUR)



et la console d'exécution qui fournit le résultat escompté !



On considère la suite (z_n) à termes complexes définie par : $z_0 = 1 + \mathrm{i}$ et, pour tout entier naturel n, par

$$z_{n+1} = \frac{z_n + |z_n|}{3}.$$

Pour tout entier naturel n, on pose : $z_n = a_n + \mathrm{i}b_n$, où a_n est la partie réelle de z_n et b_n est la partie imaginaire de z_n.

Le but de cet exercice est d'étudier la convergence des suites (a_n) et (b_n).

Partie A

1. Donner a_0 et b_0.

2. Calculer z_1, puis en déduire que $a_1 = \frac{1+\sqrt{2}}{3}$ et $b_1 = \frac{1}{3}$.

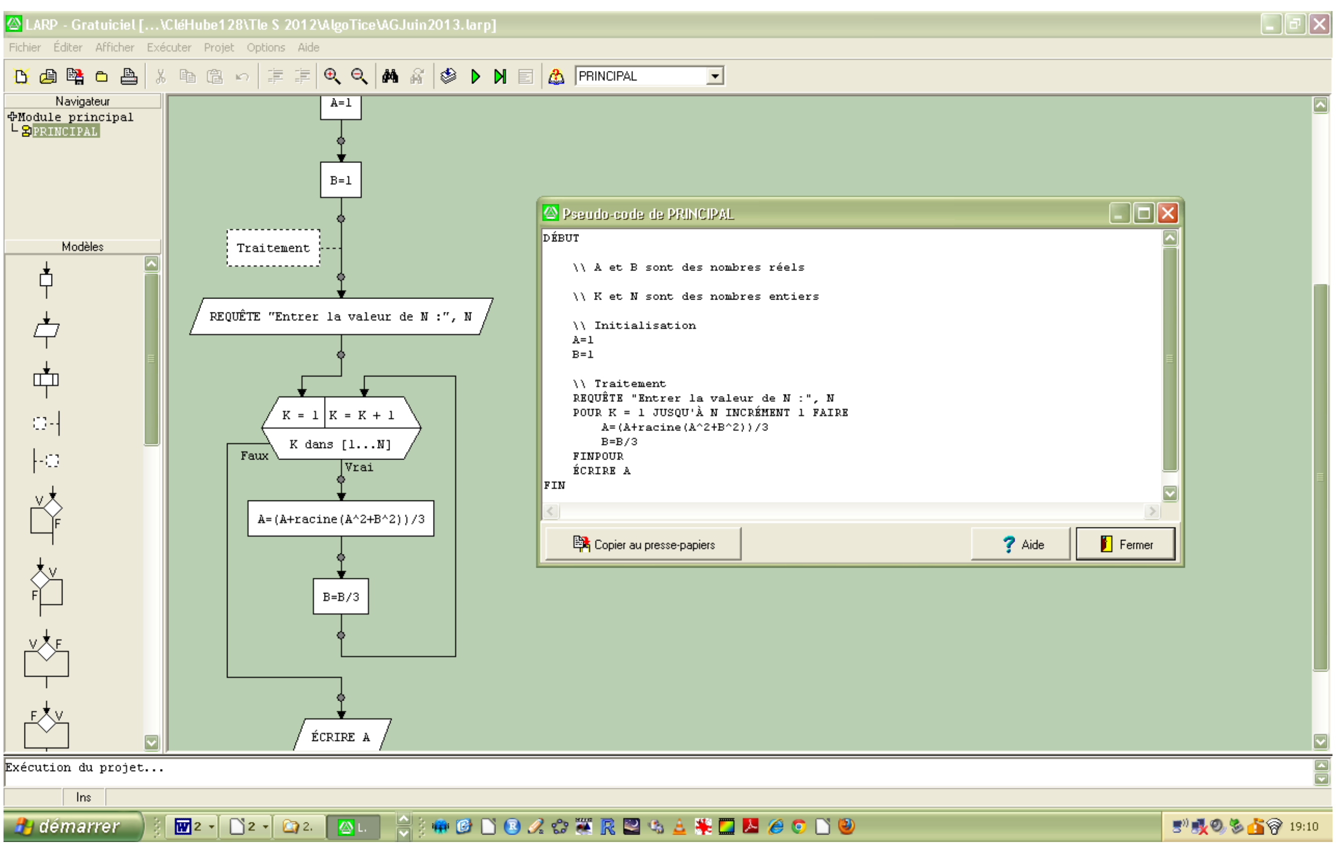
3. On considère l'algorithme suivant :

```
Variables :        A et B des nombres réels
                   K et N des nombres entiers
Initialisation :   Affecter à A la valeur 1
                   Affecter à B la valeur 1
Traitement :
Entrer la valeur de N
Pour K  variant de 1 à N
        Affecter à A la valeur (A+√(A²+B²))/3
        Affecter à B la valeur B/3
FinPour
Afficher A
```

a. On exécute cet algorithme en saisissant $N = 2$. Recopier et compléter le tableau ci-dessous contenant l'état des variables au cours de l'exécution de l'algorithme (on arrondira les valeurs calculées à 10^{-4} près).

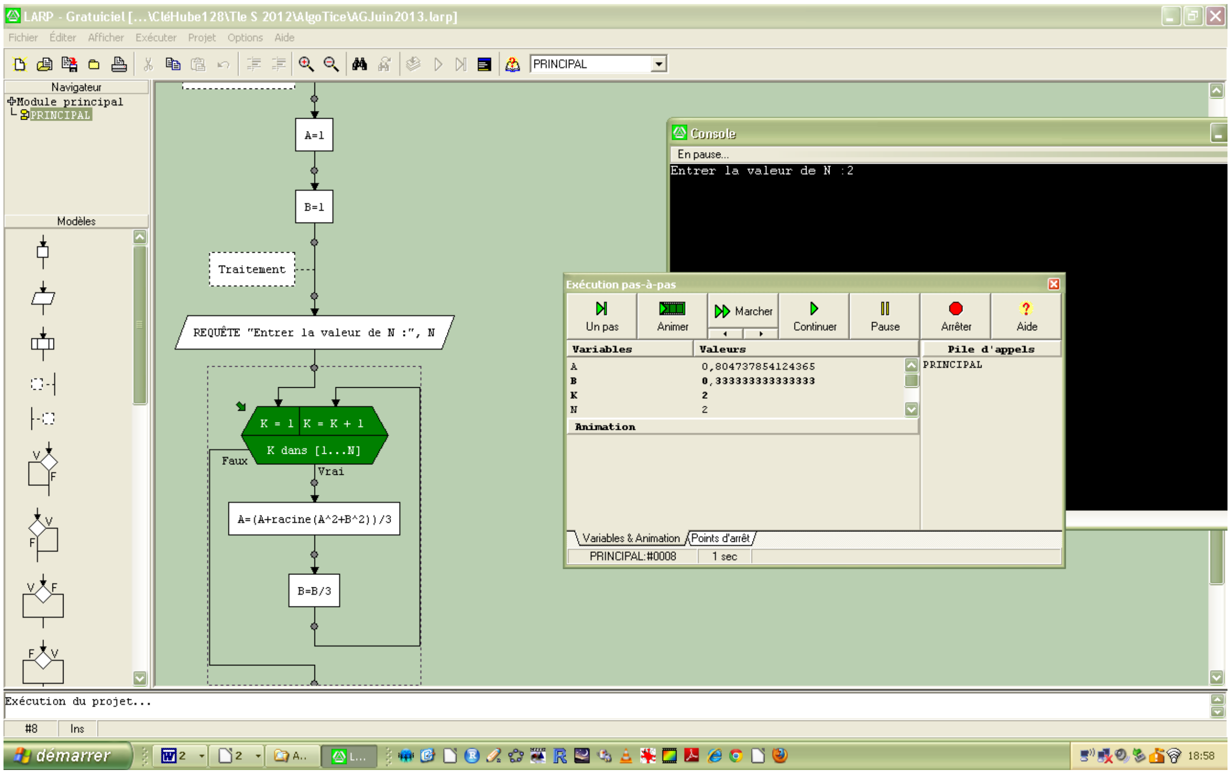
K	A	B
1		
2		

L'algorithme sous LARP… (Boucle POUR)

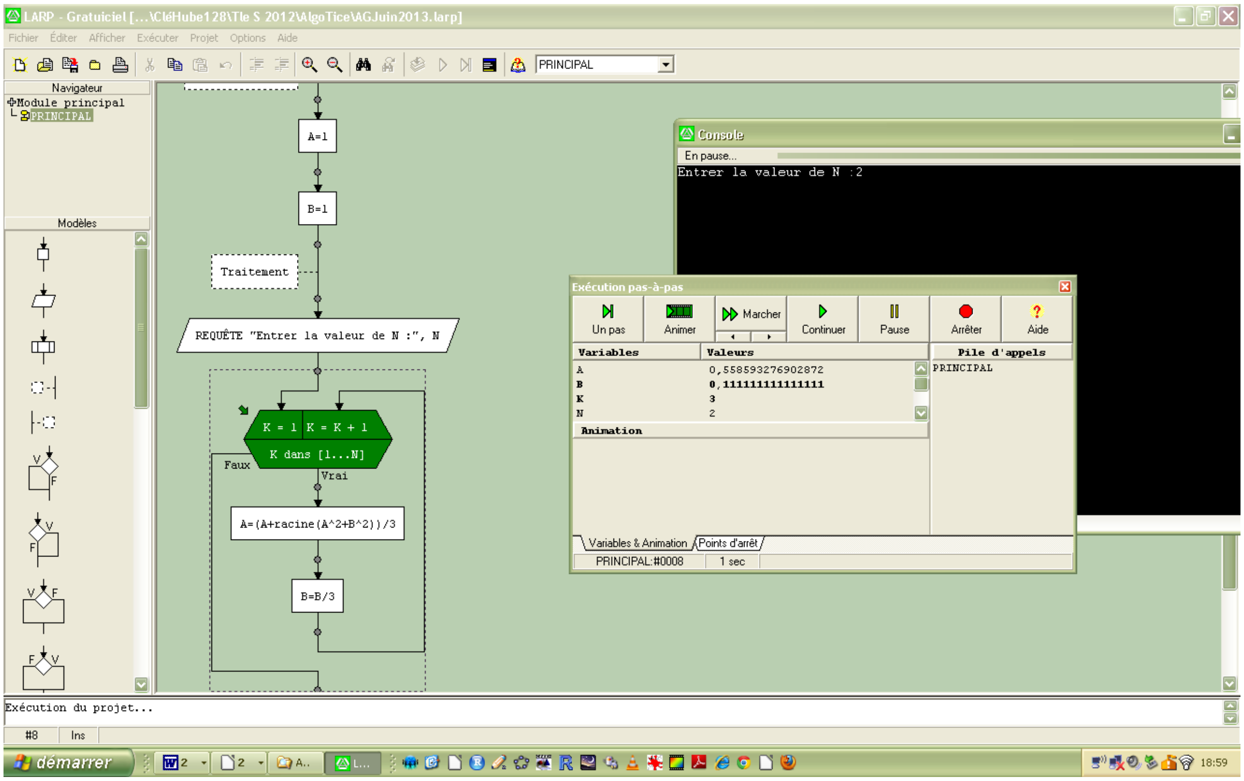


Avec la console d'exécution pas-à-pas :

K vient de passer à 2 ; A1 et B1 sont affichés !



Ci-dessous, K vient de passer à 3 ; A2 et B2 sont affichés !



Pour finir, je propose la résolution (du moins dans sa partie algorithmique), d'un exercice de Terminale S, mettant en jeu une suite définie par sommation – Nathan, *Hyperbole* 2012, 153 p. 170.

ÉNONCÉ

153 Algorithmique

1. Donner une valeur approchée de ln 5 à l'aide de la calculatrice.

2. Voici un algorithme.

> **Entrée**
> Saisir N (N nombre entier naturel non nul)
> **Initialisation**
> S prend la valeur 0
> **Traitement**
> Pour k de N jusqu'à 5N
> | S prend la valeur S + $\frac{1}{k}$
> FinPour
> **Sortie**
> Afficher S

a) Exprimer S en fonction de N.
b) Coder l'algorithme dans un langage et saisir le programme à l'ordinateur.
c) Tester le programme obtenu pour les valeurs de N égales à 1000, 10 000 et 100 000.

1. La calculatrice (Casio Graph 35) affiche **1,609437912**

$$\ln 5 \approx 1{,}61 \text{ à } 10^{-2} \text{ près.}$$

2. a) S en fonction de N.
On a le tableau d'avancement suivant :

k	S
$\times$	0
N	$\frac{1}{N}$
$N+1$	$\frac{1}{N}+\frac{1}{N+1}$
$N+2$	$\frac{1}{N}+\frac{1}{N+1}+\frac{1}{N+2}$
$\vdots$	$\vdots$
$5N-1$	$\frac{1}{N}+\frac{1}{N+1}+\ldots+\frac{1}{5N-1}$
$5N$	$\frac{1}{N}+\frac{1}{N+1}+\ldots+\frac{1}{5N-1}+\frac{1}{5N}$

D'où : $S=\frac{1}{N}+\frac{1}{N+1}+\ldots+\frac{1}{5N-1}+\frac{1}{5N}=\sum_{k=N}^{5N}\frac{1}{k}$.

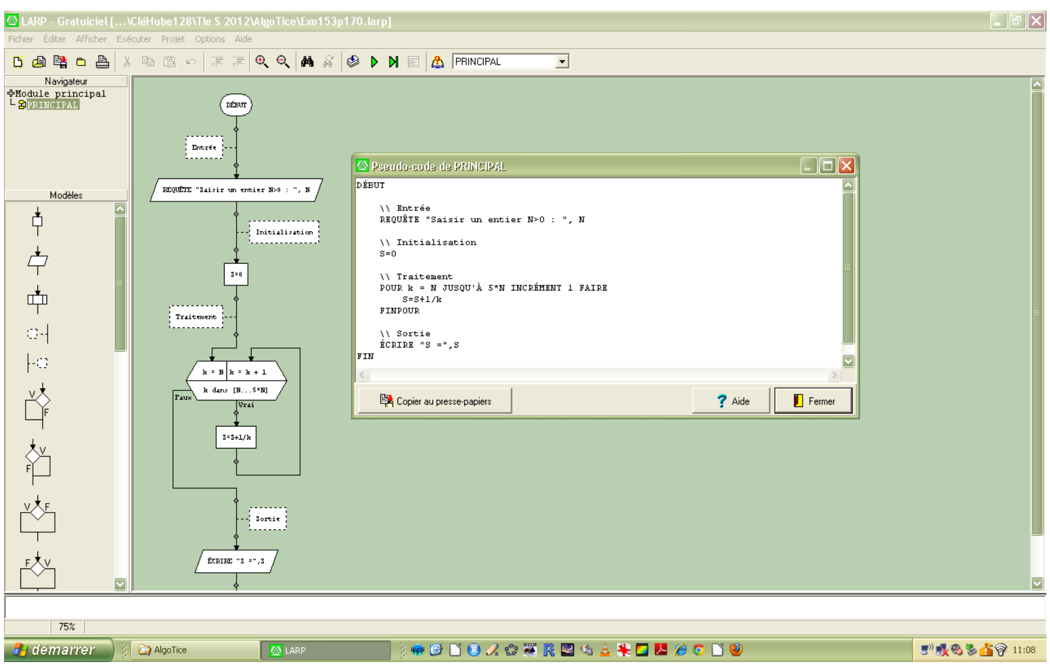
b) Codage machine

Le plus accessible est le codage en langage calculatrice… mais les capacités restreintes de celle-ci ne vont pas permettre « d'aller très loin »…

CASIO Graph 35	TI-83 Plus
"SAISIR N :"?→N↵	:Input "SAISIR N :",N
0→S↵	:0→S
For N→K To 5N↵	:For (K,N,5N)
S+(1÷K)→S↵	:S+(1/K)→S
Next↵	:End
S◢	:Disp S

Ci-dessous, le pseudo-code de **LARP**, puis une impression d'écran avec l'organigramme exécutable qui se construit comme un jeu d'enfants…

```
DÉBUT
   \\ Entrée
   REQUÊTE "Saisir un entier N>0 : ", N
   \\ Initialisation
   S=0
   \\ Traitement
   POUR k = N JUSQU'À 5*N INCRÉMENT 1 FAIRE
      S=S+1/k
   FINPOUR
   \\ Sortie
   ÉCRIRE "S =",S
FIN
```



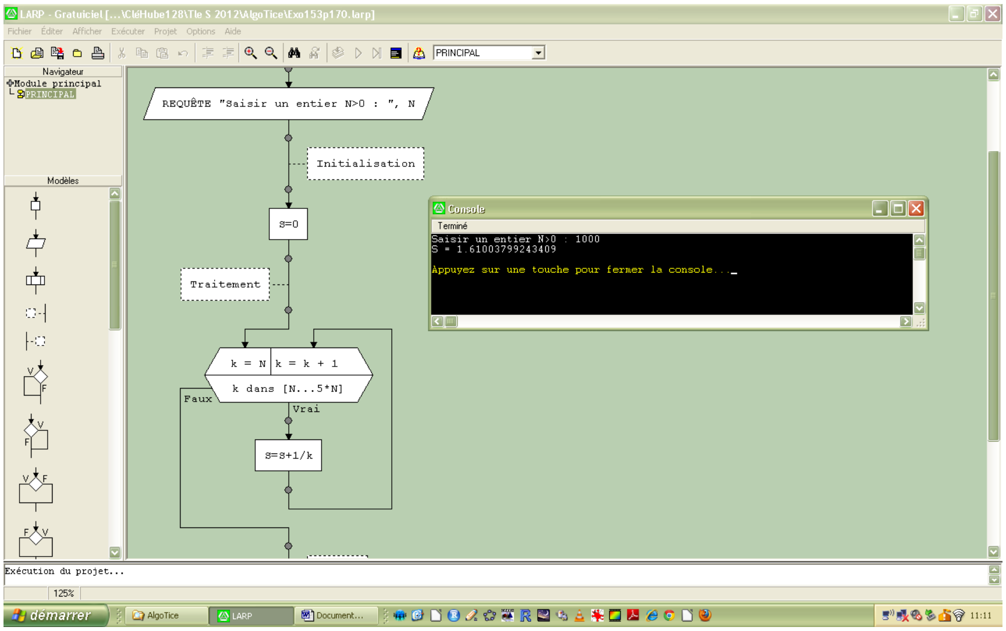
c) Exécution de l'algorithme…

Comme prévu précédemment, la calculatrice est en galère pour des valeurs de N trop grandes…

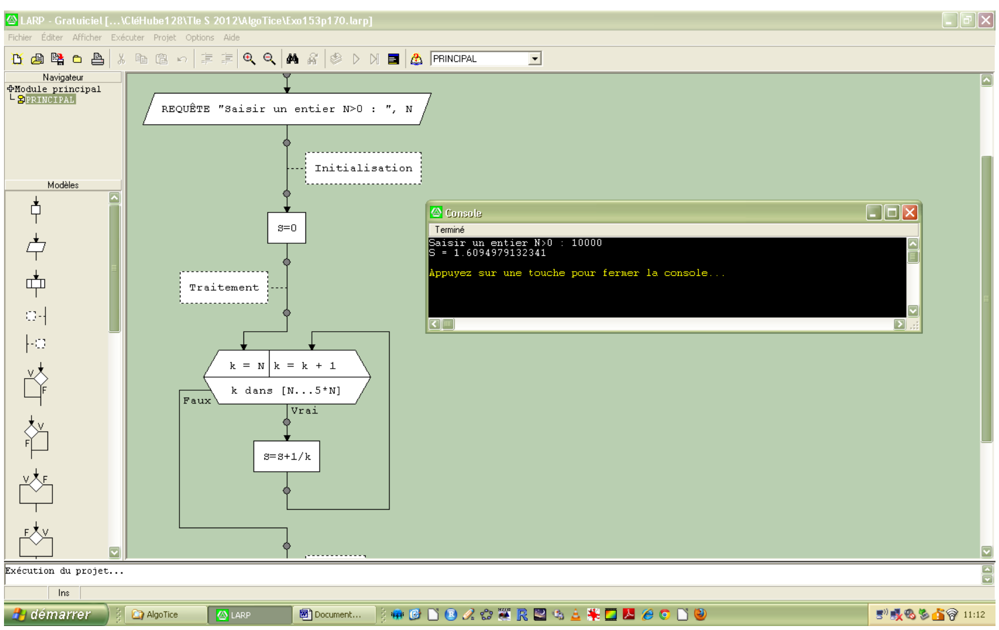
CASIO Graph 35	TI-83 Plus
Pour N=1000, au bout de 1 min 23 s, la calculatrice affiche **1.6**10037992 Pour N=10000, au bout de 14 min 16 s, la calculatrice affiche **1.6094**97913	Pour N=1000, au bout de 1 min 31 s, la calculatrice affiche **1.6**10037992 Pour N=10000, au bout de 14 min 36 s, la calculatrice affiche **1.6094**97913
Pour N=100000, au bout de 2h15 (environ), la calculatrice affiche **1.6094**43911	Pour N=100000, au bout de 2h15 (environ), la calculatrice affiche **1.6094**43912

Voici maintenant ce que donne l'exécution de l'algorithme sous LARP :

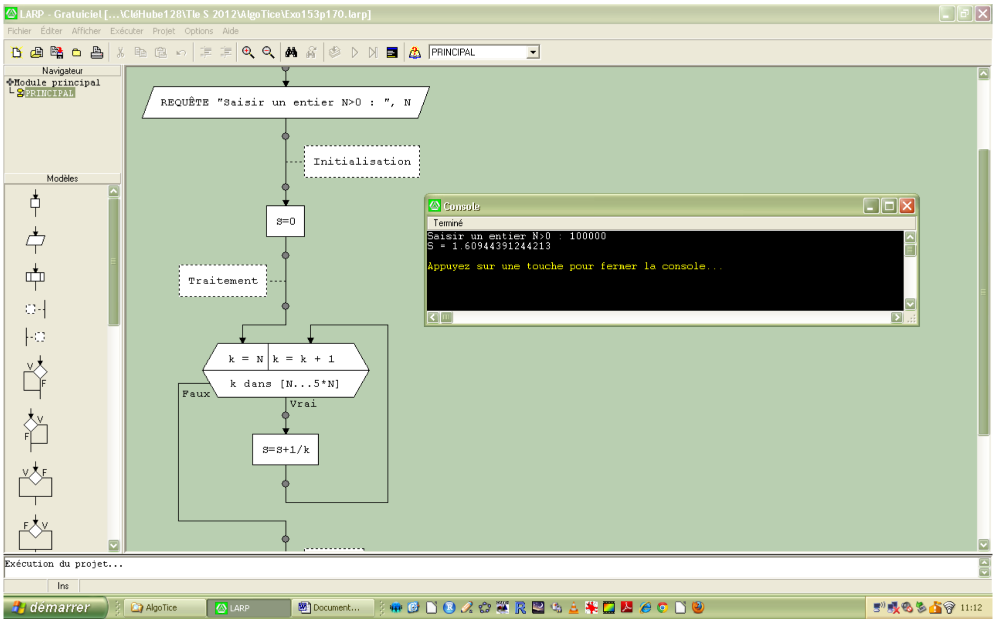
N=1000 ; Affichage : S = **1,6**1003799243409



N=10000 ; Affichage : S = **1,6094**979132341



N=100000 ; Affichage : S = **1,60944**391244213



<u>Remarque</u> : Je teste N=1000000 ; au bout d'1 min 05 s, la console affiche **1,60943**851243431
Et je teste N=10000000 (oui c'est ça, 10 millions !) ; au bout de 11 min 16 s, la console affiche **1,6094379**7243341

Le gros avantage ici, c'est que LARP est un langage COMPILÉ, ce qui en rend son exécution relativement rapide. Pour info, j'ai mis à l'épreuve l'ami Hub'***R***[2]… et le résultat est, pour une fois, sans appel : c'est moi qui gagne !

d) S semble se rapprocher de $\ln 5$…

☞ **Au risque de me répéter…**

On peut aller encore un peu plus loin avec ce logiciel, mais pour se placer déjà au niveau requis dans l'évaluation actuelle en Lycée, je le trouve très suffisant, et surtout, extrêmement simple d'utilisation, très intéressant sur le plan pédagogique et bien dans l'esprit des programmes (certes les organigrammes ont été un peu boudés mais qu'est-ce que leur caractère visuel peut être intéressant !!!).
C'est pour moi un logiciel (le logiciel par excellence peut-être ?) très intéressant pour la sensibilisation, l'initiation, l'apprentissage de l'algorithmique.
Comme je l'ai déjà signalé plus haut, par la suite, on atteint certes ses limites lorsque les choses se compliquent un peu, et il n'a pas la palette d'utilisation et d'application d'un ***R***, par exemple.

À vos LARP, prêts, partez !
Régalez-vous !
J'attends bien évidemment vos premières sensations... (critiques)
Certains d'entre vous connaissent peut-être... ?
À suivre…

Stéphan, tireur à LARP.

[2] Logiciel (et langage) de prédilection de mon ami Hubert RAYMONDAUD… et de bien d'autres !!!