

BORDEAUX 2018

**PROGRAMMER EN CYCLE 4
DANS DES SITUATIONS DE LA VIE
DE TOUS LES JOURS**

Thierry CHEVALARIAS
Professeur au collège de Jaunay-Marigny
Membre de l'IREM de POITIERS

PLAN :

1. Programmes officiels
2. Définitions d'algorithme
3. Notre but
4. Liste des différentes tâches
5. Exemples de situations
6. Mise en œuvre
7. Liste des différentes pédagogies
8. Que doivent retenir les élèves ?
9. La question de la pertinence des outils
10. Les évaluations

PROGRAMMES CYCLE 4

Mathématiques

Le programme de mathématiques est rédigé pour l'ensemble du cycle. Les connaissances et compétences visées sont des attendus de la fin du cycle. Pour y parvenir, elles devront être travaillées de manière progressive et réinvesties sur toute la durée du cycle. Des repères de progressivité indiquent en particulier quelles notions ne doivent pas être introduites dès le début du cycle, mais seulement après que d'autres notions ont été rencontrées, puis stabilisées.

Ce programme est ancré dans les cinq domaines du socle et il est structuré selon les quatre thèmes classiques : nombres et calculs ; organisation et gestion de données, fonctions ; grandeurs et mesures ; espace et géométrie. **En outre, un enseignement de l'informatique est dispensé conjointement en mathématiques et en technologie.** Ces domaines du socle et ces thèmes du programme ne sont évidemment pas étanches.

La mise en œuvre du programme doit permettre de développer les six compétences majeures de l'activité mathématique : **chercher, modéliser, représenter, raisonner, calculer, communiquer**, qui sont détaillées dans le tableau ci-après.

Pour ce faire, **une place importante doit être accordée à la résolution de problèmes**, qu'ils soient internes aux mathématiques ou liés à **des situations issues de la vie quotidienne** ou d'autres disciplines. Le programme fournit des outils permettant de modéliser des situations variées sous forme de problèmes mathématisés.

Au cycle 4, l'élève développe son intuition en passant d'un mode de représentation à un autre : numérique, graphique, algébrique, géométrique, etc. Ces changements de registre sont favorisés par l'usage de logiciels polyvalents tels que le tableur ou les logiciels de géométrie dynamique. L'utilisation du tableur et de la calculatrice est nécessaire pour gérer des données réelles et permet d'inscrire l'activité mathématique dans les domaines 3, 4 et 5 du socle.

L'enseignement de l'informatique au cycle 4 n'a pas pour objectif de former des élèves experts, mais de leur apporter des clés de décryptage d'un monde numérique en évolution constante. Il permet d'acquérir des méthodes qui construisent la pensée algorithmique et développe des compétences dans la représentation de l'information et de son traitement, la résolution de problèmes, le contrôle des résultats. Il est également l'occasion de mettre en place des modalités d'enseignement fondées sur une pédagogie de projet, active et collaborative. Pour donner du sens aux apprentissages et valoriser le travail des élèves, cet enseignement doit se traduire par la réalisation de productions collectives (programme, application, animation, sites, etc.) dans le cadre d'activités de création numérique, au cours desquelles les élèves développent leur autonomie, mais aussi le sens du travail collaboratif.

Thème E - Algorithmique et programmation

Au cycle 4, les élèves s'initient à la programmation en développant dans une démarche de projet quelques programmes simples, sans viser une connaissance experte et exhaustive d'un langage ou d'un logiciel particulier. En créant un programme, ils développent des méthodes de programmation, revisitent les notions de variables et de fonctions sous une forme différente, et s'entraînent au raisonnement.

Attendus de fin de cycle

Écrire, mettre au point et exécuter un programme simple

Connaissances et compétences associées	Exemples de situations, d'activités et de ressources pour l'élève
<p>Décomposer un problème en sous-problèmes afin de structurer un programme reconnaître des schémas.</p> <p>Écrire, mettre au point (tester, corriger) et exécuter un programme en réponse à un problème donné.</p> <p>Écrire un programme dans lequel des actions sont déclenchées par des événements extérieurs.</p> <p>Programmer des scripts se déroulant en parallèle.</p> <ul style="list-style-type: none">» Notions d'algorithmique et de programme.» Notion de variable informatique.» Déclenchement d'une action par un événement, séquences d'instructions, boucles, instructions conditionnelles.	<p>Jeux dans un labyrinthe, jeu de Pong, bataille navale, jeu de nim, tic tac toe.</p> <p>Réalisation de figure à l'aide d'un logiciel de programmation pour consolider les notions de longueur et d'angle.</p> <p>Initiation au chiffrement (Morse, chiffre de César, code ASCII...).</p> <p>Construction de tables de conjugaison, de pluriels, jeu du cadavre exquis...</p> <p>Calculs simples de calendrier.</p> <p>Calculs de répertoire (recherche, recherche inversée...).</p> <p>Calculs de fréquences d'apparition de chaque lettre dans un texte pour distinguer sa langue d'origine : français, anglais, italien, etc.</p>

Repères de progressivité

En 5^e, les élèves s'initient à la programmation événementielle. Progressivement, ils développent de nouvelles compétences, en programmant des actions en parallèle, en utilisant la notion de variable informatique, en découvrant les boucles et les instructions conditionnelles qui complètent les structures de contrôle liées aux événements.

La formation au **raisonnement** et l'initiation à la **démonstration** sont des objectifs essentiels du cycle 4. Le raisonnement, au cœur de l'activité mathématique, doit prendre appui sur des situations variées (par exemple problèmes de nature arithmétique ou géométrique, mais également mise au point d'un **programme** qui doit tourner sur un ordinateur ou pratique de jeux pour lesquels il faut développer une stratégie gagnante, individuelle ou collective, ou maximiser ses chances).

L'utilisation d'outils comme le tableur, la calculatrice, un logiciel de géométrie dynamique ou de **programmation** permet de gérer des données réelles ou expérimentales, de faire des représentations et des simulations, de programmer des objets techniques et d'inscrire l'activité mathématique dans les domaines 4 et 5 du socle.

■ Thème E – Algorithmique et programmation

Au cycle 4, les élèves s'initient à la programmation, en développant dans une démarche de projet quelques programmes simples, sans viser une connaissance experte et exhaustive d'un langage ou d'un logiciel particulier. En créant un programme, ils développent des méthodes de programmation, revisitent les notions de variables et de fonctions sous une forme différente, et s'entraînent au raisonnement.

Exemples d'activités possibles : jeux dans un labyrinthe, jeu de Pong, bataille navale, jeu de nim, tic tac toe, jeu du cadavre exquis.

Attendus de fin de cycle

- Écrire, mettre au point et exécuter un programme simple.

Écrire, mettre au point, exécuter un programme

Connaissances

- Notions d'algorithmique et de programme
- Notion de variable informatique

Compétences associées

- Écrire, mettre au point (tester, corriger) et exécuter un programme en réponse à un problème donné

ALGORITHMIQUE ET PROGRAMMATION

Écrire, mettre au point, exécuter un programme

Les repères qui suivent indiquent une progressivité dans le **niveau de complexité des activités** relevant de ce thème. Certains élèves sont capables de réaliser des activités de troisième niveau dès le début du cycle.

1 ^{er} niveau	2 ^e niveau	3 ^e niveau
<p>À un premier niveau, les élèves mettent en ordre et/ou complètent des blocs Scratch fournis par le professeur pour construire un programme simple. L'utilisation progressive des instructions conditionnelles et/ou de la boucle « répéter ... fois ») permet d'écrire des scripts de déplacement, de construction géométrique ou de programme de calcul.</p>	<p>À un deuxième niveau, les connaissances et les compétences en algorithmique et en programmation s'élargissent par :</p> <ul style="list-style-type: none"> - l'écriture d'une séquence d'instructions (boucle « si ... alors » et boucle « répéter ... fois ») ; - l'écriture de programmes déclenchés par des événements extérieurs ; - l'intégration d'une variable dans un programme de déplacement, de construction géométrique, de calcul ou de simulation d'une expérience aléatoire. 	<p>À un troisième niveau, l'utilisation simultanée de boucles « répéter ... fois », et « répéter jusqu'à ... » et d'instructions conditionnelles permet de réaliser des figures, des calculs et des déplacements plus complexes. L'écriture de plusieurs scripts fonctionnant en parallèle permet de gérer les interactions et de créer des jeux.</p> <p>La décomposition d'un problème en sous-problèmes et la traduction d'un sous-problème par la création d'un bloc-utilisateur contribuent au développement des compétences visées.</p>

PROGRAMMES DU LYCÉE

Aménagement du programme de mathématiques

4. Algorithmique et programmation

La démarche algorithmique est, depuis les origines, une composante essentielle de l'activité mathématique. Au cycle 4, en mathématiques et en technologie, les élèves ont appris à écrire, mettre au point et exécuter un programme simple. Ce qui est proposé dans ce programme est une consolidation des acquis du cycle 4 autour de deux idées essentielles :

- la notion de fonction d'une part, et
- la programmation comme production d'un texte dans un langage informatique d'autre part.

Dans le cadre de cette activité, les élèves sont entraînés :

- à décrire des algorithmes en langage naturel ou dans un langage de programmation ;
- à en réaliser quelques-uns à l'aide d'un programme simple écrit dans un langage de programmation textuel;
- à interpréter des algorithmes plus complexes.

Un langage de programmation simple d'usage est nécessaire pour l'écriture des programmes. Le choix du langage se fera parmi les langages interprétés, concis, largement répandus, et pouvant fonctionner dans une diversité d'environnements.

L'algorithmique a une place naturelle dans tous les champs des mathématiques et les problèmes ainsi traités doivent être en relation avec les autres parties du programme (fonctions, géométrie, statistiques et probabilité, logique) mais aussi avec les autres disciplines ou la vie courante.

À l'occasion de l'écriture d'algorithmes et de petits programmes, il convient de donner aux élèves de bonnes habitudes de rigueur et de les entraîner aux pratiques systématiques de vérification et de contrôle. En programmant, les élèves revisitent les notions de variables et de fonctions sous une forme différente. Il convient d'y être attentif.

Définitions d'algorithme

Algorithme :

Le mot « algorithme » vient de Al Khwarizmi, surnom du mathématicien arabe Muhammad Ibn Musa (9^{ème} siècle), né à Khwarizem, en Ouzbékistan.

Son sens a changé plusieurs fois avant d'en arriver à ses significations actuelles.

Un algorithme est une suite finie de règles à appliquer dans un ordre déterminé à un nombre fini de données pour arriver avec certitude (c'est-à-dire sans indétermination ou ambiguïté), en un nombre fini d'étapes, à un certain résultat et cela indépendamment des données.

Un algorithme ne résout donc pas seulement un problème unique mais toute une classe de problèmes ne différant que par les données mais gouvernés par les mêmes prescriptions.

(...)

Un programme informatique est un algorithme rédigé dans un langage compris par l'ordinateur.

Dictionnaire des mathématiques 1996

Alain Bouvier – Michel George – François Le Lionnais

Un **algorithme**, très simplement, c'est une méthode. Une façon systématique de procéder pour faire quelque chose : trier des objets, situer des villes sur une carte, multiplier deux nombres, extraire une racine carrée, chercher un mot dans le dictionnaire... Il se trouve que certaines actions mécaniques - peut-être toutes ! - se prêtent bien à la décortication. On peut les décrire de manière générale, identifier des procédures, des suites d'actions ou de manipulations précises à accomplir séquentiellement. C'est cela, un **algorithme**.

En tant que méthode, il répond donc à des questions du type : « comment faire ceci ? », « obtenir cela ? », « trouver telle information ? », « calculer tel nombre ? ».

C'est un concept pratique, qui traduit la notion intuitive de procédé systématique, applicable mécaniquement, sans réfléchir, en suivant simplement un mode d'emploi précis.

Qu'est-ce qu'un algorithme ? 2004

Philippe Flajolet et Étienne Parizot

[HTTPS://INTERSTICES.INFO/JCMS/C_5776/QU-EST-CE-QU-UN-ALGORITHME?MEDIEGO_RUUID=0C3B84E3-4A5D-11E6-940D-8DE9DD0092F3](https://interstices.info/jcms/C_5776/qu-est-ce-qu-un-algorithme?mediego_ruuid=0c3b84e3-4a5d-11e6-940d-8de9dd0092f3)

"Un **algorithme** est une procédure de résolution de problème, s'appliquant à une famille **d'instances** du problème et produisant, en un nombre fini d'étapes constructives, effectives, non-ambigües et organisées, la réponse au problème pour toute **instance** de cette famille."

Simon Modeste

<https://tel.archives-ouvertes.fr/tel-00783294/file/Modeste-these-TEL.pdf>

Notre but

Intégrer une part de l'algorithmique et de la programmation dans notre cours de mathématiques et ne pas le faire vivre uniquement à côté.

Donc nous avons cherché des situations présentes dans notre enseignement qui nous permettent d'utiliser l'algorithmique et la programmation.

Nous nous sommes également confrontés aux outils déjà existants pour voir la pertinence.

Projet d'ajustement et de clarification des programmes de mathématiques
des cycles 2, 3 et 4 – 31 mai 2018

. L'élève consolide sa compréhension de notions mathématiques au programme comme les ordres de grandeur, la proportionnalité, le calcul littéral, les systèmes de coordonnées, le repérage ou les statistiques en les mobilisant dans des situations issues de la physique, la chimie, les sciences de la vie et de la Terre, la technologie, ou la géographie.

Les différentes tâches que nous avons dégagés sur l'algorithmique et la programmation :

- Lecture et analyse (Que fait ce programme ?)
- Test (vérifier qu'un programme fasse bien la tâche attendue)
- Correction (Corriger ce programme pour qu'il fasse telle chose)
- Simplification ou modification (Transformer ce programme pour qu'il fasse telle chose ou pour qu'il utilise moins de variables ou pour économiser le nombre d'étapes)
- Implémentation (Transformer un algorithme en programme)
- Construction (Ecrire un algorithme et le programmer pour faire telle chose)

Les logiciels et leur langage

- GeoTortue : Logo
- Geogebra : Langage propre ou Javascript
- Excel : VisualBasic
- Scratch : SmallTalk (Smalltalk est un langage de programmation orienté objet, réflexif et dynamiquement typé. Il a été créé en 1972. Il est inspiré par les langages Lisp et Simula)
- Algobox : Javascript
- Langage Python

Exemples de situations

1. Pont de Barddhaman- Freyssinet
2. Grille de défense
3. Distance de recul
4. Date de Pâques
5. Chronomètre
6. Convertisseur Celsius – Fahrenheit
7. Méthode de Héron (Métriques)
8. Racine carrée et méthode de Héron
9. Racine carrée de 2 et méthode de Théon
10. Pavage Quadrilatère
11. Codage César
12. Polygone régulier
13. Polygone régulier étoilé
14. Carte bancaire
15. Code INSEE

Technique

1. Addition de fractions
2. Addition en HMS
3. Résolution d'équation de degré 1
4. Décomposition en facteurs premiers
5. Test nombre premier
6. Tri Eratosthène
7. Calcul de la médiane
8. Grapheur

Visualisation

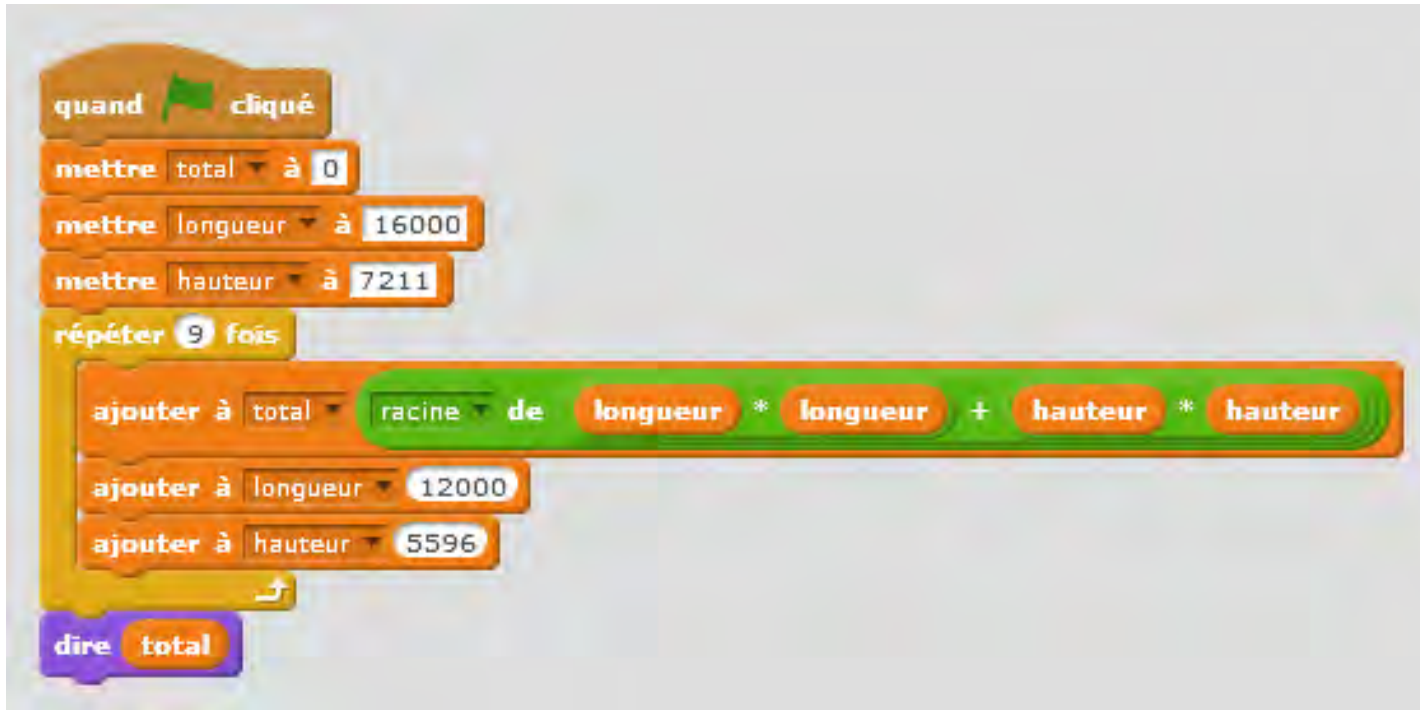
1. Simulation de lancer de dé (Avec liste)
2. Somme de deux dés
3. Simulation de lancer de pièces
4. Jeu lancer de pièces

Jeux

1. Labyrinthe
2. Morpion
3. Puissance 4
4. Pong
5. Cadavre exquis

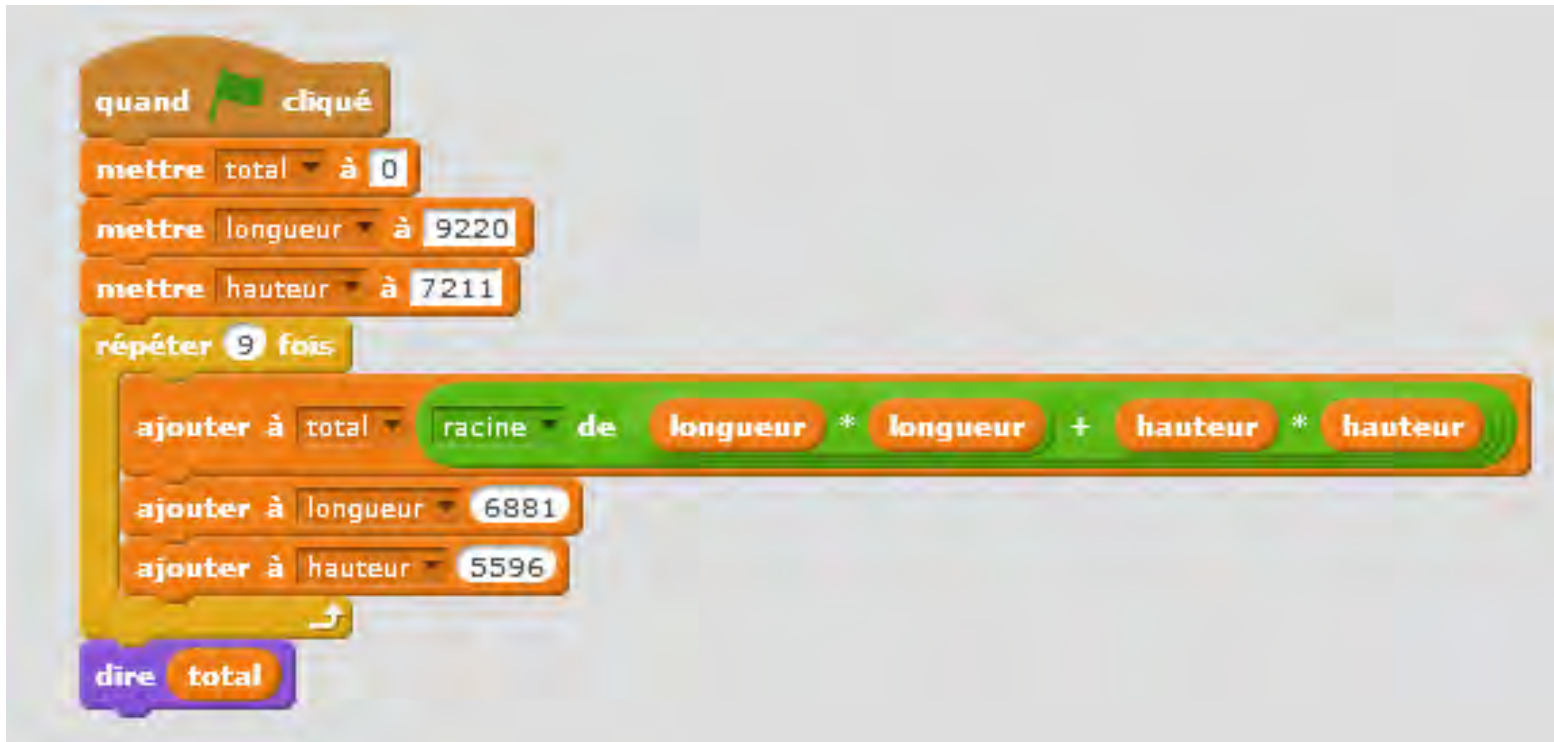
Exemples de situations

Pour la partie gauche



Que faut-il modifier pour la partie de droite ?

Pour la partie droite



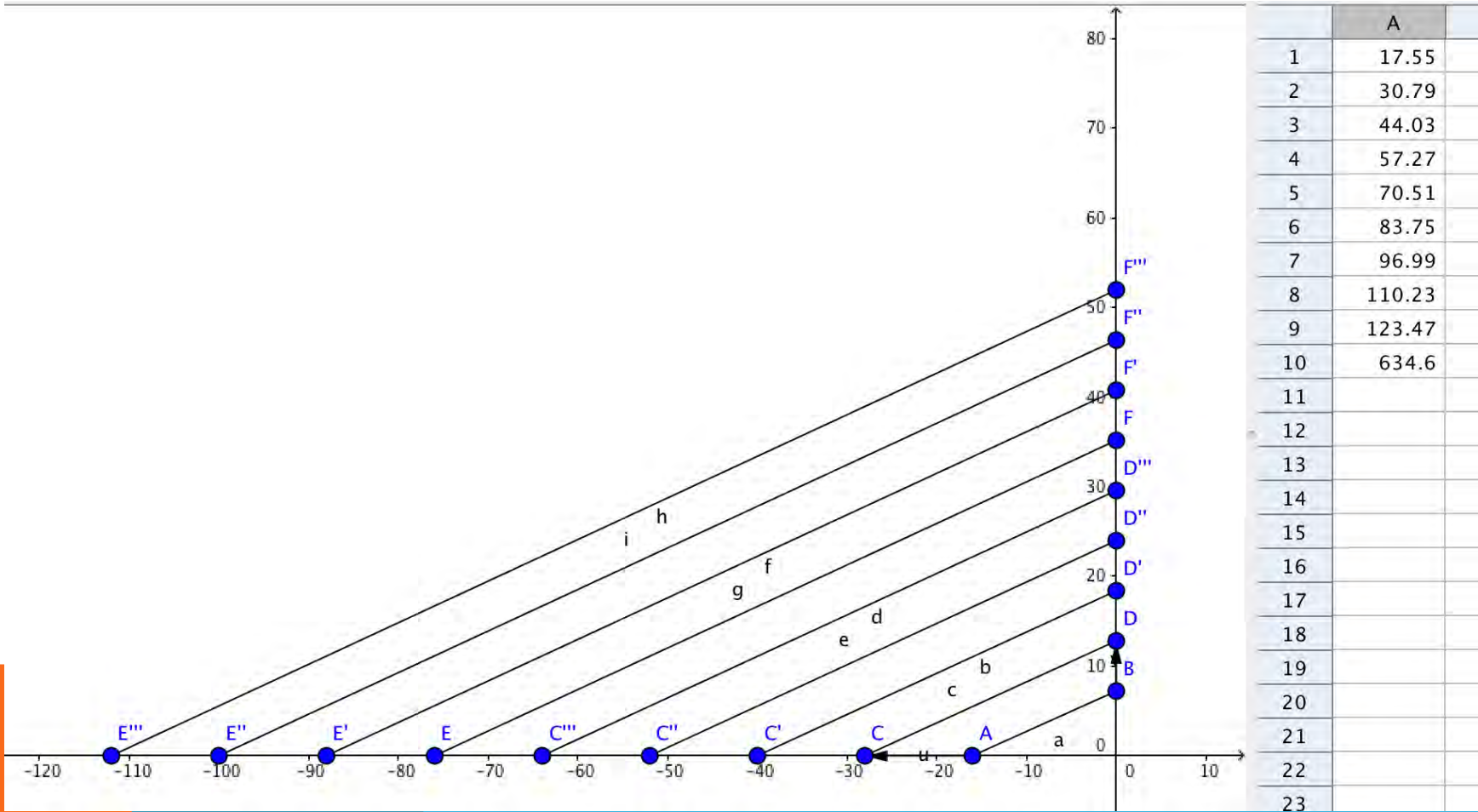
Variables
Boucles

	A	B	C	D	E
1					
2	Hauteur	5596		Largeur	12000
3					
4	16000	7211	17550		
5	28000	12807	30790		
6	40000	18403	44030		
7	52000	23999	57271		
8	64000	29595	70511		
9	76000	35191	83752		
10	88000	40787	96993		
11	100000	46383	110233		
12	112000	51979	123474		
13			634604		
14					

Avec le Tableur

	A	B	C	D	E
1					
2	Hauteur	5596		Largeur	6881
3					
4	9220	7211	11705		
5	16101	12807	20573		
6	22982	18403	29442		
7	29863	23999	38311		
8	36744	29595	47180		
9	43625	35191	56050		
10	50506	40787	64919		
11	57387	46383	73788		
12	64268	51979	82657		
13			424625		
14					

Avec Geogebra





Grille de défense



Calculer la longueur de ferrure en fonction de la longueur du côté du carré, de la taille du quart de cercle et du nombre de barreaux dans le but de pouvoir fabriquer cette grille à la taille désirée.

```

quand cliqué
supprimer l'élément tout de la liste donnée
demander côté du carré et attendre
mettre côté à réponse
mettre total à 4 * côté
demander Rayon du quart de cercle et attendre
mettre rayon à réponse
ajouter à total 3.14 * rayon / 2
demander nombre de barreaux et attendre
mettre nombre à réponse
mettre angle à 90 / nombre + 1
mettre intermédiaire à angle
répéter nombre fois
  si intermédiaire < 45 alors
    ajouter à total côté / cos de intermédiaire - rayon
    ajouter côté / cos de intermédiaire - rayon à donnée
    ajouter à intermédiaire angle
  sinon
    ajouter à total côté / cos de 90 - intermédiaire - rayon
    ajouter côté / cos de 90 - intermédiaire - rayon à donnée
    ajouter à intermédiaire angle

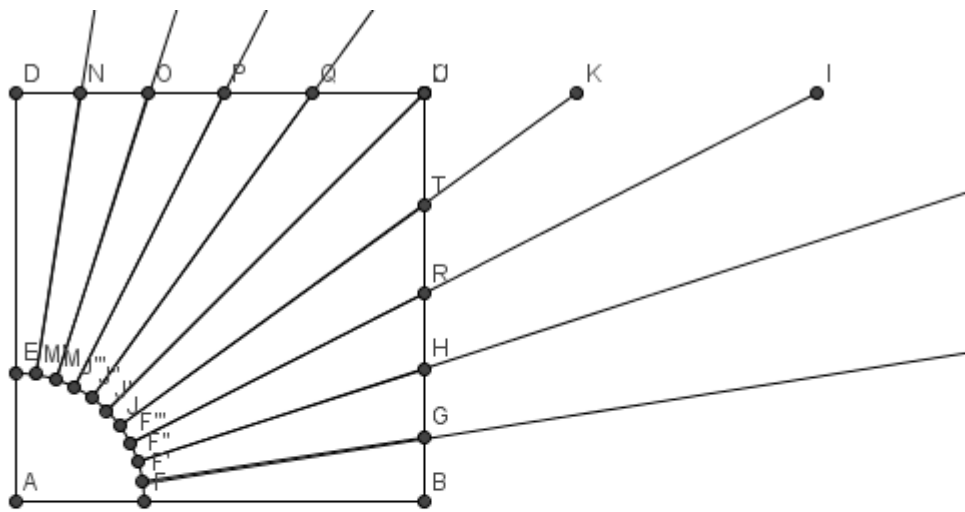
```

côté 6
 rayon 1.9
 nombre 9
 angle 9
 total 71.43614

donnée	
1	4.174791
2	4.408773
3	4.833957
4	5.516408
5	6.585281
6	5.516408
7	4.833957
8	4.408773
9	4.174791

+ longueur: 9

Variables
Boucles
Instructions conditionnelles



Nombre de barreaux

$n = 10$

Rayon du quart de cercle

$c = 1.9$



Hauteur

$b = 6$

Largeur

$a = 6$



Pourtour	Quart Ce...	Barreaux	
24	2.98	44.45	71.44
		4.41	
		4.83	
		0	
		5.52	
		6.59	
		5.52	
		4.83	
		4.41	
		4.17	
		0	
		0	
		0	
		0	
		0	
		0	
		0	
		0	
		0	
		4.17	

Distance de recul

« Calculer à quelle distance il faut regarder un écran correspond en fait à savoir à quelle distance notre œil ne fait plus la différence entre 2 pixels. En effet, lorsque nous sommes trop près d'un écran quel qu'il soit, il est possible de distinguer les pixels les uns des autres. En se reculant, cette distinction va être de plus en plus difficile jusqu'à ce que notre œil ne sépare plus les points et les confonde. Cette faculté de distinction de 2 points est le pouvoir séparateur de l'œil. Il correspond en théorie à un angle de $1/60^\circ$. »

En déduire la distance minimale pour regarder l'écran ayant les caractéristiques suivantes :

TV 4K : résolution : 3840×2160 pixels.

Diagonale : 152cm (60")

TV Full HD (HD TV 1080p) : résolution : 1920×1080 pixels.

Diagonale : 102cm (40")

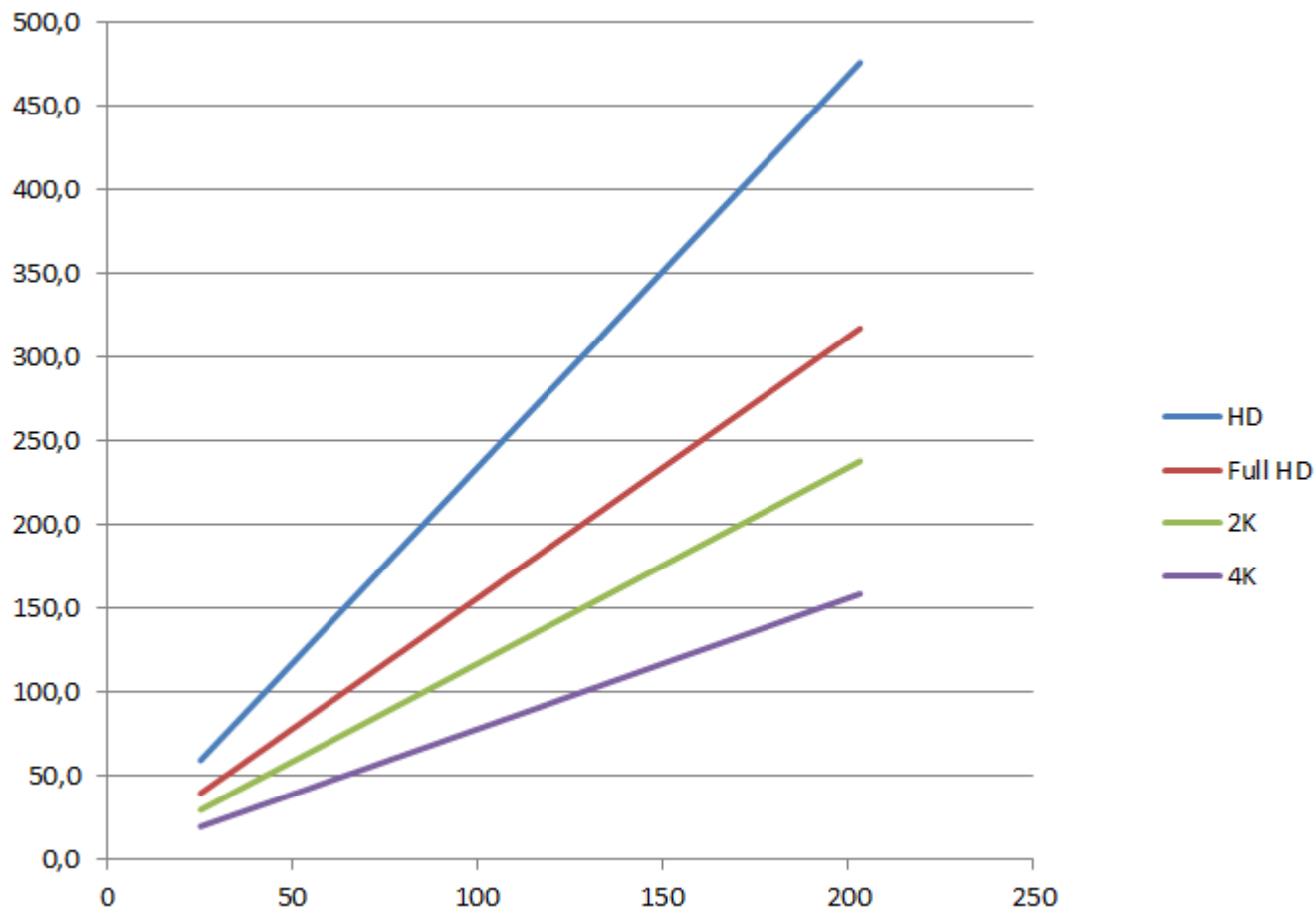
Puis construire un algorithme qui donne cette distance quand on lui a donné la résolution et la diagonale.

```
quand cliqué
demander Taille de la diagonale en cm : et attendre
mettre diagonale à réponse
demander format 4K (1) ou full HD (2) : et attendre
mettre format à réponse
si format = 1 alors
mettre largeur à 3840
sinon
mettre largeur à 1920
mettre pixel à 16 * diagonale / racine de 337 * largeur
mettre distance à pixel / tan de 1 / 60
dire distance
```

diagonale 102
format 2
distance 159.176095
largeur 1920
pixel 0.046302

Variables
Instructions conditionnelles

	Diagonale (pouces)	10	12	14	15	17	19	22	24	26	28	30	32	39	40	42
	Diagonale (cm)	25,4	30,48	35,56	38,1	43,18	48,26	55,88	60,96	66,04	71,12	76,2	81,28	99,06	101,6	106,68
HD	1280*720	59,5	71,3	83,2	89,2	101,1	113,0	130,8	142,7	154,6	166,5	178,4	190,3	231,9	237,8	249,7
Full HD	1920*1080	39,6	47,6	55,5	59,5	67,4	75,3	87,2	95,1	103,1	111,0	118,9	126,8	154,6	158,6	166,5
2K	2560*1440	29,7	35,7	41,6	44,6	50,5	56,5	65,4	71,3	77,3	83,2	89,2	95,1	115,9	118,9	124,9
4K	3840*2160	19,8	23,8	27,7	29,7	33,7	37,7	43,6	47,6	51,5	55,5	59,5	63,4	77,3	79,3	83,2



Date de Pâques

La définition précise du jour de Pâques fut établie en 325 par le concile de Nicée. Les Pères de l'Église réunis par l'empereur Constantin la fixèrent ainsi :

« Pâques est le dimanche qui suit le 14^e jour de la Lune qui atteint cet âge le 21 mars ou immédiatement après. »

Autrement dit cette fête doit être célébrée le premier dimanche après la première pleine lune suivant l'équinoxe de printemps.

Pour calculer cette date, on dispose de plusieurs algorithmes :

Algorithme de Gauss, algorithme de Meeus, algorithme de Conway, ...

Algorithme de Gauss pour les années > 1582:

Date de Pâques grégorienne (algorithme de Gauss)

Dividende	Diviseur	Quotient	Reste
Année	19		a
Année	4		b
Année	7		c
Année	100	k	
$13 + 8k$	25	p	
k	4	q	
$15 - p + k - q$	30		M
$4 + k - q$	7		N
$19a + M$	30		d
$2b + 4c + 6d + N$	7		e

Exemple pour 2016 :

$$2016 : 19 = 106$$

$$2016 : 4 = 504$$

$$2016 : 7 = 288$$

$$2016 : 100 = 20$$

$$13 + 8k = 13 + 160 = 173$$

$$20 : 4 = 5$$

$$15 - 6 + 20 - 5 = 24$$

$$M = 24$$

$$4 + 20 - 5 = 19$$

$$19 * 2 + 24 = 62$$

$$d = 2$$

$$2*0 + 4*0 + 6*2 + 5 = 17$$

$$H = (22 + 2 + 3) \text{ Mars} = 27 \text{ Mars}$$

$$Q = (27 - 31) = (2 + 3 - 9) \text{ Avril} = -4 \text{ Avril}$$

$$173 : 25 = 6$$

$$24 : 30 = 0$$

$$19 : 7 = 2$$

$$62 : 30 = 2$$

$$17 : 7 = 2$$

$$a = 2$$

$$b = 0$$

$$c = 0$$

$$k = 20$$

$$p = 6$$

$$q = 5$$

$$N = 5$$

$$e = 3$$

Les Pâques grégoriennes en calendrier grégorien sont le :

1. $H = (22 + d + e)$ mars
2. ou le $Q = (H - 31) = (d + e - 9)$ avril ;
3. Si $d = 29$ et $e = 6$, remplacer le 26 avril par le 19 avril ;
4. Si $d = 28$, $e = 6$ et $\text{RESTE}((11M + 11) / 30) < 19$, remplacer le 25 avril par le 18 avril.

réponse 2016

H 27 Mars

```
quand cliqué
  cacher
  mettre H à 0
  demander "Quelle année ?" et attendre
  mettre v à réponse
  mettre a à v modulo 19
  mettre b à v modulo 4
  mettre c à v modulo 7
  mettre k à arrondi de v / 100 - 0.5
  mettre p à arrondi de 13 + 8 * k / 25 - 0.5
  mettre q à arrondi de k / 4 - 0.5
  mettre M à 15 - p + k - q modulo 30
  mettre N à 4 + k - q modulo 7
  mettre d à 19 * a + M modulo 30
  mettre e à 2 * b + 4 * c + 6 * d + N modulo 7
  mettre H à 22 + d + e
  si d = 29 et e = 6 alors
    mettre H à 50
  si d = 28 et e = 6 et 11 * M + 11 modulo 30 < 19 alors
    mettre H à 49
  si H < 32 alors
    mettre H à regroupe H Mars
  sinon
    mettre H à regroupe H - 31 Avril
```

Variables
Instructions conditionnelles

Algorithme de Meeus pour les années > 1582:

L'algorithme de Meeus pour le calcul de la date de Pâques est la réunion de deux méthodes : l'algorithme de Delambre pour le calendrier julien, publié en 1814 et l'algorithme de Butcher, pour le calendrier grégorien, publié en 1877.

Si $m = 3$, le dimanche de Pâques est le $(j + 1)$ mars

Si $m = 4$, le dimanche de Pâques est le $(j + 1)$ avril

Exemple pour 2016 :

$$2016 : 19 = 106 \text{ reste } 2$$

$$n = 2$$

$$2016 : 100 = 20 \text{ reste } 16$$

$$c = 20 \text{ et } u = 16$$

$$20 : 4 = 5 \text{ reste } 0$$

$$s = 5 \text{ et } t = 0$$

$$28 : 25 = 1 \text{ reste } 3$$

$$p = 1$$

$$20 - 1 + 1 = 20 \quad 20 : 3 = 6 \text{ reste } 2$$

$$q = 6$$

$$19 * 2 + 20 - 5 - 6 + 15 = 62$$

$$62 : 30 = 2 \text{ reste } 2$$

$$e = 2$$

$$16 : 4 = 4 \text{ reste } 0$$

$$b = 4 \text{ et } d = 0$$

$$32 + 2*0 + 2*4 - 2 - 0 = 38$$

$$38 : 7 = 5 \text{ reste } 3$$

$$L = 3$$

$$2 + 11*2 + 22*3 = 90$$

$$90 : 451 = 0 \text{ reste } 90$$

$$h = 0$$

$$2 + 3 - 7*0 + 114 = 119$$

$$119 : 31 = 3 \text{ reste } 26 \text{ donc le } 26 / 03, \text{ le } 26 \text{ Mars}$$

Comme $m = 3$ alors c'est le $(j+1)$ Mars donc le 27 Mars

Date de Pâques grégorienne (algorithme de Butcher)

Dividende	Diviseur	Quotient	Reste	Explication
Année	19		n	cycle de Méton
Année	100	c	u	centaine et rang de l'année
c	4	s	t	siècle bissextile
$c + 8$	25	p		proemptose
$c - p + 1$	3	q		métemptose
$19n + c - s - q + 15$	30		e	épacte
u	4	b	d	année bissextile
$32 + 2t + 2b - e - d$	7		L	lettre dominicale
$n + 11e + 22L$	451	h		correction
$e + L - 7h + 114$	31	m	j	mois et quantième du Samedi saint

Algorithme de Conway pour les années > 1582:

Date de Pâques grégorienne (Algorithme de Conway)

Dividende	Diviseur	Quotient	Reste	Expression	Explication
Année	100	s	t		s : année séculaire, t : millésime
t	4	a			Terme bissextile
s	4		p		
9 - 2*p	7		jps		jour-pivot séculaire ^{Note 4}
jps + t + a	7		jp		jour-pivot de l'année courante
Année	19		g		
				G = g + 1	Cycle de Méton
s	4	b			Métempose
8 (s + 11)	25	r			Proemptose
				C = -s + b + r	Correction séculaire
11 G + C	30		d		
d + 30	30		d		Pleine Lune pascale ^{Note 5}
551 - 19 d + G	544	h			Correction des exceptions à l'épacte ^{Note 6}
50 - d - h	7		e		Écart de la Pleine Lune pascale au Jour-pivot
e + jp	7		f		Jour de la Pleine Lune pascale
				R = 57 - d - f - h	Dimanche de Pâques

R est la date de Pâques en jours de mars

Si $R \leq 31$ alors

Mois = mars

Quantième = R

sinon

Mois = avril

Quantième = R - 31

Exemple pour 2016 :

$2016 : 100 = 20$ reste 16 $s = 20$ et $t = 16$ $16 : 4 = 4$ $a = 4$
 $20 : 4 = 5$ reste 0 $p = 0$ $9 - 2 * 0 = 9$ $9 : 7 = 1$ reste 2 $jps = 2$
 $2 + 16 + 4 = 22$ $22 : 7 = 3$ reste 1 $jp = 1$
 $2016 : 19 = 106$ reste 2 $g = 2$ $G = 2 + 1 = 3$
 $20 : 4 = 5$ reste 0 $b = 5$ $8 * (20 + 11) = 248$ $248 : 25 = 9$ reste 23 $r = 9$
 $C = -20 + 5 + 9 = -4$ $11 * 3 + -4 = 29$ $29 : 30 = 0$ reste 29 $d = 29$
 $551 - 19 * 29 + 3 = 3$ $3 : 544 = 0$ reste 3 $h = 0$ $50 - 29 - 0 = 21$ $21 : 7 = 3$ reste 0
 $e = 0$
 $0 + 1 = 1$ $1 : 7 = 0$ reste 1 $f = 1$ $R = 57 - 29 - 1 - 0 = 27$
 Donc le 27 Mars

L'adoption, en 1582, de la réforme du calendrier julien promue par le Pape Grégoire XIII entraînait inévitablement des modifications du calcul de la date de Pâques. La suppression d'une année bissextile tous les quatre-cents produisait des décalages entre de cycle solaire et le cycle lunaire qui devaient être pris en compte. De plus, les astronomes Clavius et Lilius, attachés au Pape Grégoire XIII, profitèrent de la réforme pour introduire une correction du Cycle de Méton dont on savait depuis longtemps qu'il n'était qu'une approximation. La suppression de 3 jours bissextiles tous les quatre-cents ans se traduit dans le calcul de la date de Pâques par un décalage appelé métemptose. De plus, la correction du cycle de Méton conduit également à un décalage appelé proemptose.

Le calcul grégorien de la date de Pâques n'est pas une création *ex nihilo* mais une modification du calcul de la date de Pâques julienne. Le principe de ce calcul consiste à mettre en correspondance le cycle solaire (appelé aussi *équation solaire*), qui permet de calculer quels jours de l'année sont des dimanches, et le cycle lunaire (appelé aussi *équation lunaire*) permettant, lui, de déterminer la date des Nouvelles Lunes. Pour une année donnée, l'équation solaire est caractérisée par la Lettre dominicale qui spécifie le premier dimanche de janvier ; elle donne le moyen de calculer les jours de l'année qui sont un dimanche, en particulier en mars et avril. L'équation lunaire est caractérisée par l'épacte, c'est-à-dire l'âge de la Lune au 1^{er} janvier ; celle-ci permet de fixer les dates des Nouvelles Lunes de l'année, en particulier pour mars et avril. À l'aide de ces termes on détermine le dimanche qui tombe immédiatement après le quatorzième jour de la Lune qui tombe ou qui suit immédiatement le 21 mars. Les nouveautés apportées par le calendrier grégorien ont été reportée sur l'épacte. Des règles de *saut d'épacte* appliquent à l'épacte les décalages de la date de Pâques dus à la métemptose et à la proemptose.

Méthode de calcul de l'épacte grégorienne :

$$c = \text{QUOTIENT}(A / 100)$$

$$\acute{E}pacte = \text{RESTE}((11 \times \text{RESTE}(A / 19) + 8 - c + \text{QUOTIENT}(c / 4) + \text{QUOTIENT}((8 \times c + 13) / 25)) / 30)$$

Si $\acute{E}pacte = 25$ et que le nombre d'or $N = \text{RESTE}(\text{Année} / 19) > 10$, alors $\acute{E}pacte = 26$.

Exemple pour 2016 :

$$A = 2016$$

$$2016 : 100 = 20 \text{ reste } 16$$

$$2016 : 19 = 106 \text{ reste } 2$$

$$20 : 4 = 5$$

$$8 \times 20 + 13 = 173 \quad 173 : 25 = 6 \text{ reste } 23$$

$$11 \times 2 + 8 - 20 + 5 + 6 = 21$$

$$\text{Epacte} = 21$$

$$c = 20$$

$$\text{Reste}(A/19) = 2$$

$$\text{Quotient}(c/4) = 5$$

$$\text{Quotient}(8 \times c + 13) / 25 = 6$$

$$21 : 30 = 0 \text{ reste } 21$$

$$L = 7 - \text{RESTE}[(A + \text{QUOTIENT}[A / 4] + \text{QUOTIENT}[A / 400] - \text{QUOTIENT}[A / 100] + 6) / 7]$$

avec $L = 1 : A ; L = 2 : B$, etc.

Exemple pour 2016 :

$$A = 2016$$

$$2016 : 4 = 504 \text{ reste } 0$$

$$2016 : 400 = 5 \text{ reste } 16$$

$$2016 : 100 = 20 \text{ reste } 16$$

$$2016 + 504 + 5 - 20 + 6 = 2511$$

$$2511 : 7 = 358 \text{ reste } 5$$

$$L = 7 - 5 = 2 \text{ donc } B$$

$$\text{Quotient}(A/4) = 504$$

$$\text{Quotient}(A/400) = 5$$

$$\text{Quotient}(A/100) = 20$$

Date des Pâques grégoriennes

Épacte	Lettre dominicale						
	A	B	C	D	E	F	G
0	16A	17A	18A	19A	20A	14A	15A
1	16A	17A	18A	19A	13A	14A	15A
2	16A	17A	18A	12A	13A	14A	15A
3	16A	17A	11A	12A	13A	14A	15A
4	16A	10A	11A	12A	13A	14A	15A
5	9A	10A	11A	12A	13A	14A	15A
6	9A	10A	11A	12A	13A	14A	8A
7	9A	10A	11A	12A	13A	7A	8A
8	9A	10A	11A	12A	6A	7A	8A
9	9A	10A	11A	5A	6A	7A	8A
10	9A	10A	4A	5A	6A	7A	8A
11	9A	3A	4A	5A	6A	7A	8A
12	2A	3A	4A	5A	6A	7A	8A
13	2A	3A	4A	5A	6A	7A	1A
14	2A	3A	4A	5A	6A	31M	1A
15	2A	3A	4A	5A	30M	31M	1A

16	2A	3A	4A	29M	30M	31M	1A
17	2A	3A	28M	29M	30M	31M	1A
18	2A	27M	28M	29M	30M	31M	1A
19	26M	27M	28M	29M	30M	31M	1A
20	26M	27M	28M	29M	30M	31M	25M
21	26M	27M	28M	29M	30M	24M	25M
22	26M	27M	28M	29M	23M	24M	25M
23	26M	27M	28M	22M	23M	24M	25M
24	23A	24A	25A	19A	20A	21A	22A
25	23A	24A	25A	19A	20A	21A	22A
26	23A	24A	18A	19A	20A	21A	22A
27	23A	17A	18A	19A	20A	21A	22A
28	16A	17A	18A	19A	20A	21A	22A
29	16A	17A	18A	19A	20A	21A	15A

Donc pour 21 B on trouve le 27 Mars

La date de Pâques fait l'objet d'une fonction dans OpenOffice qui est :
DimanchedePaques(année)

Mais pas dans Excel qui propose comme formule :
=ARRONDI(DATE(Année;4;MOD(234-11*MOD(Année;19);30))/7;0)*7-6

La formule est donc :

$a = \text{année modulo } 19$

$b = (234 - 11 \times a) \text{ modulo } 30$

$c = \text{date} (\text{année}, \text{mois} = \text{Avril}, b)$

$d = \text{arrondi} (c/7 ; 0)$ donc arrondi à l'unité

Réponse = $d \times 7 - 6$ à écrire en format Date

Remarque les dates sont traduites en nombre dans Excel avec comme première date le 1/1/1900 qui correspond à 1 donc la réponse pour 2016 est obtenue en tapant 116 ou 2016.

Le calcul des années bissextiles est intégré dans le fait qu'il utilise l'année et donc l'ordinateur redonne le nombre de jours pour atteindre le jour choisi en tenant compte des règles des années bissextiles.

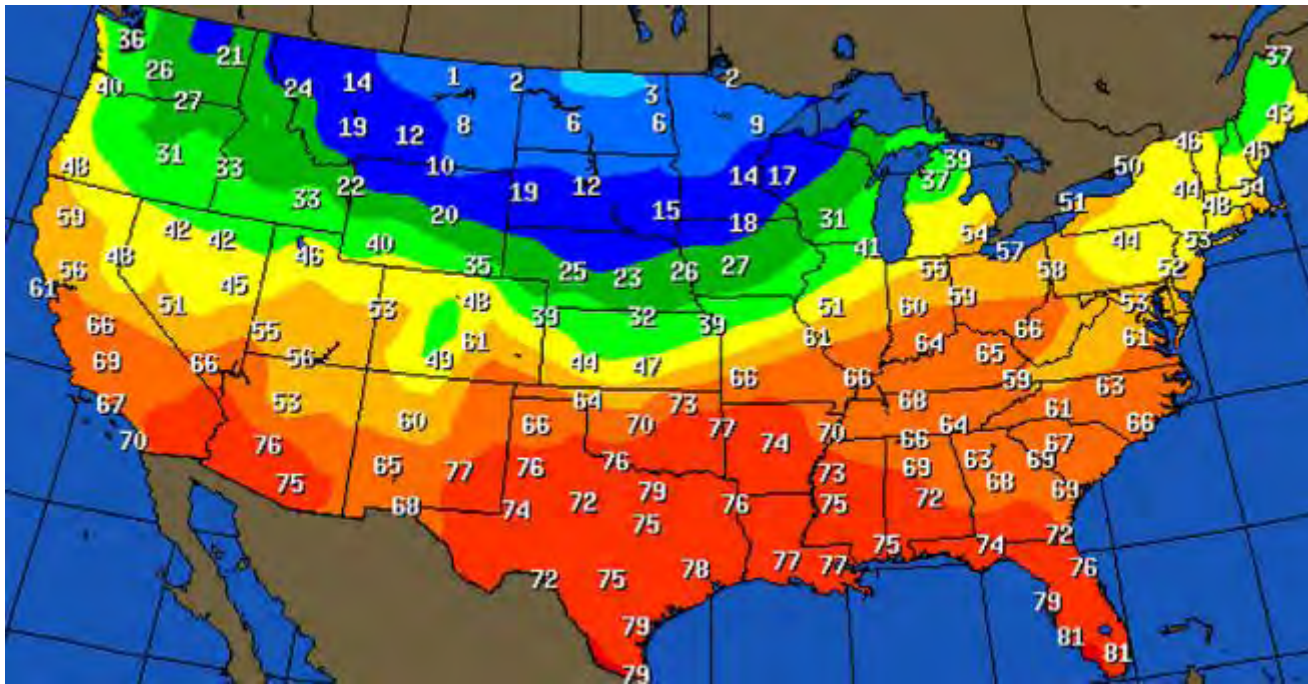
Donc cette méthode n'est pas transposable en un algorithme.

Chronomètre



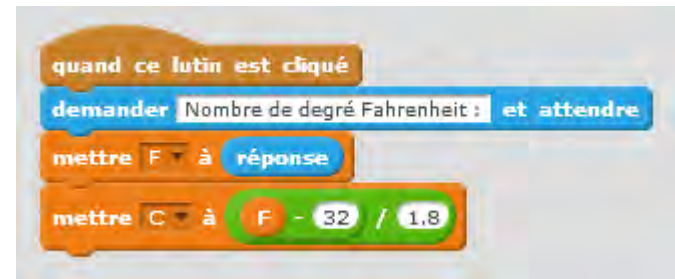
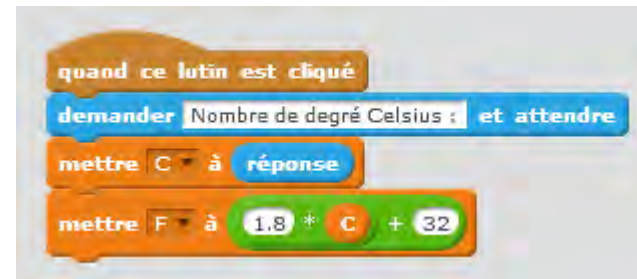
Variables
Boucles
Instructions conditionnelles
Scripts en parallèle

DUREES



Voici une carte météo des Etats-Unis mais les températures ne sont pas exprimées en degrés Celsius. Construire un convertisseur me permettant de la traduire.

Convertisseur Celsius - Fahrenheit



Variables
Scripts en parallèle

Celsius	Fahrenheit
10	50
10	50

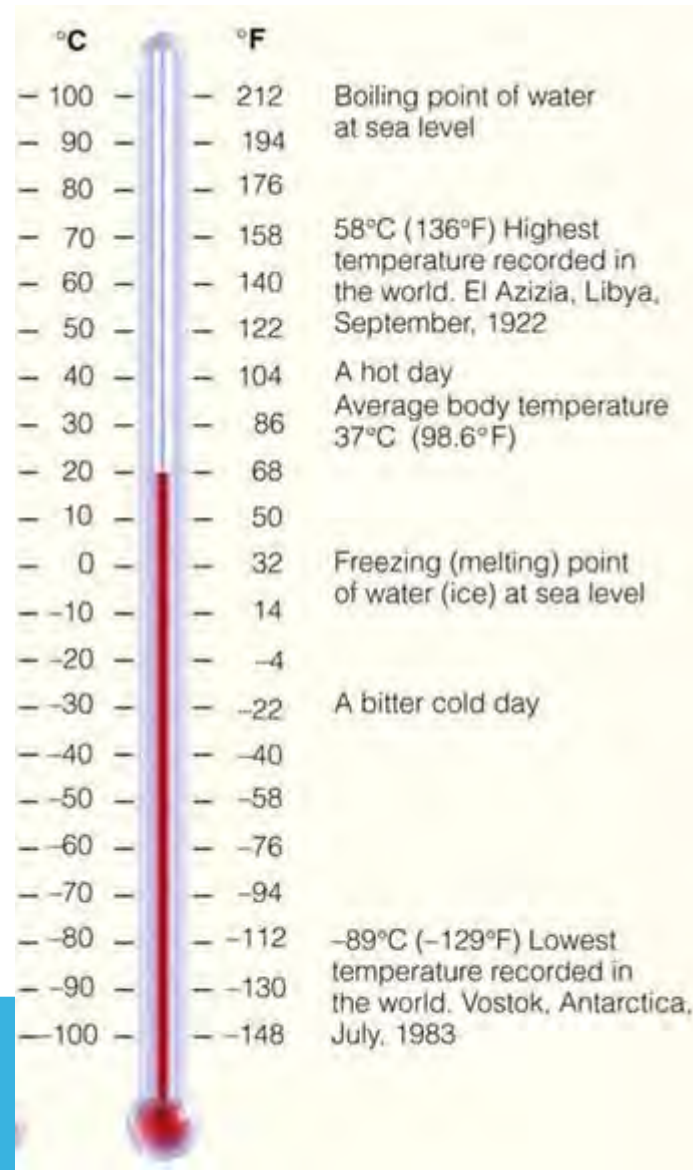
Avec le tableur

Celsius	Fahrenheit
10	=B4*1,8+32
=(D6-32)/1,8	50

OU

Fahrenheit	-4,0	-3,0	-2,0	-1,0	0,0	1,0	2,0	3,0	4,0	5,0	6,0	7,0	8,0	9,0	10,0	11,0	12,0	13,0	14,0
Celsius	-20,0	-19,4	-18,9	-18,3	-17,8	-17,2	-16,7	-16,1	-15,6	-15,0	-14,4	-13,9	-13,3	-12,8	-12,2	-11,7	-11,1	-10,6	-10,0
Fahrenheit	15,0	16,0	17,0	18,0	19,0	20,0	21,0	22,0	23,0	24,0	25,0	26,0	27,0	28,0	29,0	30,0	31,0	32,0	33,0
Celsius	-9,4	-8,9	-8,3	-7,8	-7,2	-6,7	-6,1	-5,6	-5,0	-4,4	-3,9	-3,3	-2,8	-2,2	-1,7	-1,1	-0,6	0,0	0,6
Fahrenheit	34,0	35,0	36,0	37,0	38,0	39,0	40,0	41,0	42,0	43,0	44,0	45,0	46,0	47,0	48,0	49,0	50,0	51,0	52,0
Celsius	1,1	1,7	2,2	2,8	3,3	3,9	4,4	5,0	5,6	6,1	6,7	7,2	7,8	8,3	8,9	9,4	10,0	10,6	11,1
Fahrenheit	53,0	54,0	55,0	56,0	57,0	58,0	59,0	60,0	61,0	62,0	63,0	64,0	65,0	66,0	67,0	68,0	69,0	70,0	71,0
Celsius	11,7	12,2	12,8	13,3	13,9	14,4	15,0	15,6	16,1	16,7	17,2	17,8	18,3	18,9	19,4	20,0	20,6	21,1	21,7
Fahrenheit	72,0	73,0	74,0	75,0	76,0	77,0	78,0	79,0	80,0	81,0	82,0	83,0	84,0	85,0	86,0	87,0	88,0	89,0	90,0
Celsius	22,2	22,8	23,3	23,9	24,4	25,0	25,6	26,1	26,7	27,2	27,8	28,3	28,9	29,4	30,0	30,6	31,1	31,7	32,2

Mais aussi



Méthode de Héron (Métriques)

Algorithme pour le calcul de la racine carrée

Alors, puisque les 720 n'ont pas le côté exprimable, nous prendrons le côté avec une très petite différence ainsi.

Puisque le carré le plus proche de 720 est 729 et qu'il a 27 comme côté, divise les 720 par le 27 : il en résulte 26 et deux tiers ;

ajoute les 27 : il en résulte 53 et deux tiers ;

de ceux-ci, la moitié : il en résulte $26 \frac{1}{2}$;

le côté de 720 sera donc $26 \frac{1}{2}$ à très peu près.

En effet, $26 \frac{1}{2}$ par eux-mêmes : il en résulte $720 \frac{1}{6}$;

de sorte que la différence est une 36^e part d'unité.

Et si nous voulons que la différence se produise par une part plus petite que $\frac{1}{6}$, au lieu de 729, nous placerons les 720 et $\frac{1}{6}$ maintenant trouvés, et, en faisant les mêmes choses, nous trouverons qu'il en résulte une différence inférieure, de beaucoup, à $\frac{1}{6}$.

Racine carrée et méthode de Héron

La méthode de Héron est l'algorithme suivant :

Partir d'une approximation x de \sqrt{a} .

Remplacer x par la moyenne de x et de $\frac{a}{x}$

Recommencer tant qu'on n'a pas atteint la précision voulue.

Exemple : on va prendre $a = 3$.

L'entier 1 est une première approximation de $\sqrt{3}$ car $1 \times 1 = 1$.

Appliquons la méthode de Héron :

on la remplace par la moyenne de 1 et de $\frac{3}{1}$; c'est 2. On

recommence : la moyenne de 2 et de $\frac{3}{2}$ est $\frac{7}{4} = 1,75$. puis, à

nouveau, la moyenne de $\frac{7}{4}$ et de $\frac{12}{7}$ est $\frac{97}{56}$, ce qui fait environ

1,73214... etc



Après avoir démontré ces valeurs théoriquement, on cherchera comment, étant donnée, l'aire d'un carré dont le côté n'est pas d'une longueur déterminée, nous calculerons ce côté approximativement.

En général, si nous cherchons le côté du carré (la racine carrée) de quelque nombre, nous prenons d'abord le côté (racine) du nombre carré le plus approchant. Ensuite le doublant, et divisant par ce double, le reste du nombre donné réduit en primes, nous ôtons du quotient, le carré; puis réduisant encore le reste en secondes, et le divisant par le double des unités et des primes, nous aurons à peu près le nombre cherché, qui exprime le côté de l'espace carré.

Mais, pour rendre cette démonstration de la distinction des parties d'un tout, plus sensible, par son application à quelque nombre contenu dans la composition (de Ptolémée), nous l'avons appliquée au nombre 4500 dont il a fait le côté de $67^{\circ} 4' 55''$. Supposons $ABGD$ l'aire du carré, seule rationnelle, et qu'il s'agisse d'en calculer le côté approximativement. Puisque le carré le plus proche 4500, ayant un côté rationnel, est 4489, carré du côté 67, ôtons le carré AZ , ou 4489, dont le côté est 67, du carré $ABGD$, le gnomon BZD vaudra donc les 11 parties restantes. Réduisons-les en primes, nous en aurons 660. Ensuite doublons $EZ=67$, parce que EZ est multiplié deux fois prenons ZH comme prolongeant EZ , et divisons les 660 primes par 134.

Cette division donne 4 primes, et nous avons ainsi chacune des droites ET , HK . En complétant les parallélogrammes TZ , ZK , nous aurons 536 pour les deux, ou 268 pour chacun. Puis réduisant les 124 primes restantes en 7440 secondes, nous en retranchons le parallélogramme de complément $ZL=16$ secondes, de manière qu'en ajoutant le gnomon au premier carré AZ , nous ayons le carré AL fait sur le côté $67^{\circ} 4'$, de $4497^{\circ} 56' 16''$. Enfin le gnomon restant, BL LD est de $2^{\circ} 3' 44''$, c'est-à-dire de 7424 secondes. Doublant donc encore TL , comme étant prolongée par LK , et divisant ces 7424 secondes par $134^{\circ} 8'$, il viendra 55 secondes à peu près au quotient, pour la valeur de chacune des droites TB et KD . Complétant les parallélogrammes BL , LD , nous les aurons de 7377 secondes et 20 tierces, et chacun de $3688^{\circ} 40'''$; mais il reste encore $46'' 40'''$ qui font à peu près le carré LG , dont le côté est $55''$. Ainsi, le côté du carré $ABGD$, 4500, se trouve être à peu près de $67^{\circ} 4' 55''$.

Théon d'Alexandrie
algorithme de calcul de la racine carrée
(commentaire de l'Almageste)

Racine carrée et méthode de Théon

THÉON D'ALEXANDRIE

ALGORITHME DE CALCUL DE LA RACINE CARRÉE
(COMMENTAIRE DE L'ALMAGESTE)

Nombre choisi

135

Nombre dont le carré
est le plus grand
mais inférieur

11

11,618889



```
quand je reçois message2
mettre reste à nombre - racine * racine
mettre reste à 60 * reste
mettre inter à plancher de reste / 2 * racine
mettre reste2 à reste - inter * 2 * racine
mettre reste2 à reste2 * 60
mettre reste2 à reste2 - inter * inter
mettre inter2 à plancher de reste2 / 2 * racine + inter / 60
mettre resultat à racine + inter / 60 + inter2 / 3600
```



```
quand cliqué
cacher la variable racine
demander Donner un nombre : et attendre
mettre nombre à réponse
envoyer à tous message1

quand je reçois message1
mettre racine à 1
répéter jusqu'à racine * racine > nombre
ajouter à racine 1
mettre racine à racine - 1
montrer la variable racine
envoyer à tous message2
```

Variables

Boucles

Instructions conditionnelles

Scripts en parallèle

Racine carrée de 2 et méthode de Théon

Algorithme :

$$X_0 = 1, Y_0 = 1$$

$$X_{n+1} = X_n + Y_n$$

$$Y_{n+1} = 2X_n + Y_n$$

$$Y_n/X_n \text{ tend vers } \sqrt{2}$$

Variables

Boucles

Instructions conditionnelles



r	
1	1.5
2	1.4
3	1.416667
4	1.413793
5	1.414286
6	1.414201
7	1.414216

+ longueur: 7

Pavage Quadrilatère

The image shows a Scratch script for drawing a grid of squares. The script starts with a 'when clicked' event, followed by 'hide', 'clear all', and 'choose color for the pen'. It then sets 'posx' to -200 and 'posy' to 160. A 'repeat 10 times' loop contains a 'repeat 7 times' loop. Inside the inner loop, it goes to the coordinates 'posx' and 'posy', sets the pen to 'Carré', lifts the pen, adds 10 to the pen color, and moves 'posy' down by 40. After the inner loop, it adds 40 to 'posx' and sets 'posy' back to 160.

```
quand cliqué
cacher
effacer tout
choisir la couleur pour le stylo
mettre posx à -200
mettre posy à 160
répéter 10 fois
  répéter 7 fois
    aller à x: posx y: posy
    stylo en position d'écriture
    Carré
    relever le stylo
    ajouter 10 à couleur du stylo
    ajouter à posy -40
  ajouter à posx 40
  mettre posy à 160
```

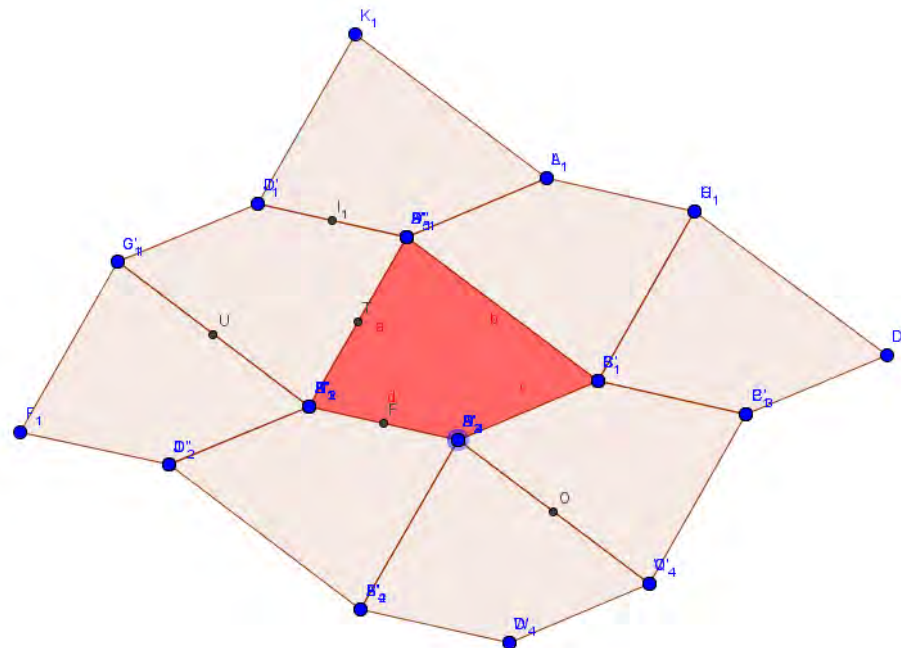
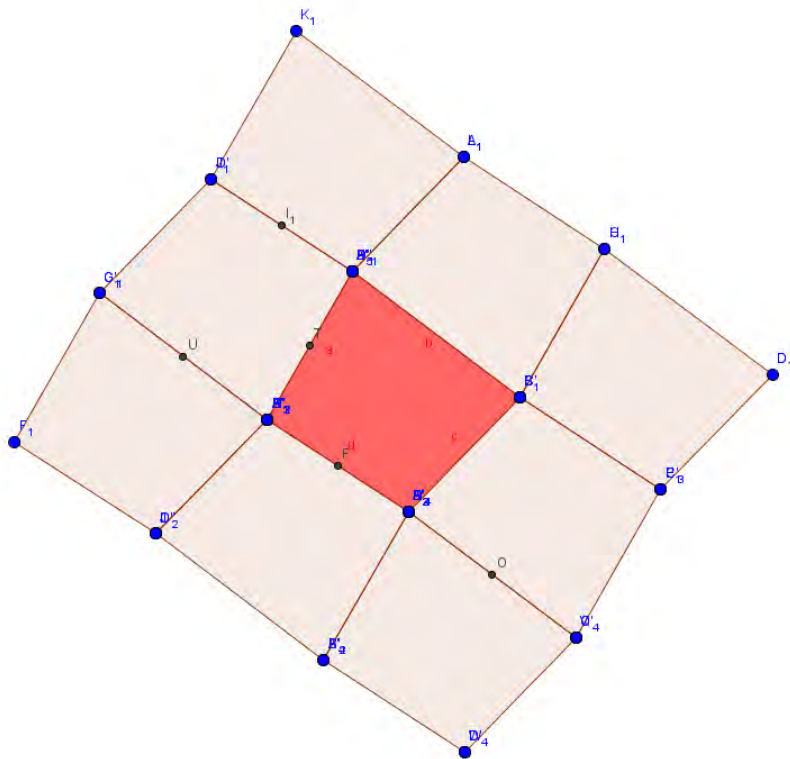
The image shows a Scratch stage titled 'PavageCarré' with a grid of 10x7 squares. The squares are drawn with a pen color of 10 and a pen style of 'Carré'. The grid is composed of 10 columns and 7 rows of squares, each with a side length of 40 pixels. The squares are drawn in a sequence that starts at (-200, 160) and moves right and down.

PavageCarré
v448

Variables
Boucles

Avec Geogebra

En se servant des symétries et de la translation.



C'est le même fichier mais le quadrilatère de base a été déformé.

Codage César

Texte	Nouveau Texte
<p>je suis chez mes cousins fred et domi.j'ai quitté lyon sous la pluie, ici le temps est magnifique. vendredi nous monterons des fagots au sommet du canigou pour alimenter le bûcher de la saint jean. fred est parti se reposer chez sa grand-mère à marseille. je ferai donc l'ascension du pic avec domi. viendras-tu avec nous ? je te sais courageuse, je pense qu'un dénivelé de 2 000 m ne te fait pas peur.</p> <p>réponds-moi vite. je t'embrasse.</p>	<p>to cesc mroj woc myecsx c pbon od nyws.t'ks aesddé viyx cyec vk zveso, sms vo dowzc ocd wkqxspaeo. foxnbons xyec wyxdobyxc noc pkqydc ke cywwod ne mkxsqye zyeb kvswoxdob vo lûmrob no vk cksxd tokx.</p> <p>pbon ocd zkbds co bozycob mroj ck qbkxn-wèbo à wkbcosvvo. to pobks nyxm v'kcmoxcsyx ne zsm kfom nyws. fsoxnbkc-de kfom xyec ? to do cksc myebkqoeco, to zoxco ae'ex néxsfové no 2 000 w xo do pksc zkc zoeb.</p> <p>bézyxnc-wys fsdo. to d'owlbkcco.</p>

Le programme :

pas	10	Lettre	Nouvelle Lettre
97	107	a	k
98	108	b	l
99	109	c	m
100	110	d	n
101	111	e	o
102	112	f	p
103	113	g	q
104	114	h	r
105	115	i	s
106	116	j	t
107	117	k	u
108	118	l	v
109	119	m	w
110	120	n	x
111	121	o	y
112	122	p	z
113	97	q	a
114	98	r	b
115	99	s	c
116	100	t	d
117	101	u	e
118	102	v	f
119	103	w	g
120	104	x	h
121	105	y	i
122	106	z	j

Sub codage()

Dim compte As Integer, nombre As Integer, pas As Integer, caract As String

Cells(2, 2) = LCase(Cells(2, 1))

Cells(2, 3) = ""

pas = Val(Cells(1, 5))

nombre = Len(Cells(2, 2))

For i = 1 To nombre

caract = Mid(Cells(2, 2), i, 1)

compte = Asc(caract)

If compte < 97 Or compte > 122 Then

caract = caract

Else

compte = compte + pas

If compte < 123 Then

caract = Chr(compte)

Else

caract = Chr(compte - 26)

End If

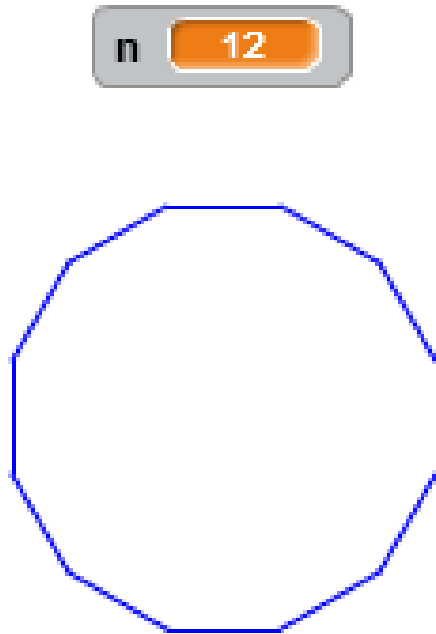
End If

Cells(2, 3) = Cells(2, 3) & caract

Next

End Sub

Polygone régulier



```
quand cliqué  
effacer tout  
demander nombre de côtés et attendre  
mettre n à réponse  
stylo en position d'écriture  
répéter n fois  
  avancer de 30  
  attendre 1 secondes  
  tourner de 360 / n degrés  
cacher
```

ANGLES

Variables
Boucles

Avec Geogebra

GeoGebra

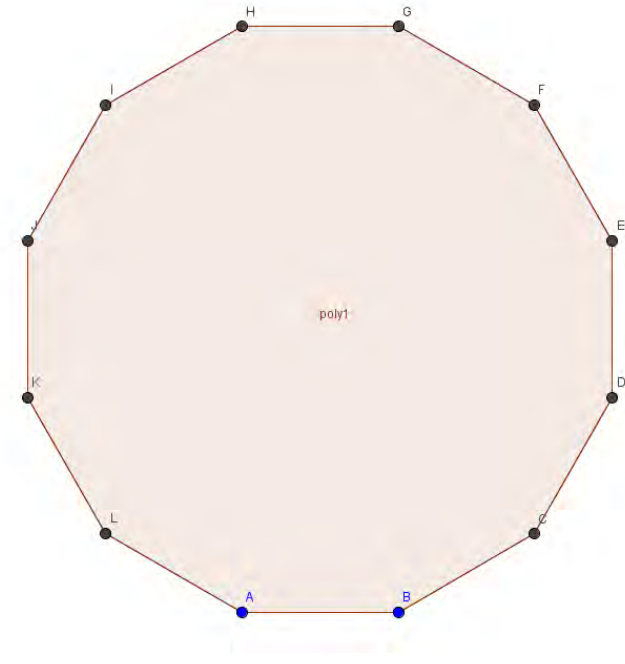
Fichier Éditer Affichage Options Outils Fenêtre Aide



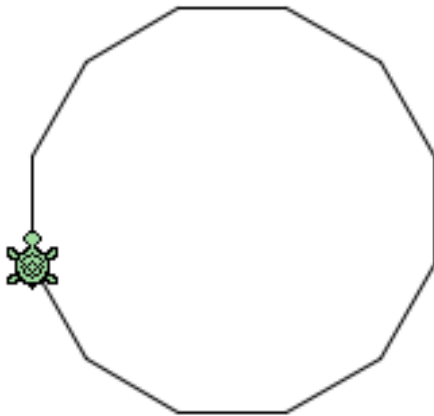
Algèbre

- Point
 - A = (11.32, -4.9)
 - B = (14.3, -4.9)
- Polygone
 - poly1 = 99.43
- Segment
 - a = 2.98

- Polygone
- Polygone régulier
- Polygone indéformable
- Polygone semi-déformable

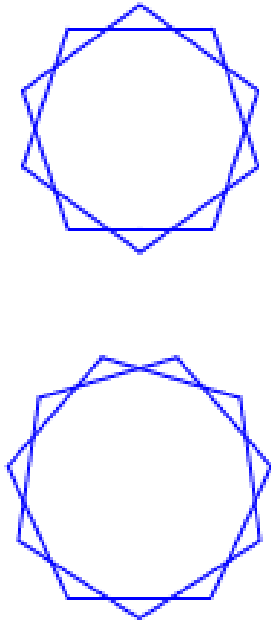


Avec Geotortue



```
> rep 12 [av 40; td 30]
```

Polygone régulier étoilé

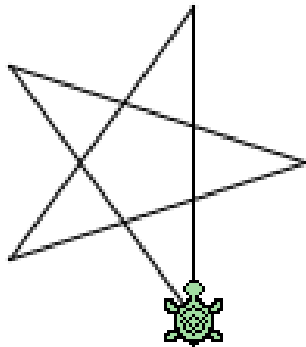


```
quand cliqué
effacer tout
demander nombre de côtés et attendre
mettre n à réponse
demander tour à droite et attendre
mettre t à réponse
aller à x: -25 y: -25
s'orienter à 90
style en position d'écriture
si t = 1 alors
  répéter n fois
    avancer de 50
    attendre 1 secondes
    tourner de 360 / n degrés
sinon
  si n modulo 2 = 0 alors
    répéter n / 2 fois
      avancer de 50
      attendre 1 secondes
      tourner de 720 / n degrés
    relever le stylo
    mettre t à -25 + 25 / tan de 360 / n - 25 / sin de 360 / n
    aller à x: 0 y: t
    s'orienter à 90 - 360 / n
    style en position d'écriture
    répéter (n / 2) fois
      avancer de 50
      attendre 1 secondes
      tourner de 720 / n degrés
    relever le stylo
    aller à x: 0 y: -25
    s'orienter à 90
    style en position d'écriture
  sinon
    répéter n fois
      avancer de 50
      attendre 1 secondes
      tourner de 720 / n degrés
relever le stylo
cacher
```

Variables
Boucles
Instructions conditionnelles

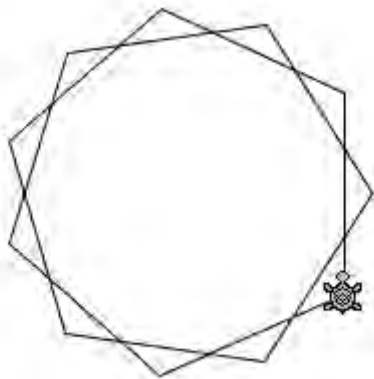
ANGLES

Avec Geotortue



Fenêtre de commande _____

```
> rep 5 [av 100; tg 144]
```



Fenêtre de commande _____

```
> rep 11 [av 100; tg 65.4]
```

Carte bancaire

Le numéro de carte bancaire n'est pas construit au hasard. Il comporte en général 16 chiffres, qui ont le sens suivant :

- le premier chiffre identifie le type de carte : 3 pour American Express, 4 pour Visa, 5 pour Mastercard...
- les trois chiffres suivants identifient la banque : un numéro de carte bancaire commençant par 4970 désignera toujours une carte Visa émise par la Banque Postale, un numéro commençant par 5612 une carte Mastercard émise par le Crédit Agricole.
- les sept numéros suivants identifient le numéro de compte dans la banque, chaque banque ayant son propre système d'identification.
- le dernier chiffre, enfin, est la **clé de Luhn**. Il joue le même rôle que la clé du RIB ou du numéro INSEE. Il est calculé à partir des autres chiffres en respectant l'algorithme suivant :

Numéro de carte à compléter : 4970 1012 3456 789

On multiplie par deux le 1er chiffre, le 3ème, le 5ème,...

Si le résultat est plus grand que 9, on retranche 9

Après la première étape : 8950 2022 6416 589

On calcule la somme de tous les chiffres

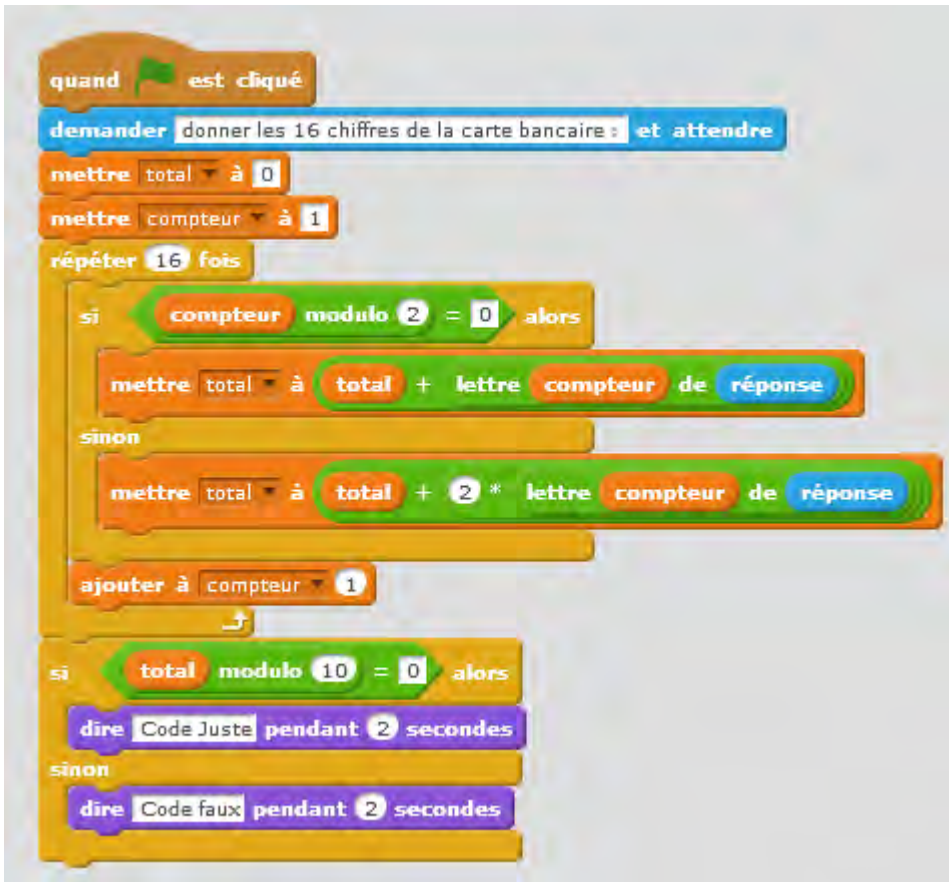
Nombre somme : 67

On prend le reste dans la division par 10

Puis on calcule 10 - résultat

Clé de la carte : 3

Numéro complet de la carte : 4970 1012 3456 7893



Variables
Boucles
Instructions conditionnelles

Avec Excel

Rang	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	Clé
Numéro	4	9	7	0	1	0	1	2	3	4	5	6	7	8	9	3
	8	9	5	0	2	0	2	2	6	4	1	6	5	8	9	67

Idées des formules

C4 fx =SI(C3*2>9;C3*2-9;C3*2)

B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
Rang	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	Clé
Numéro	4	9	7	0	1	0	1	2	3	4	5	6	7	8	9	3
	8	9	5	0	2	0	2	2	6	4	1	6	5	8	9	67

R3 fx =10-MOD(R4;10)

B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
Rang	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	Clé
Numéro	4	9	7	0	1	0	1	2	3	4	5	6	7	8	9	3
	8	9	5	0	2	0	2	2	6	4	1	6	5	8	9	67

Numéro INSEE

Cas	Positions	Signification
Tous	1	sexe : 1 pour les hommes, 2 pour les femmes, 3 pour les personnes étrangères de sexe masculin en cours d'immatriculation en France ^A , 4 pour les personnes étrangères de sexe féminin en cours d'immatriculation en France ^A
	2 et 3	deux derniers chiffres de l'année de naissance (ce qui donne l'année à un siècle près)
	4 et 5	mois de naissance
A	6 et 7	département de naissance métropolitain (2A ou 2B pour la Corse) ^C
	8, 9 et 10	code officiel de la commune de naissance ^{C-D}
B	6, 7 et 8	département de naissance en outre-mer ^C
	9 et 10	deux chiffres du code commune de naissance ^{C-D}
C	6 et 7	naissance hors de France ^C
	8, 9 et 10	identifiant du pays de naissance ^C
Tous	11, 12 et 13	numéro d'ordre de la naissance dans le mois et la commune (ou le pays) ^{D-E}

Calcul de la clé de contrôle : faire la division entière (ou Euclidienne) par 97 du nombre formé par les 13 premiers chiffres.

Vous obtenez un reste que vous soustrayez de 97.

Cette différence est la clé de contrôle.

Pour la Corse, les lettres A et B sont remplacées par des zéros, et on soustrait du nombre à 13 chiffres ainsi obtenu 1 000 000 pour A et 2 000 000 pour B avant d'effectuer la division. On peut aussi, plus simplement, remplacer 2A par 19 et 2B par 18.

Sans la Corse

```
quand [drapeau] est cliqué
demander [Code INSEE:] et attendre
dire [regroupe La clé est: 97 - réponse modulo 97] pendant 2 secondes
```

Avec la Corse

```
quand [drapeau] est cliqué
supprimer l'élément [tout] de la liste [Code]
répéter 13 fois
  demander [Numéro:] et attendre
  ajouter [réponse] à [Code]
si [élément 7 de [Code] = A] alors
  remplacer l'élément 6 de la liste [Code] par 1
  remplacer l'élément 7 de la liste [Code] par 9
si [élément 7 de [Code] = B] alors
  remplacer l'élément 6 de la liste [Code] par 1
  remplacer l'élément 7 de la liste [Code] par 8
mettre [total] à 0
mettre [compteur] à 13
répéter 13 fois
  mettre [total] à [total + élément [compteur] de [Code] * 10^ de 13 - [compteur]]
  ajouter à [compteur] -1
dire [97 - total modulo 97]
```

Variables
Boucles
Instructions conditionnelles

Avec Excel

Rang	1	2	3	4	5	6	7	8	9	10	11	12	13	Clé
Code	1	7	2	0	9	1	6	0	1	5	1	0	2	44
	1	7	2	0	9	1	6	0	1	5	1	0	2	Code
	1E+12	7E+11	2E+10	0	900000000	10000000	6000000	0	10000	5000	100	0	2	1720916015102

Idées des formules

fx =SI(I3="A";9;SI(I3="B";8;I3))

	B	C	D	E	F	G	H	I
Rang	1	2	3	4	5	6	7	
Code	1	7	2	0	9	1	6	
	1	7	2	0	9	1	6	
	1E+12	7E+11	2E+10	0	900000000	10000000	6000000	

fx =C4*PUISSANCE(10;13-C2) fx =97-MOD(P5;97)

	B	C	D	E	D	E	F	G	H	I	J	K	L	M	N	O	P
Rang	1	2	3		2	3	4	5	6	7	8	9	10	11	12	13	Clé
Code	1	7	2		7	2	0	9	1	6	0	1	5	1	0	2	44
	1	7	2		7	2	0	9	1	6	0	1	5	1	0	2	Code
	1E+12	7E+11	2E+10		7E+11	2E+10	0	9E+08	1E+07	6E+06	0	10000	5000	100	0	2	1720916015102

TECHNIQUE

Addition de fractions

```
quand cliqué  
demander premier numérateur et attendre  
mettre n1 à réponse  
demander premier dénominateur et attendre  
mettre d1 à réponse  
demander deuxième numérateur et attendre  
mettre n2 à réponse  
demander deuxième dénominateur et attendre  
mettre d2 à réponse  
mettre d à d1 * d2  
mettre n à n1 * d2 + n2 * d1  
mettre i1 à n  
mettre i2 à d
```

$$\frac{n1 \ 2}{d1 \ 6} + \frac{n2 \ 5}{d2 \ 9} = \frac{i1 \ 48}{i2 \ 54}$$

Variables

Addition en HMS

```
quand cliqué
montrer
demander Heure 1 et attendre
mettre H1 à réponse
demander Minute 1 et attendre
mettre M1 à réponse
demander Seconde 1 et attendre
mettre S1 à réponse
demander Heure 2 et attendre
mettre H2 à réponse
demander Minute 2 et attendre
mettre M2 à réponse
demander Seconde 2 et attendre
mettre S2 à réponse
mettre S3 à S1 + S2
mettre M3 à M1 + M2
mettre H3 à H1 + H2
si S3 > 59 alors
  ajouter à M3 1
  ajouter à S3 -60
si M3 > 59 alors
  ajouter à H3 1
  ajouter à M3 -60
```

Additionv2

$$\begin{array}{r} 6 & 24 & 45 \\ + 3 & 35 & 58 \\ \hline \end{array}$$

Additionv2

$$\begin{array}{r} 6 & 24 & 45 \\ + 3 & 35 & 58 \\ \hline 10 & 0 & 43 \end{array}$$

Variables
Instructions conditionnelles

Résolution d'équation de degré 1



$$a \ 4 \ X + b \ 3 = c \ 3 \ X + d \ 2$$

$$e \ 1 \ X = f \ -1$$

$$X = \frac{g \ -1}{h \ 1}$$

Décomposition en facteurs premiers



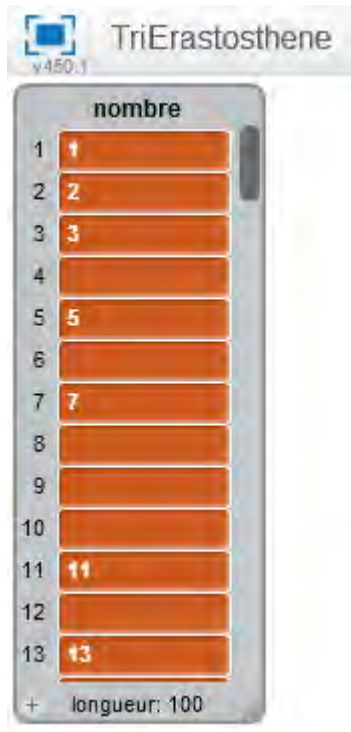
Variables
Boucles
Instructions conditionnelles

Test nombre premier



Variables
Boucles
Instructions conditionnelles

Tri Ératosthène



Variables
Boucles

Calcul de la médiane

donnée

1	3
2	6
3	7
4	9
5	15
6	24

+ longueur: 6

médiane 8

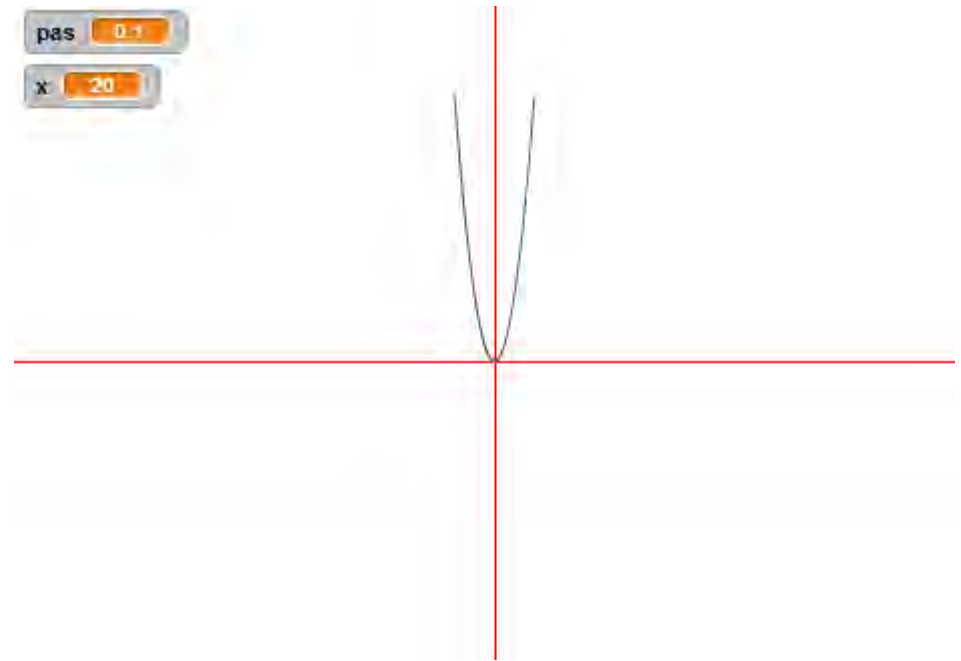
```
quand cliqué
supprimer l'élément tout de la liste donnée
demander nombre de valeurs et attendre
mettre total à réponse
répéter total fois
  demander nombre et attendre
  ajouter réponse à donnée
mettre compteur1 à 1
mettre compteur2 à 2
répéter (total - compteur1) fois
  répéter (total - compteur1) fois
    si (élément compteur1 de donnée > élément compteur2 de donnée) alors
      mettre intermédiaire à élément compteur2 de donnée
      remplacer l'élément compteur2 de la liste donnée par élément compteur1 de donnée
      remplacer l'élément compteur1 de la liste donnée par intermédiaire
      ajouter à compteur2 1
    ajouter à compteur1 1
    mettre compteur2 à compteur1 + 1
  si (total modulo 2 = 1) alors
    mettre médiane à élément (total / 2 + 0.5) de donnée
  sinon
    mettre médiane à (élément (total / 2) de donnée + élément (total / 2 + 1) de donnée) / 2
```

Variables
Boucles
Instructions conditionnelles

Grapheur

```
quand espace est cliqué
effacer tout
mettre la couleur du stylo à 0
dire Dessin des axes pendant 2 secondes
relever le stylo
aller à x: -240 y: 0
dire axe des abscisses pendant 2 secondes
stylo en position d'écriture
aller à x: 240 y: 0
relever le stylo
aller à x: 0 y: -180
dire axe des ordonnées pendant 2 secondes
stylo en position d'écriture
aller à x: 0 y: 180

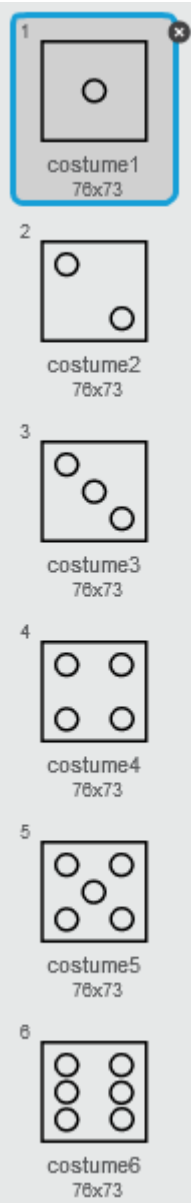
quand cliqué
relever le stylo
choisir la couleur pour le stylo
aller à x: 0 y: 0
mettre pas à 0,1
mettre x à -20
répéter jusqu'à x > 20
  aller à x: x y: 1 / 3 * x * x
  stylo en position d'écriture
  ajouter à x pas
```



Variables
Boucles

VISUALISATION

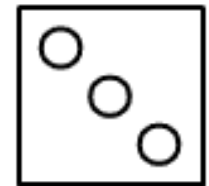
Simulation de lancer de dé (Avec liste)



```
quand cliqué
supprimer l'élément tout de la liste Tirage
demander Nombre de tirages désirés : et attendre
répéter réponse fois
  répéter 10 fois
    mettre valeur à nombre aléatoire entre 1 et 6
    basculer sur costume valeur
    attendre 0.2 secondes
  ajouter valeur à Tirage
  attendre 0.5 secondes
```

Tirage	
1	1
2	3
3	5
4	5
5	1
6	3

+ longueur: 6



Boucles

CHANCES

Somme de deux dés

```
quand est cliqué
  cacher
  effacer tout
  mettre la taille du stylo à 3
  supprimer l'élément tout de la liste compteur
  répéter 12 fois
    ajouter 0 à compteur
  répéter 1000 fois
    mettre dé1 à nombre aléatoire entre 1 et 6
    mettre dé2 à nombre aléatoire entre 1 et 6
    mettre resultat à dé1 + dé2
    aller à x: -80 + 10 * resultat y: -150 + 1 * élément resultat de compteur
    remplacer l'élément resultat de la liste compteur par élément resultat de compteur + 1
    stylo en position d'écriture
    mettre la couleur du stylo à 60 + 15 * resultat
    aller à x: -80 + 10 * resultat y: -149 + 1 * élément resultat de compteur
    relever le stylo
```



Variables
Boucles
Instructions conditionnelles


CHANCES

Simulation de lancer de pièces



Scratch script for simulating coin tosses:

- quand cliqué
- supprimer l'élément tout de la liste Tirage
- demander Nombre de tirages : et attendre
- mettre à 150 % de la taille initiale
- répéter réponse fois
 - aller à x: 0 y: -100
 - basculer sur costume nombre aléatoire entre 1 et 6
 - mettre retournement à 6 * nombre aléatoire entre 3 et 6
 - répéter retournement fois
 - costume suivant
 - ajouter 6 à y
 - répéter retournement fois
 - costume suivant
 - ajouter -6 à y
 - si costume n° < 4 alors
 - basculer sur costume Face2
 - ajouter Face à Tirage
 - sinon
 - basculer sur costume Pile
 - ajouter Pile à Tirage
 - attendre 1 secondes



Results of the simulation:

- 1 Face2 41x41
- 2 Face 40x32
- 3 Face3 40x32
- 4 Pile 41x40
- 5 Pile2 40x33
- 6 Pile3 40x33



Tirage

1	Face
2	Face
3	Pile
4	Face
5	Face
6	Face
7	Face
8	Pile
9	Pile
10	Face

+ longueur: 10



Boucles
Instructions conditionnelles

Jeu lancer de pièces

JeuTiragePiecePipe

```
quand [drapeau] est cliqué
supprimer l'élément tout de la liste Tirage
répéter 3 fois
  demander Pile (P) ou Face (F) et attendre
  mettre choix à réponse
  si choix = F alors
    mettre choix à 1
  sinon
    mettre choix à 4
  mettre à 150 % de la taille initiale
  aller à x: 0 y: -100
  basculer sur le costume nombre aléatoire entre 1 et 6
  mettre retournement à 6 * nombre aléatoire entre 3 et 6
  répéter retournement fois
    costume suivant
    ajouter 6 à y
  répéter retournement fois
    costume suivant
    ajouter -6 à y
  basculer sur le costume choix
  attendre 1 secondes
  si choix = costume # alors
    ajouter Gagné à Tirage
  sinon
    ajouter Perdu à Tirage
mettre compteur à 1
```

```
répéter 7 fois
  demander Pile (P) ou Face (F) et attendre
  mettre choix à réponse
  si choix = F alors
    mettre choix à 1
  sinon
    mettre choix à 4
  mettre à 150 % de la taille initiale
  aller à x: 0 y: -100
  basculer sur le costume nombre aléatoire entre 1 et 6
  mettre retournement à 6 * nombre aléatoire entre 3 et 6
  répéter retournement fois
    costume suivant
    ajouter 6 à y
  répéter retournement fois
    costume suivant
    ajouter -6 à y
  si compteur = 4 alors
    basculer sur le costume choix
    ajouter Gagné à Tirage
  sinon
    si choix = 1 alors
      basculer sur le costume Pile
      ajouter Perdu à Tirage
    sinon
      basculer sur le costume Face2
      ajouter Perdu à Tirage
  ajouter à compteur 1
```

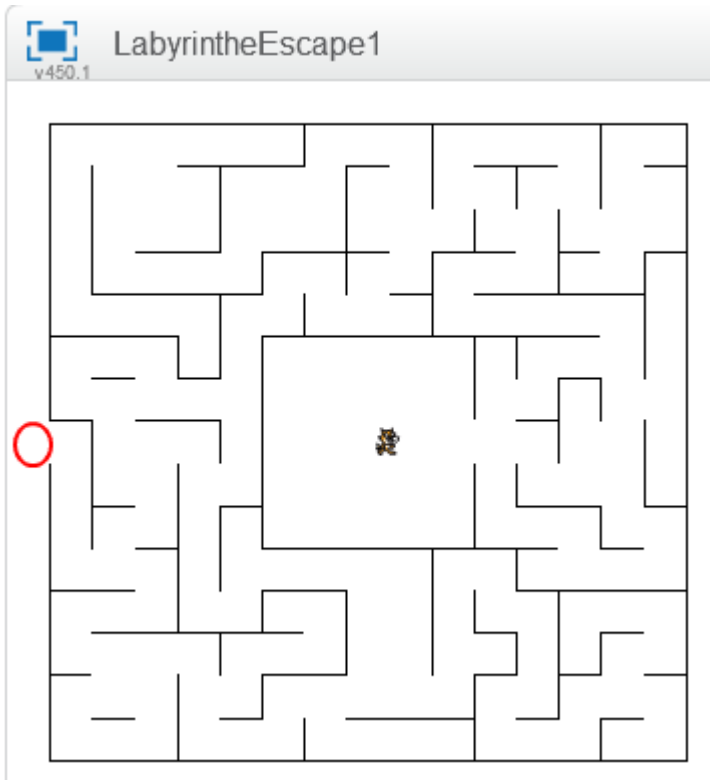


Variables
Boucles
Instructions conditionnelles

CHANCES

JEUX

Labyrinthe



```
quand [drapeau] est cliqué
  aller à x: -50 y: 0
  s'orienter à 90
  mettre à 13 % de la taille initiale
  répéter indéfiniment
    si [touche flèche haut] pressée? alors
      s'orienter à 0
      avancer de 2
    si [touche flèche bas] pressée? alors
      s'orienter à 180
      avancer de 2
    si [touche flèche droite] pressée? alors
      s'orienter à 90
      avancer de 2
    si [touche flèche gauche] pressée? alors
      s'orienter à -90
      avancer de 2
    si [couleur noire] touchée? alors
      aller à x: -50 y: 0
      dire Touché pendant 1 secondes
    si [couleur rouge] touchée? alors
      dire Gagné
```

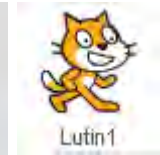
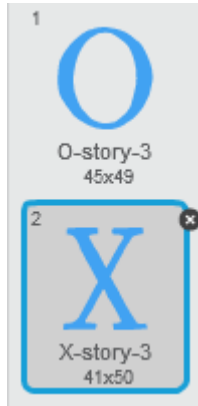
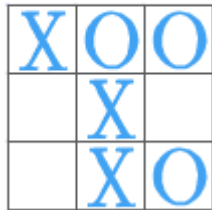
Boucles
Instructions conditionnelles

Morpion



morpion

v450.1



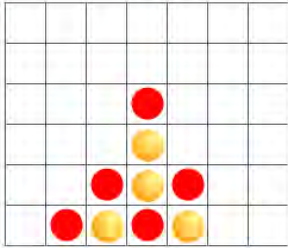
```
quand [drapeau vert] est cliqué
  cacher
  aller à x: -200 y: 160
  effacer tout
  mettre la taille du stylo à 1
  répéter 3 fois
    répéter 3 fois
      stylo en position d'écriture
      Carré
      relever le stylo
      ajouter 34 à x
    ajouter -34 à y
  ajouter -102 à x
  envoyer à tous message1
```

```
définir Carré
  mettre la couleur du stylo à [noir]
  s'orienter à 90
  répéter 4 fois
    avancer de 34
    tourner de 90 degrés
```

```
quand je reçois message1
  répéter 9 fois
    mettre à 60 % de la taille initiale
    demander Colonne et attendre
    mettre colonne à réponse
    demander Ligne et attendre
    mettre ligne à réponse
    s'orienter à 90
    aller à x: -216 + 34 * colonne y: 176 - 34 * ligne
    estampiller
    costume suivant
```

```
quand [drapeau vert] est cliqué
  cacher
```

Boucles
Variables
Instructions conditionnelles
Scripts en parallèle

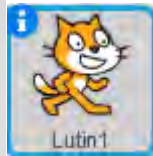


Puissance 4

Colonne

```

quand est cliqué
supprimer l'élément tout de la liste Hauteur
répéter 7 fois
    ajouter 1 à Hauteur
cacher
aller à x: -200 y: 160
effacer tout
mettre la taille du stylo à 1
répéter 6 fois
    répéter 7 fois
        stylo en position d'écriture
        Carré
        relever le stylo
        ajouter 34 à x
    ajouter -34 à y
    ajouter -238 à x
envoyer à tous message1
    
```

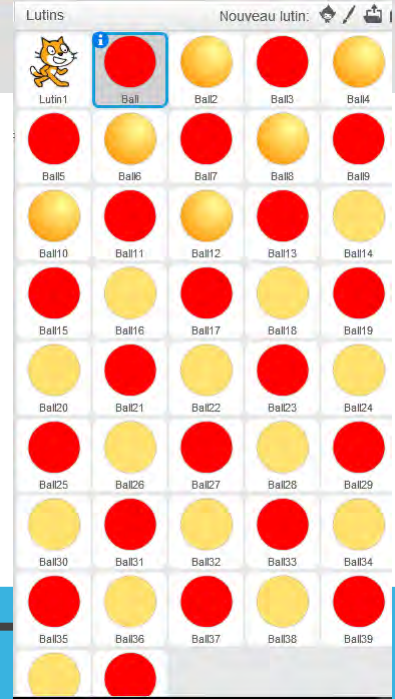


```

quand je reçois message1
mettre à 60 % de la taille initiale
demander Colonne et attendre
s'orienter à 90
aller à x: -216 + 34 * réponse y: -62 + 34 * élément réponse de Hauteur
remplacer l'élément réponse de la liste Hauteur par élément réponse de Hauteur + 1
montrer
envoyer à tous Jeton2
    
```

```

quand est cliqué
cacher
    
```

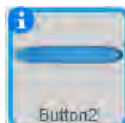


Boucles
Scripts en parallèle
Listes

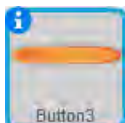


Pong

```
quand [drapeau] est cliqué
mettre direction à nombre aléatoire entre 1 et 90
s'orienter à direction
répéter indéfiniment
  avancer de 6
  rebondir si le bord est atteint
  si [Button2] touché? alors
    si direction > 90 alors
      s'orienter à 180 - direction
    sinon
      s'orienter à -180 - direction
  si [Button3] touché? alors
    si direction > 0 alors
      s'orienter à 180 - direction
    sinon
      s'orienter à -180 - direction
```



```
quand [flèche droite] est pressé
  avancer de 10
quand [flèche gauche] est pressé
  avancer de -10
```



```
quand [z] est pressé
  avancer de 10
quand [a] est pressé
  avancer de -10
```



Boucles
Instructions conditionnelles
Scripts en parallèle

Jeu de Tir



Scratch script for the Ball object:

- when green flag is clicked: hide
- when I receive message1: go to x: abscisse y: ordonnée + 37, set rotation to 0, show, repeat until ordonnée y > 180: advance 5, if Bat2 touched? then say Gagné pendant 1 secondes, hide



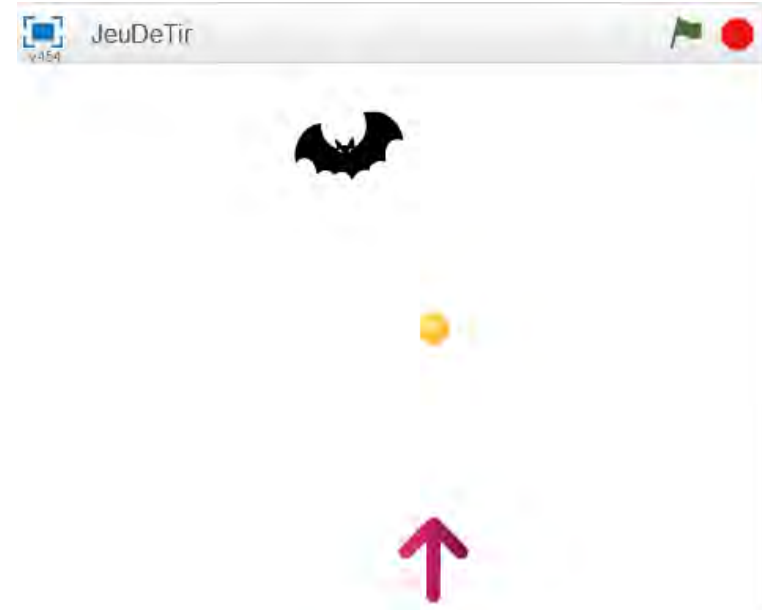
Scratch script for the Bat2 object:

- when green flag is clicked: repeat indefinitely: advance 2, bounce if edge is reached



Scratch script for the Arrow1 object:

- when right arrow is pressed: advance 10, set ordonnée to ordonnée y, set abscisse to abscisse x
- when left arrow is pressed: advance -10, set ordonnée to ordonnée y, set abscisse to abscisse x
- when space is pressed: send message1 to all



Boucles
Variables
Instructions conditionnelles
Scripts en parallèle

Cadavre exquis

```
définir Sujet
supprimer l'élément tout de la liste sujet
ajouter Georges à sujet
ajouter Berthe à sujet
ajouter Milou à sujet
ajouter Le plombier à sujet
```

sujet		verbe		prédictat		Choix	
1	Georges	1	mange	1	des pommes.	1	2
2	Berthe	2	caresse	2	un chat.	2	1
3	Milou	3	ronge	3	un os.	3	4
4	Le plombier	4	répare	4	le lavabo.		
+ longueur: 4		+ longueur: 4		+ longueur: 4		+longueur: 3	

```
quand est cliqué
supprimer l'élément tout de la liste Choix
Sujet
Verbe
Prédictat
répéter 3 fois
  demander Donner un nombre entre 1 et 4 et attendre
  ajouter réponse à Choix
dire regroupe élément élément 1 de Choix de sujet regroupe élément élément 2 de Choix de verbe élément élément 3 de Choix de prédictat
```



Boucles
Variables

Brochure dans laquelle vous pouvez retrouver ces idées

Algorithmique et programmation AU CYCLE 4 à partir des grandeurs

fenêtre de commande
rep 5 [av 100; tg 144]

Convertisseur Euro

Valeur en Euro :
Valeur en Franc :
Valeur en Dollar :
Valeur en Livre :

Remette à zéro Convertir

Nombre de segments :
Valeur du point de vue :
Horizon : 0-1 Longueur : 0-1

Scratch script:

```
afficher tout  
demander nombre de côtés et attendre  
nombre de côtés  
style en position d'écriture  
mettre à jour  
boucle de 10  
boucle de 10  
chercher
```

B	C	D
1	2	
1	7	
1	7	
1E+12	7E+11	

Université de Poitiers

Avril 2017

INSTITUT DES SCIENCES FONDAMENTALES ET APPLIQUÉES UNIVERSITÉ DE POITIERS

IREM de Poitiers • Groupe collège

IREM de Poitiers

MISE EN ŒUVRE

QUELLES CONDITIONS POUR QUELLE PROGRESSION?

On ne peut pas faire sans tenir compte du nombre d'élèves et du matériel disponible .
Le matériel est un élément dont il est question dans les documents d'accompagnement
du cycle 3 .

Initiation à la programmation aux cycles 2 et 3

Introduction

L'initiation à la programmation constitue une nouveauté importante pour les cycles 2 et 3. Elle s'inscrit dans les objectifs du socle commun de connaissances, de compétences et de culture, où il est précisé, dans le domaine 1 (Les langages pour penser et communiquer) :

« [L'élève] sait que des langages informatiques sont utilisés pour programmer des outils numériques et réaliser des traitements automatiques de données. Il connaît les principes de base de l'algorithmique et de la conception des programmes informatiques. Il les met en œuvre pour créer des applications simples. ». Il s'agit aux cycles 2 et 3 d'amorcer un travail qui sera poursuivi au cycle 4.

L'initiation à la programmation apparaît dans les programme au sein du thème Espace et géométrie en lien avec l'objectif « (Se) repérer et (se) déplacer en utilisant des repères » au cycle 2 et « (Se) repérer et (se) déplacer dans l'espace en utilisant ou en élaborant des représentations » au cycle 3.

La diversité des équipements sur le territoire nécessite de s'appuyer sur des activités faisant appel des supports variés :

- sans matériel spécifique, « en débranché » ;
- des robots programmables ;
- des applications en ligne utilisables sur ordinateurs ou tablettes ;
- des logiciels pouvant être installés sur des ordinateurs ou des tablettes.

L'initiation à la programmation pourra être une opportunité pour des travaux interdisciplinaires :

- avec le champ questionner le monde au cycle 2, par exemple, autour de la question du repérage ;
- ou avec le français, dans le développement des usages du langage oral ou écrit, notamment en créant des histoires illustrées par de courtes animations créées par les élèves ;
- ou encore en langues vivantes en créant des animations où les personnages dialoguent en langue étrangère ;
- etc.

Pour cette enseignement, on peut penser à des méthodes différentes comme :

Première méthode :

Une présentation en classe entière du logiciel pour montrer les fonctionnalités pendant une vingtaine de minutes.

Puis en salle informatique, en groupe, avec un élève par poste, on propose un parcours qui permet de se servir des différentes fonctionnalités entrevues pour en prendre possession.

Par exemple :

1. Changer le lutin et programmer la possibilité de déplacer le lutin au clavier (flèche du haut pour se déplacer vers le haut ...)
2. En cliquant sur le drapeau le lutin doit faire un tour de l'écran, rapetissant à chaque changement de direction.
3. Programmer le « vol » de la chauve-souris qui devra parcourir la diagonale de l'écran.
4. Faire se rencontrer deux lutins qui devront se parler.
5. Construire un jeu de pong (suivant l'avancée de chacun).

Deuxième méthode :

1h30 d'« exploration libre » sur le logiciel puis petit bilan et présentation de la variable à partir de difficultés rencontrés par deux élèves (l'un voulant faire un vaisseau dont le tir bougeait avec le vaisseau -utilité de la variable pour stocker l'abscisse du « tir » au moment de l'action- l'autre voulant « faire » un score)

- pendant 3 séances et par groupes de 2 (certains ayant poursuivi le travail au CDI) programmation, avec mon aide si besoin, d'une animation ou d'un jeu avec les contraintes suivantes :

le thème à choisir parmi les suivants (Labyrinthe ; Voyage en montgolfière ; Décollage d'avion ; Poisson dans l'océan ; Icare) – *pour donner une idée de départ à ceux qui n'en auraient pas et un cadre à ceux susceptibles de partir dans toutes les directions en même temps.*

les « figures imposées » : un lutin qui bouge avec le clavier ; au moins un positionnement (x ; y) ; au moins une interaction (Si ... alors ...) ; une variable (score, comptage, autre ...) ; un chronomètre si c'est utile.

- évaluation (2,5 séances)

Chaque élève a présenté son travail devant la classe qui a, avec le professeur, évalué l'animation selon les critères suivants : présentation orale du travail ; fonctionnement correct de l'animation ; technicité (figures imposées) ; conformité au thème choisi ; originalité de la réalisation ; réponses aux demandes d'explications.

Troisième méthode:

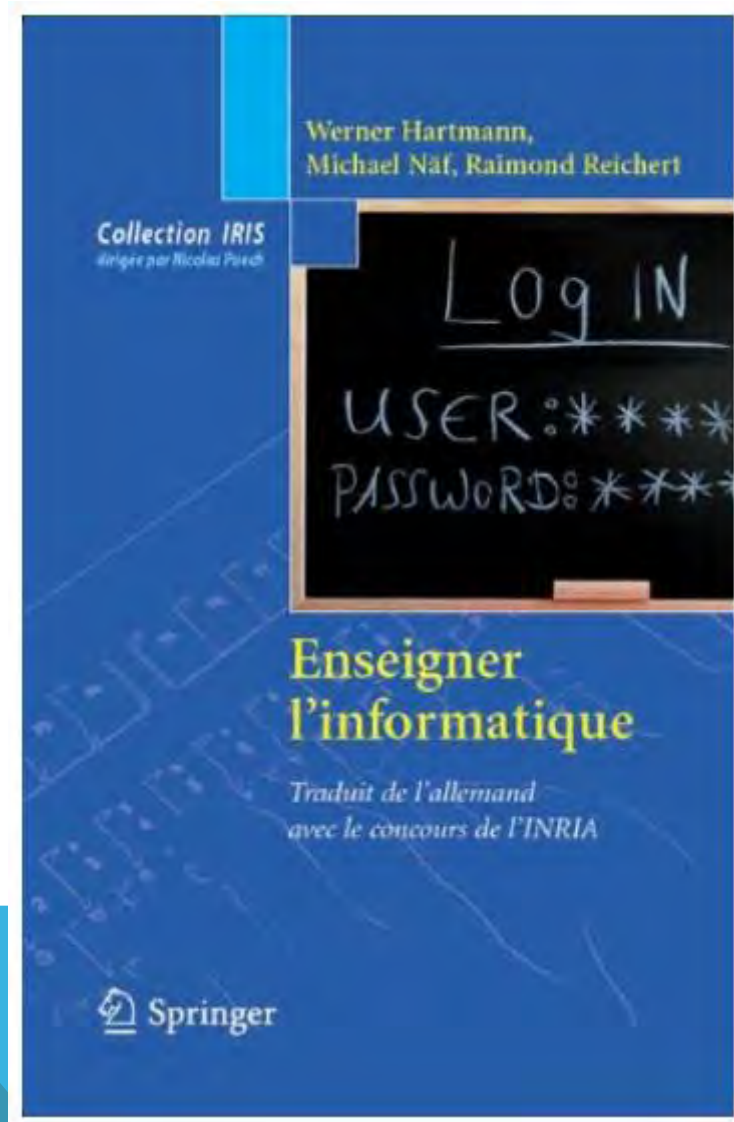
Séances par thème avec la classe entière en salle informatique (2 élèves par poste) ou avec une demi-classe (1 élève par poste)

La progression est prévu avec 6 thèmes :

- **Thème 1** : Déplacement d'un objet, contrôle avec les flèches et construction d'un labyrinthe
- **Thème 2** : Travail sur les orientations et les boucles (tracés de polygone régulier selon des méthodes différentes) jusqu'au tracé de polygones étoilés avec un nombre impair de côté.
- **Thème 3** : Les blocs et leur manipulation. On retrouve les transformations qui doivent être étudiées. En fin, il faut construire un quadrillage 3×3 avec le bloc.
- **Thème 4** : Travail sur la « variable » réponse. On demande le prénom et le personnage dit « bonjour prénom ». On demande l'année de naissance et il donne l'âge. On construit un programme qui donne une multiplication à faire, on répond et il vérifie la réponse (arrivée des variables).
- **Thème 5** : Communication entre objets (Script en parallèle). Il faut gérer 3 lutins et leurs interactions.
- **Thème 6** : Gestion des listes. Je fournis un fichier contenant une liste de pays et de capitales. Il faut créer un bloc qui permet d'invertir deux lignes pour qu'ensuite, à l'aide de cette instruction, on puisse remettre les capitales avec leur pays.

Les différences sont sur le degré de liberté des élèves mais moins ils sont nombreux, plus il est facile de les accompagner dans leurs projets.

Dans ce livre



Les différentes pédagogies :

Pédagogie expérientielle

La méthode expérientielle est une méthode pédagogique adaptée à de courtes séances de formation d'une durée de vingt à trente minutes.

La matière, déjà traitée est approfondie et élargie au cours de nouvelles expériences.

Dans l'enseignement de l'informatique, la méthode expérientielle convient à l'organisation d'exercices individuels sur l'ordinateur.

Programme dirigé

Le programme dirigé peut être considéré comme une extension de la méthode expérientielle. Il s'agit de proposer un matériel d'autoapprentissage pour une durée de deux à dix leçons. Les programmes dirigés se basent sur le principe du *mastery learning* ou pédagogie de la maîtrise: l'étudiant doit parfaitement comprendre un sujet avant de passer au suivant. Pour les plus rapides, un programme dirigé propose un complément. Dans l'enseignement de l'informatique, les programmes dirigés sont adaptés à des groupes d'apprentissage hétérogènes ou à des thèmes abstraits particulièrement difficiles.

Travail en groupe

Le travail en groupe est une méthode d'enseignement très répandue dans le monde entier. La sociabilité occupe ici une place prépondérante par rapport à la communication et la compréhension de la substance elle-même. Le travail en groupe est exigé dans de nombreuses professions de l'informatique et cette méthode d'enseignement permet d'acquérir les compétences et l'expérience nécessaires à cet effet. Cependant, le travail en groupe peut également être utilisé pour des sujets qui ne peuvent être traités que de manière ponctuelle et non pas par une approche globale. La division d'un sujet en différents groupes, également appelée mosaïque de groupes, est une méthode d'enseignement bien adaptée ici.

Apprentissage par la découverte

L'enseignement de l'informatique se caractérise souvent par la transmission d'une théorie suivie d'exercices pratiques. Des aspects importants tels que le travail autonome, la créativité et la réflexion critique sont peu pris en compte ici. L'apprentissage par la découverte met explicitement l'accent sur ces aspects.

Pédagogie de projet

Les projets occupent une position dominante dans le domaine de l'informatique. Il est donc tout naturel d'appliquer la pédagogie de projet dans l'enseignement de l'informatique. Les étudiants s'attaquent à un problème ou au développement d'un produit en collaboration avec un ou plusieurs enseignants. Les problèmes à résoudre sont ensuite déterminés, un plan de projet est établi et le travail est finalement réalisé, tout ceci en commun. Les projets conviennent pour donner un aperçu de la nature de systèmes informatiques complexes.

Enseigner l'informatique par Hartmann, Näf et Reichert
Chez Springer

Que doivent retenir de notre enseignement les élèves ?

Connaissance des produits :

- Liée au produit
- Court terme
- Apprentissage par cœur
- Faits isolés
- Peu de transfert possible
- Concret

Connaissance des concepts :

- Indépendant du produit
- Long terme
- Compréhension et structuration
- Relations
- Transfert possible
- Abstrait

Enseigner l'informatique par Hartmann, Näf et Reichert
Chez Springer

Exemple de bilan de fin d'année

Programmation et algorithmique : A retenir

1. Les variables

3 sortes dans Scratch :

a. Dans **Capteurs**

réponse qui est le résultat de **demander et attendre**

b. Dans **Données**

Créer une variable permet d'avoir **variable**

c. Dans **Données**

Créer une liste permet d'avoir une liste de variable **liste**

2. Les conditionnelles

Dans **Contrôle**



Il s'agit de **si...alors...sinon** qui est parfois limitée à



Les conditions sont repérées par des blocs hexagonaux par exemple **moins que** ou **couleur touchée?**

que l'on trouve dans **Capteurs** ou **Opérateurs**

3. Les boucles

Elles sont de deux sortes :


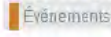
Dans **Contrôle**, **répéter 10 fois** qui est une boucle répétitive

et **répéter jusqu'à** qui est une boucle conditionnelle que l'on retrouve dans d'autres langages sous la forme Tant Que

Spécificités de Scratch

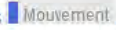



On utilise des lutins , il s'agit de « personnage ».



Chaque script doit commencer par un bloc de cette forme  que l'on trouve dans , il y a 6 choix possibles.

1. Déplacement :


Pour se repérer, chaque objet est repéré ses coordonnées en pixel, l'origine du repère étant au centre de l'image.

On peut déplacer les lutins à l'aide des instructions que l'on trouve dans .

On ne peut qu'  dans l'orientation du lutin, pour reculer, il faudra utiliser les nombres négatifs.


On peut orienter de 2 façons le lutin : à l'aide de  ou , le premier est une rotation relativement à sa position de départ, le deuxième étant une orientation absolue, c'est à dire que l'angle ne dépend que de position par rapport au Nord.

2. Tracé :


On peut faire dessiner des figures à l'aide des instructions contenues dans .

On peut effacer le dessin, choisir la taille et la couleur du crayon et choisir de faire écrire ou pas.

3. Musique :

On peut faire jouer de la musique à l'aide des instructions contenues dans .

4. Habillage :

On peut changer l'apparence du lutin ou le faire parler à l'aide des instructions contenues dans .

5. Pour éviter de répéter un groupe d'instructions, on peut créer un bloc dans , ce qui permet de ne rappeler que le nom du bloc dans le programme.

Exemple : Le premier peut être remplacé par le troisième si l'on a défini « dessin » comme dans le deuxième.



6. Script en parallèle :

On peut avoir plusieurs lutins dans un programme et chacun a son propre script. Ces scripts peuvent être indépendants et chaque lutin « vit sa vie à son rythme » ou on peut les faire interagir à l'aide de

 que l'on trouve dans . Alors un lutin ne peut commencer son action que lorsqu'un autre lui en donne l'ordre.

Pertinence

La carte FNAC coûte 30 € et permet d'avoir 5% de réduction sur certains produits.

A combien me revient le premier article que j'achète s'il coûtait au départ :

30 €, 50 €, 100 €, 300 €, 500 €, 1000 € ?

Donner l'expression de la fonction qui permet de calculer le prix payé en fonction du prix de départ noté x .

VOS AVANTAGES CARTE FNAC

- 5%**
DE REMISE IMMÉDIATE
SUR DE NOMBREUX
PRODUITS
- LIVRAISON GRATUITE**
DES LIVRES
- NOUVEAU**
PASS PARTENAIRES
Jusqu'à 35% de remises
valables chez plus de 50 partenaires
- DEEZER**
DEEZER PREMIUM+
1€/MOIS PENDANT 3 MOIS
puis 9,99€/mois
- PLUS DE 18 000**
SPECTACLES À TARIF
RÉDUIT
- CINÉ CHEQUE**
- GRILL**
courte
paille
- KUONI**
DEPUIS 1906

Voir tous les partenaires

<https://www.fnac.com/choisir-carte>

Première solution : Recherche d'une formule

Pour 30 €

On paie :

Carte : 30 € + Article : 30 € - 5% de 30 €

Donc $30 \text{ €} + 30 \text{ €} - 5\% \times 30 \text{ €} = 30 \text{ €} + 30 \text{ €} - 1,5 \text{ €} = 58,5 \text{ €}$

Mais le raisonnement est à reprendre à chaque nouvelle valeur, sinon on peut traduire le raisonnement par une formule.

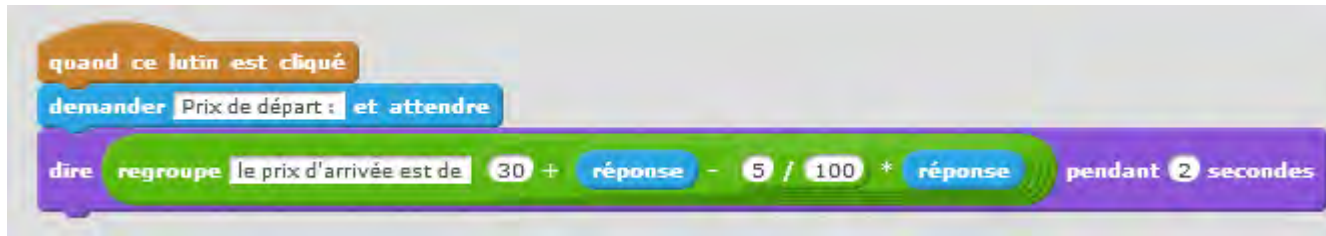
Ce qui donne comme formule :

$30 \text{ €} + \text{Prix} - 5\% \times \text{Prix}$

Ce qui dans le langage des fonctions donne :

$f : x \rightarrow 30 + x - 5\% \times x$

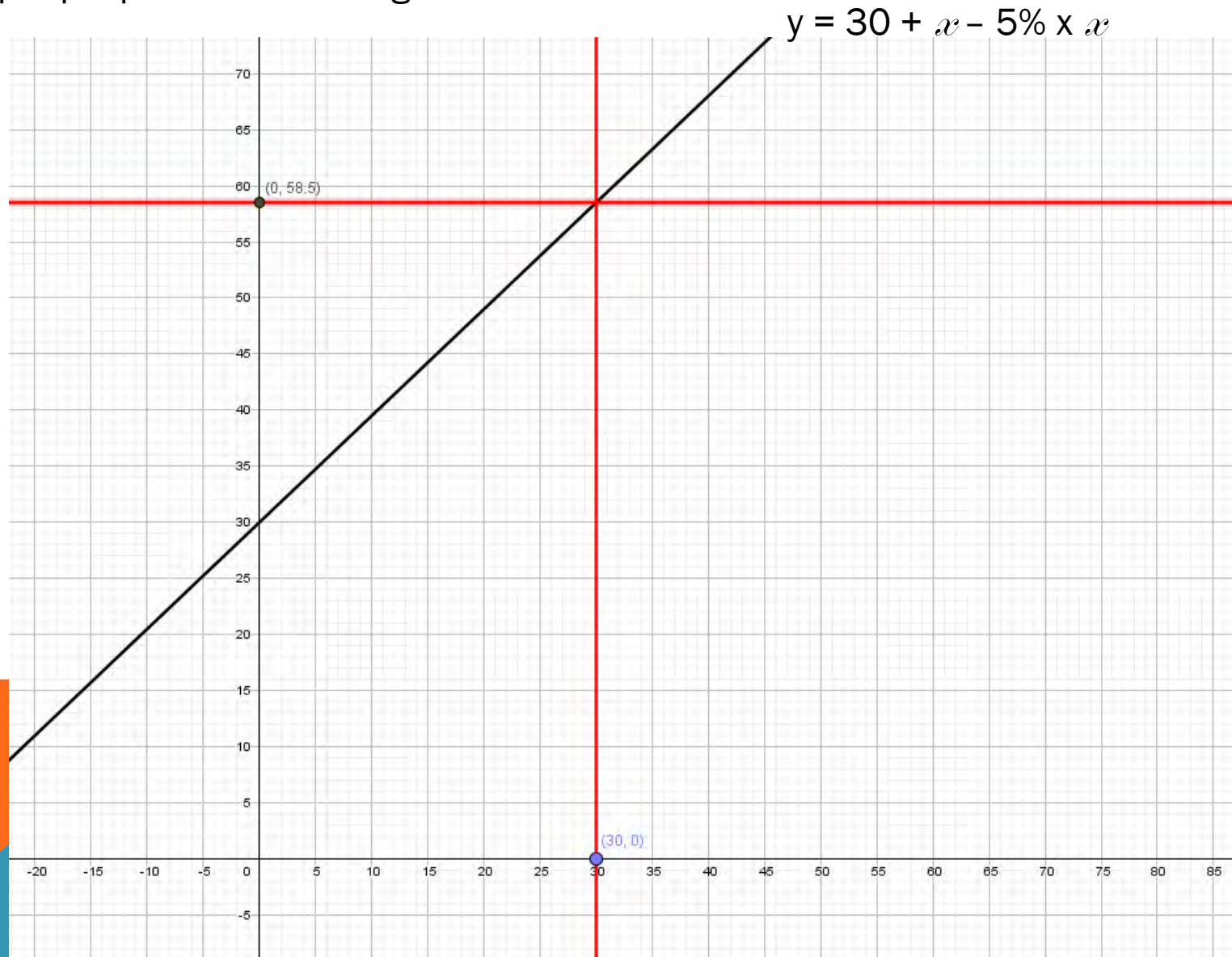
Deuxième solution : avec la programmation



Cela consiste à implémenter la formule

Troisième solution : par la représentation graphique

On peut se servir de l'expression de la fonction pour obtenir sa représentation graphique puis lire une image.



Quatrième solution : par le tableau de valeurs

On peut se servir de l'expression de la fonction pour construire un tableau de valeur puis lire la valeur de l'image.

x	5	10	15	20	25	30	35	40	45	50
f(x)	34,75	39,5	44,25	49	53,75	58,5	63,25	68	72,75	77,5

Les deux derniers cas permettent d'avoir accès à plusieurs valeurs à la fois.

Réponses :

30	50	100	300	500	1000
58,50	77,50	125,00	315,00	505,00	980,00

A partir de quel prix cela devient rentable pour le premier achat ?

Si j'ai payé 171 €, combien coutait l'article au départ ?

Si j'ai payé 779 €, combien coutait l'article au départ ?

Si j'ai payé 1172,3 €, combien coutait l'article au départ ?

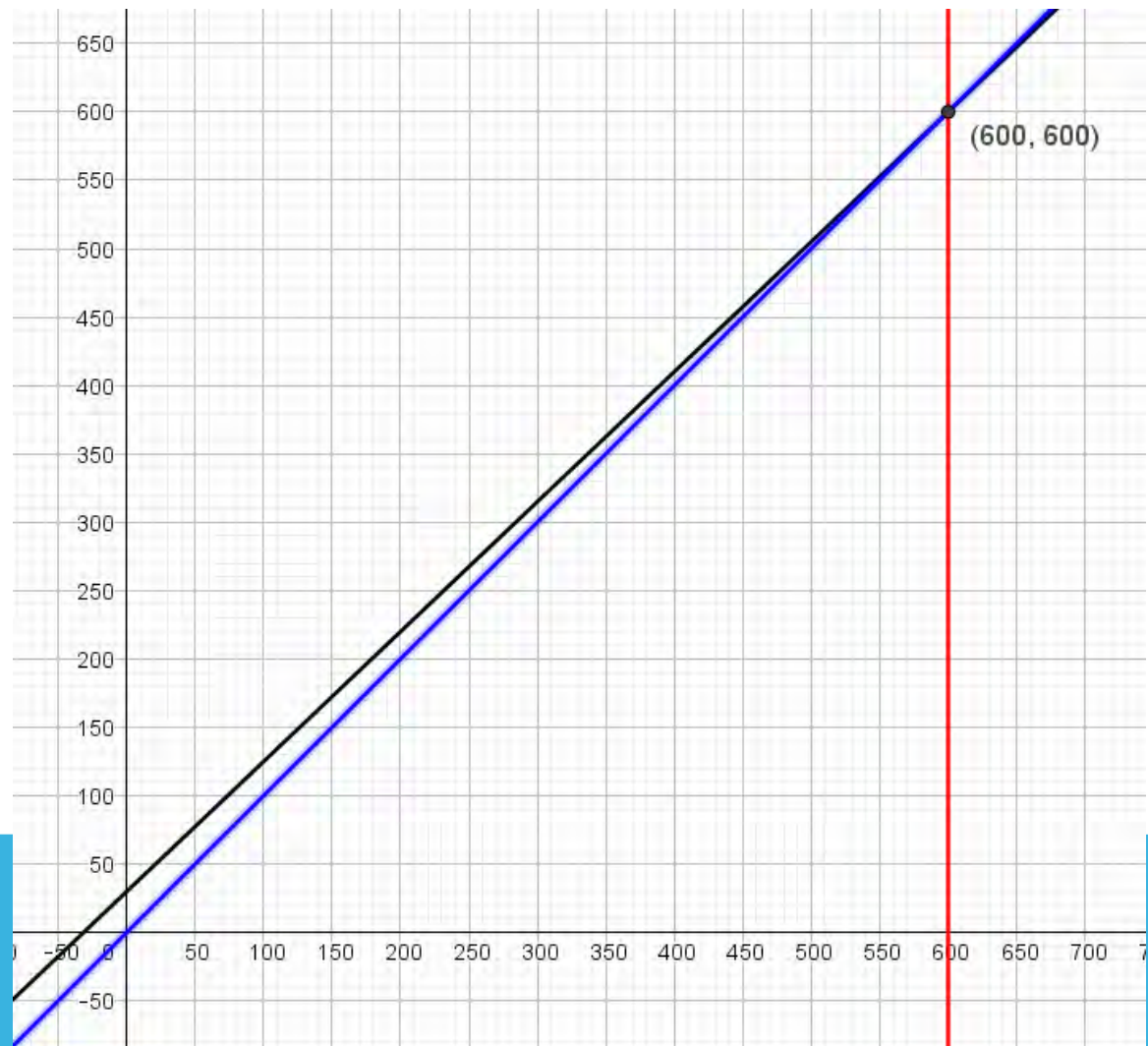
Donner l'expression de la fonction qui permet de calculer le prix départ en fonction du prix payé noté x .



Soit on cherche la solution de :

$$30 \text{ €} + \text{Prix} - 5\% \times \text{Prix} = \text{Prix}$$

Soit on lit graphiquement



Dans les deux cas, cela devient rentable à partir de 600 €.

Pour cette question, les méthodes 3 et 4 reste valable sans changement, il s'agit juste d'une lecture d'antécédent(s).

x	148,4	148,41	148,42	148,43	148,44	148,45	148,46	148,47	148,48
f(x)	170,98	170,99	171,00	171,01	171,02	171,03	171,04	171,05	171,06



Pour la première méthode, il faut résoudre l'équation suivante :

$$\begin{array}{rclcl} 30 \text{ €} & + \text{ Prix} - 5\% \times \text{ Prix} & = & & 171 \text{ €} \\ 30 \text{ €} & + 0,95 \times \text{ Prix} & = & & 171 \text{ €} \\ -30 \text{ €} & & & & -30 \text{ €} \\ & 0,95 \times \text{ Prix} & = & & 141 \text{ €} \\ & : 0,95 & & & : 0,95 \\ & \text{Prix} & \approx & & 148,42 \text{ €} \end{array}$$

Ce qui donne comme formule :

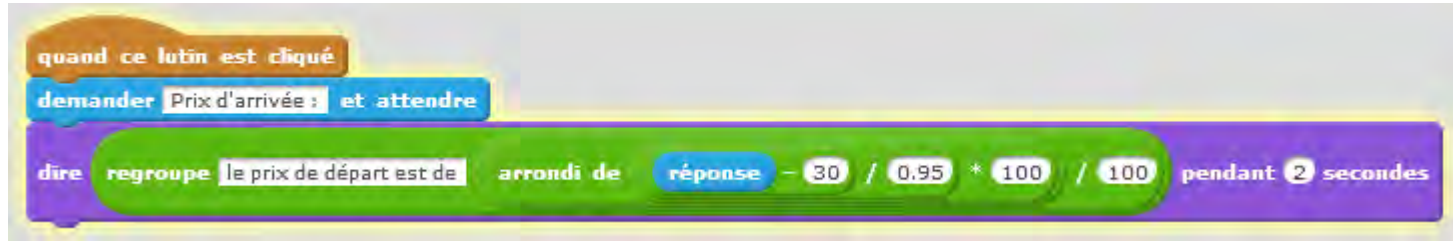
$$(\text{Prix} - 30) : 0,95$$

Ou dans le langage des fonctions

$$g : x \rightarrow (x - 30) : 0,95$$

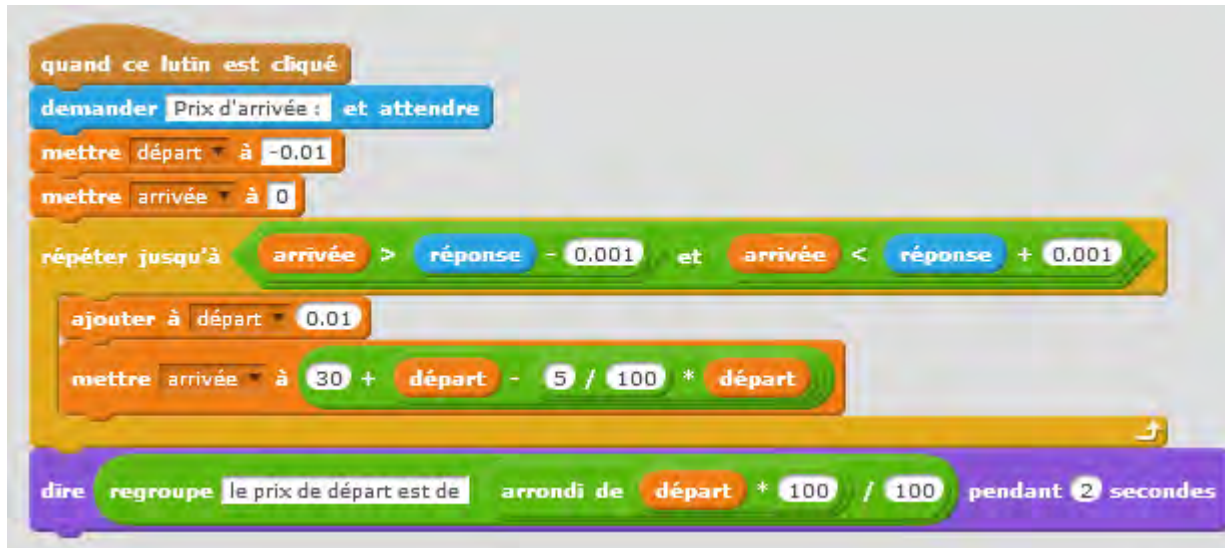
On peut également le programmer avec Scratch et la formule

soit avec l'idée de l'équation



Mais dans ce cas là, il va falloir gérer les arrondis pour avoir une valeur au centime près.

soit avec l'idée du tableur



Mais dans ce cas là, il va falloir gérer « les nombres flottants »

Solutions :

171	779	1172,3
148,42	788,42	1202,42



Diviseur

Objectif :

Analyser et comprendre un programme puis s'en inspirer. Utiliser les listes de variables de Scratch.

Travail :

Voici le programme « PremierouPasBase » qui dit si un nombre est premier ou pas :

```

quand cliqué
  demander Donner un nombre et attendre
  mettre nombre à réponse
  mettre compteur à 2
  répéter jusqu'à nombre modulo compteur = 0
  ajouter à compteur 1
  dire regroupe compteur regroupe est un diviseur de nombre pendant 1 secondes
  si nombre = compteur alors
    dire Ce nombre est premier
  sinon
    dire Ce nombre n'est pas premier
  
```

1. Tester le
Noter les valeurs que vous avez essayé :
2. Expliquer comment il fonctionne en répondant à la question suivante

NB : `nombre modulo compteur` donne le reste de la division euclidienne de `nombre` par `compteur`

Noter les valeurs que prennent nombre et compteur si 'on a donné la valeur 15 au nombre.

Etape	<code>nombre modulo compteur</code>	Nombre	Compteur
1			
2			
3			

Puis reprendre le même travail si l'on avait choisi 35 comme valeur pour le nombre

Etape	<code>nombre modulo compteur</code>	Nombre	Compteur
1			
2			
3			
4			
5			

3. Pourquoi a-t-on mis `mettre compteur à 2` et pas `mettre compteur à 1` ?
.....
4. Pourquoi est-on sûr que le « répéter jusqu'à » va s'arrêter ?
.....
5. En s'inspirant du programme précédent, construire un programme qui demande un nombre et qui dit s'il est pair ou impair.
Sauvegarder le dans le répertoire de la classe/Travail/Maths en mettant votre nom dans le titre du fichier.
6. Construire un programme qui demande deux nombres (le plus grand d'abord) et dit si le deuxième est un diviseur du premier.
Sauvegarder le dans le répertoire de la classe/Travail/Maths en mettant votre nom dans le titre du fichier.
7. En se servant du programme « PremierouPasBase », le transformer pour obtenir tous les diviseurs d'un nombre.

Indications : il faudra créer une liste `Créer une liste` que l'on appellera diviseur,

On aura alors les commandes ci-contre :

Et il faudra modifier le répéter.

```

supprimer l'élément tout de la liste diviseur
ajouter thing à diviseur
  
```

8. Demander deux nombres et trouver leur plus grand diviseur commun (PGCD) et leur plus petit multiple commun (PPCM).

EVALUATION

BREVET PONDICHÉRY 2 MAI 2017

EXERCICE 3

7 POINTS

Créer une variable

- Etape 1
- Etape 2
- Résultat
- x

On considère le programme de calcul ci-contre dans lequel x, Étape 1, Étape 2 et Résultat sont quatre variables.

```
quand [drapeau] est cliqué
  demander "Choisis un nombre." et attendre
  mettre x à réponse
  dire "Je multiplie le nombre par 6." pendant 2 secondes
  mettre Etape 1 à 6 * x
  dire "J'ajoute 10 au résultat." pendant 2 secondes
  mettre Etape 2 à Etape 1 + 10
  dire "Je divise le résultat par 2." pendant 2 secondes
  mettre Résultat à Etape 2 / 2
  dire regroupe "J'obtiens finalement" Résultat
```

1.
 - a. Julie a fait fonctionner ce programme en choisissant le nombre 5. Vérifier que ce qui est dit à la fin est : « J'obtiens finalement 20 ».
 - b. Que dit le programme si Julie le fait fonctionner en choisissant au départ le nombre 7 ?
2. Julie fait fonctionner le programme, et ce qui est dit à la fin est : « J'obtiens finalement 8 ». Quel nombre Julie a-t-elle choisi au départ ?
3. Si l'on appelle x le nombre choisi au départ, écrire en fonction de x l'expression obtenue à la fin du programme, puis réduire cette expression autant que possible.
4. Maxime utilise le programme de calcul ci-dessous :

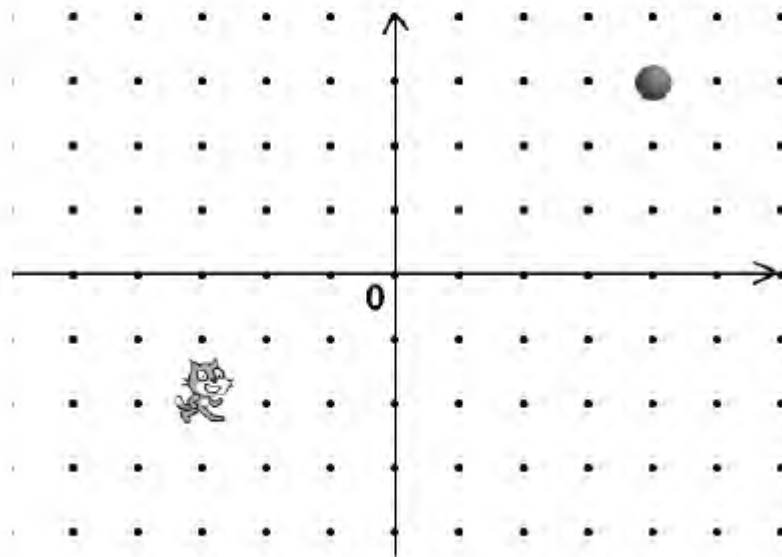
- Choisir un nombre.
- Lui ajouter 2
- Multiplier le résultat par 5

Peut-on choisir un nombre pour lequel le résultat obtenu par Maxime est le même que celui obtenu par Julie ?

BREVET AMÉRIQUE DU NORD 2017

Exercice 5 (5 points)

L'image ci-dessous représente la position obtenue au déclenchement du bloc départ d'un programme de jeu.



L'arrière-plan est constitué de points espacés de 40 unités.

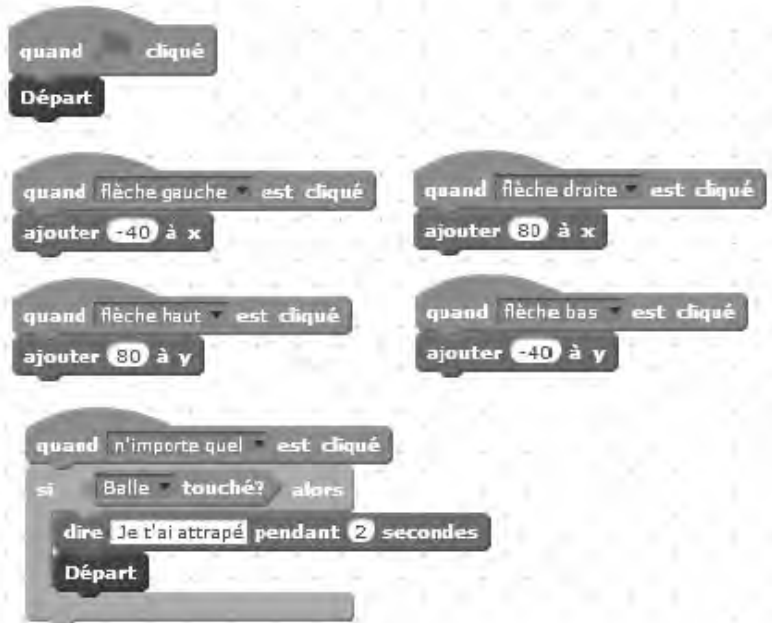
Dans cette position, le chat a pour coordonnées $(-120 ; -80)$.

Le but du jeu est de positionner le chat sur la balle.

- 1) Quelles sont les coordonnées du centre de la balle représentée dans cette position ?
- 2) Dans cette question, le chat est dans la position obtenue au déclenchement du bloc départ.

Voici le script du lutin « chat » qui se déplace.

- a) Expliquez pourquoi le chat ne revient pas à sa position de départ si le joueur appuie sur la touche → puis sur la touche ←.



- b) Le joueur appuie sur la succession de touches suivante : → → ↑ ← ↓

Quelles sont les coordonnées x et y du chat après ce déplacement ?

- c) Parmi les propositions de succession de touches ci-dessous, laquelle permet au chat d'atteindre la balle ?

Déplacement 1	Déplacement 2	Déplacement 3
→ → → → → → ↑ ↑ ↑ ↑ ↑	→ → → ↑ ↑ ↑ → ↓ ←	↑ → ↑ → ↑ → → ↓ ↓

- 3) Que se passe-t-il quand le chat atteint la balle ?

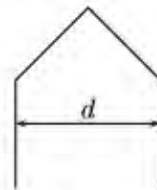
BREVET CENTRE ÉTRANGERS 2017

EXERCICE 6

9 points

Pour tracer une « rue », on a défini le tracé d'une « maison ».

```
définir maison
tourner de 90 degrés
avancer de 50
tourner de 45 degrés
avancer de 50
tourner de 90 degrés
avancer de 50
tourner de 45 degrés
avancer de 50
tourner de 90 degrés
```



tracé de la « maison ».

```
Quand est cliqué
cacher
mettre la taille du stylo à 1
aller à x: -240 y: 0
effacer tout
stylo en position écriture
s'orienter à 90
répéter n fois
  maison
  avancer de 20
```

programme principal

1. Vérifier que d est environ égal à 71 à l'unité près.
2. Un point dans une fenêtre d'exécution de votre programme a son abscisse qui peut varier de -240 à 240 et son ordonnée qui peut varier de -180 à 180 .

Quel est le plus grand, nombre entier n que l'on peut utiliser dans le programme principal pour que le tracé de la « rue » tienne dans la fenêtre de votre ordinateur où s'exécute le programme ?

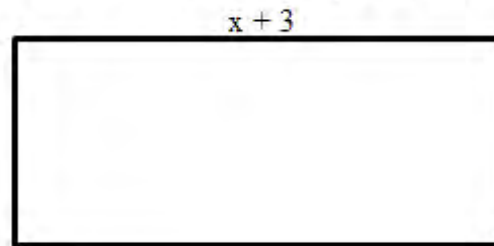


Vous pourrez tracer sur votre copie tous les schémas (à main levée ou non) qui auront permis de répondre à la question précédente et ajouter toutes les informations utiles (valeurs, codages, traits supplémentaires, noms de points ...)

Brevet blanc

Exercice :

On s'intéresse au rectangle ci-dessous



5

1. Que fait le programme ci-dessous pour ce rectangle ? Donner l'interprétation géométrique de ce programme.

```
quand est cliqué
demander Combien pour x ? et attendre
mettre inconnue à réponse
mettre inconnue à inconnue + 3
mettre inconnue à inconnue + 5
mettre inconnue à inconnue * 2
dire inconnue
```

2. Qu'affiche ce programme si on a donné la valeur 6 à x ?
3. Qu'a-t-on donné comme valeur à x si le programme affiche « 24 » ?
4. On voudrait maintenant que ce programme calcule l'aire de ce rectangle. Choisir parmi les propositions suivantes la bonne réponse. Noter la bonne réponse sur votre copie et justifier.

Proposition A :

```
quand est cliqué
demander Combien pour x ? et attendre
mettre inconnue à réponse
mettre inconnue à inconnue + 3
mettre inconnue à inconnue + 5
mettre inconnue à inconnue * inconnue
dire inconnue
```

Proposition B :

```
quand est cliqué
demander Combien pour x ? et attendre
mettre inconnue à réponse
mettre inconnue à inconnue + 3
mettre inconnue à inconnue * 5
dire inconnue
```

Proposition C :

```
quand est cliqué
demander Combien pour x ? et attendre
mettre inconnue à réponse
mettre inconnue à inconnue + 3
mettre inconnue à inconnue * inconnue
dire inconnue
```

Cette année,
on leur demande de compléter le
programme.

Brevet Pondichéry 3 Mai 2018

EXERCICE 5

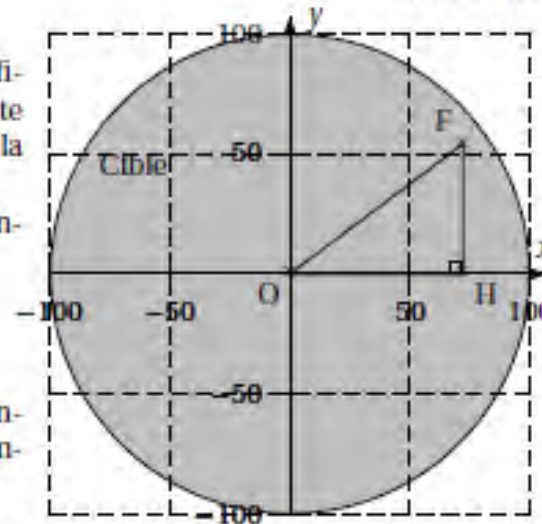
20 POINTS

Dans tout l'exercice l'unité de longueur est le mm.

On lance une fléchette sur une plaque carrée sur laquelle figure une cible circulaire (en gris sur la figure). Si la pointe de la fléchette est sur le bord de la cible, on considère que la cible n'est pas atteinte.

On considère que cette expérience est aléatoire et l'on s'intéresse à la probabilité que la fléchette atteigne la cible.

- La longueur du côté de la plaque carrée est 200.
- Le rayon de la cible est 100.
- La fléchette est représentée par le point F de coordonnées $(x; y)$ où x et y sont des nombres aléatoires compris entre -100 et 100 .



1. Dans l'exemple ci-dessus, la fléchette F est située au point de coordonnées $(72; 54)$.
Montrer que la distance OF , entre la fléchette et l'origine du repère est 90.
2. D'une façon générale, quel nombre ne doit pas dépasser la distance OF pour que la fléchette atteigne la cible?

3. On réalise un programme qui simule plusieurs fois le lancer de cette fléchette sur la plaque carrée et qui compte le nombre de lancers atteignant la cible. Le programmeur a créé trois variables nommées : **carré de OF**, **distance** et **score**.

```
Quand est cliqué
mettre score à 0
répéter 120 fois
  aller à x: nombre aléatoire entre -100 et 100 y: nombre aléatoire entre -100 et 100
  mettre Carré de OF à abscisse x * abscisse x +
  mettre distance à racine de
  si distance < ... alors
    ajouter à score 1
```

- Lorsqu'on exécute ce programme, combien de lancers sont simulés?
 - Quel est le rôle de la variable **score**?
 - Compléter et recopier sur la copie uniquement les lignes 5, 6 et 7 du programme afin qu'il fonctionne correctement.
 - Après une exécution du programme, la variable **score** est égale à 102.
À quelle fréquence la cible a-t-elle été atteinte dans cette simulation?
Exprimer le résultat sous la forme d'une fraction irréductible.
4. On admet que la probabilité d'atteindre la cible est égale au quotient : aire de la cible divisée par aire de la plaque carrée.
Donner une valeur approchée de cette probabilité au centième près.

ANNEXE

LANGAGE ET LOGICIEL

Scratch est une implémentation visuelle et dynamique du langage de programmation Smalltalk fondée sur Squeak. (Récente : 2006).

Cette plateforme d'apprentissage de codage comporte de plus de 14,5 millions de projets partagés allant de l'animation à la simulation 3D. Le nombre d'utilisateurs enregistrés est de 12 millions et ne cesse d'augmenter^[3]

Scratch permet de rendre utilisable en ligne les projets réalisés, au moyen du plugin Adobe Flash Player lequel transforme les projets Scratch en applications web.

Scratch est un environnement de développement, de site web, et un lecteur de scripts Scratch.

Le code source de Scratch est encore en développement. Il est distribué sous deux licences dont la licence GPL. Toutefois, Scratch et d'autres éléments sont des marques déposées (trademarks), dont la re-distribution est soumise à contraintes ^[4]. Une version ne contenant pas les marques déposées est mise à disposition et permet de distribuer des modifications éventuelles avec une licence GPL. Quelle que soit la version libre ou non, Scratch nécessite des logiciels non libres pour fonctionner, tel Adobe Flash Player.

Smalltalk est un langage de programmation orienté objet, réflexif et dynamiquement typé. Il fut l'un des premiers langages de programmation à disposer d'un environnement de développement intégré complètement graphique. Il a été créé en 1972.

Les principaux concepts de Smalltalk sont :

« Tout est objet. » Les chaînes de caractères, les entiers, les booléens, les définitions de classes, les blocs de code, les piles et la mémoire sont représentés en tant qu'objets ;

Tout est modifiable. Le langage permet par exemple de changer d'IDE en cours d'utilisation, sans recompiler ni redémarrer l'application. De même, il est possible de créer de nouvelles instructions de contrôle dans le langage. Certaines implémentations permettent de changer la syntaxe du langage, ou la façon dont le ramasse-miettes fonctionne ;

Le typage est dynamique, donnant ainsi une certaine concision au langage ;

Un ramasse-miettes mémoire est intégré et transparent pour le développeur ;

Un système de gestion d'exceptions avec reprise est fourni ;

Les programmes Smalltalk sont généralement compilés en bytecode, exécutés par une machine virtuelle ;

La traduction dynamique : les machines virtuelles commerciales modernes compilent le bytecode vers le code machine natif de façon à obtenir de meilleures performances, une technique dont Smalltalk-80 a été le pionnier, développé par ParcPlace Systems au milieu des années 1980. Cette idée a été adoptée par le langage de programmation Java quelque dix ans après et renommée « compilation *just-in-time* », ou JIT ;

Une classe peut hériter d'une seule autre classe (héritage simple).

Les différents types de langage

L'importance du langage se décline en cinq genres principaux :

- les "*petits*" langages de script (Awk, Rexx, Perl, Tcl),
- les *langages d'interrogation* (Sql, Dbase, Sasl),
- les *langages d'interfaces* graphiques (Tk) ou d'interface Web (Javascript, Php),
- les "*grands*" langages quasi-universels ou ciblés comme (C, C++, Pascal, Ada, Java, Smalltalk et Lisp, Prolog, Maple, Mathematica),
- les langages "*exotiques*" dont le but n'est pas toujours clair.

Dans un langage de programmation, on peut soit indiquer **comment faire** avec un langage impératif, soit indiquer **ce qu'on veut faire** avec un langage déclaratif (spécificatif ?), le mécanisme de gestion du langage gérant alors comment réaliser ce qu'on veut faire.

Les langages **impératifs** peuvent être divisés en trois catégories :

les langages *peu structurés* comme Basic, Fortran,

les langages *structurés par blocs* comme Pascal, C,

les langages conçus comme *langages objets* comme Smalltalk, C++, Java, Ada.

Les langages **déclaratifs** ont des natures différentes : on distingue

les langages *fonctionnels* comme Lisp,

les langages *logiques* comme Prolog,

les langages orientés *bases de données* comme Dbase, Sql, Sas.

Pour les langages impératifs, l'**exécution** d'un programme est *linéaire*, c'est à dire qu'elle s'effectue d'une instruction à la suivante, sachant que l'instruction suivante n'est pas forcément la ligne consécutive dans le fichier à cause des mécanismes de "contrôle du flot" que sont les tests, les boucles et les appels de sous-programmes.

Pour un langage déclaratif, une fois le but à atteindre ou le calcul à effectuer, l'exécution du langage suit un mécanisme d'exploration, de résolution ou d'interrogation qui est en général propre au langage.

Certains langages sont **typés explicitement** ce qui signifie que les variables doivent être déclarées selon des moules prédéfinis nommés types, comme par exemple byte, integer, longint pour une variable entière. Le typage impose des restrictions car il contraint le programmeur à déclarer beaucoup de choses. En retour la machine peut vérifier que tout se passe comme prévu et que rien ne "déborde".

Un langage peu typé ou typé non explicitement permet au programmeur de passer " du coq à l'âne " sans que cela pose problème à condition qu'il connaisse bien la zoologie !

Traditionnellement, les langages de script (Awk, Rexx, Perl, Tcl/ Tk) ainsi qu'Apl sont non typés explicitement. Par contre les langages compilés (C, C++, Pascal, Ada, Java, Fortran, Cobol) sont pratiquement toujours [fortement] typés explicitement.

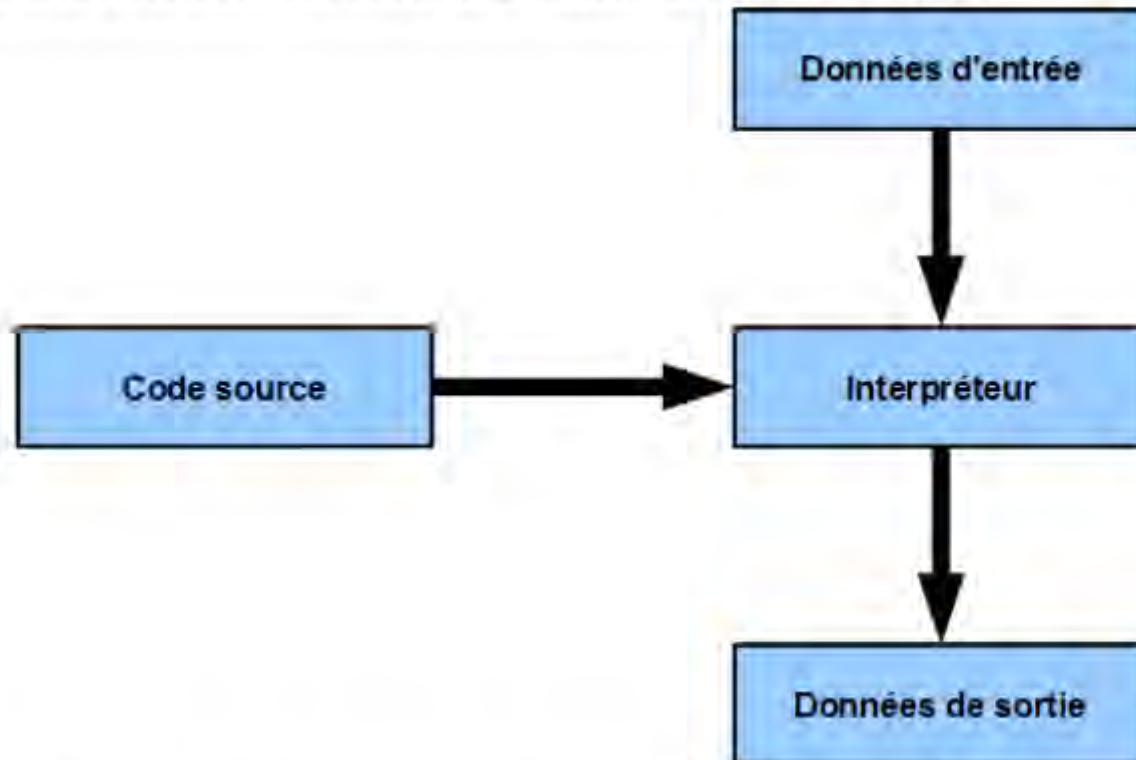
L'apport de la **méthodologie objet** a contraint certains anciens langages à modifier la syntaxe de base pour conserver des adeptes. Ainsi Pascal, langage non objet de conception est devenu Pascal Objet puis Delphi ; de même Lisp non objet à ses débuts a vu une variante nommée Clos lui conférer le statut de langage à objets. On nomme souvent ces langages "*orientés objets*" pour les distinguer des langages vraiment objets ou encore "*langages objets purs*" dont Smalltalk est le plus pur, suivi par C++ et Java.

[petite] histoire des Langages de Programmation
(gH) gilles.hunault@univ-angers.fr

Langages interprétés et langages compilés

Langages interprétés

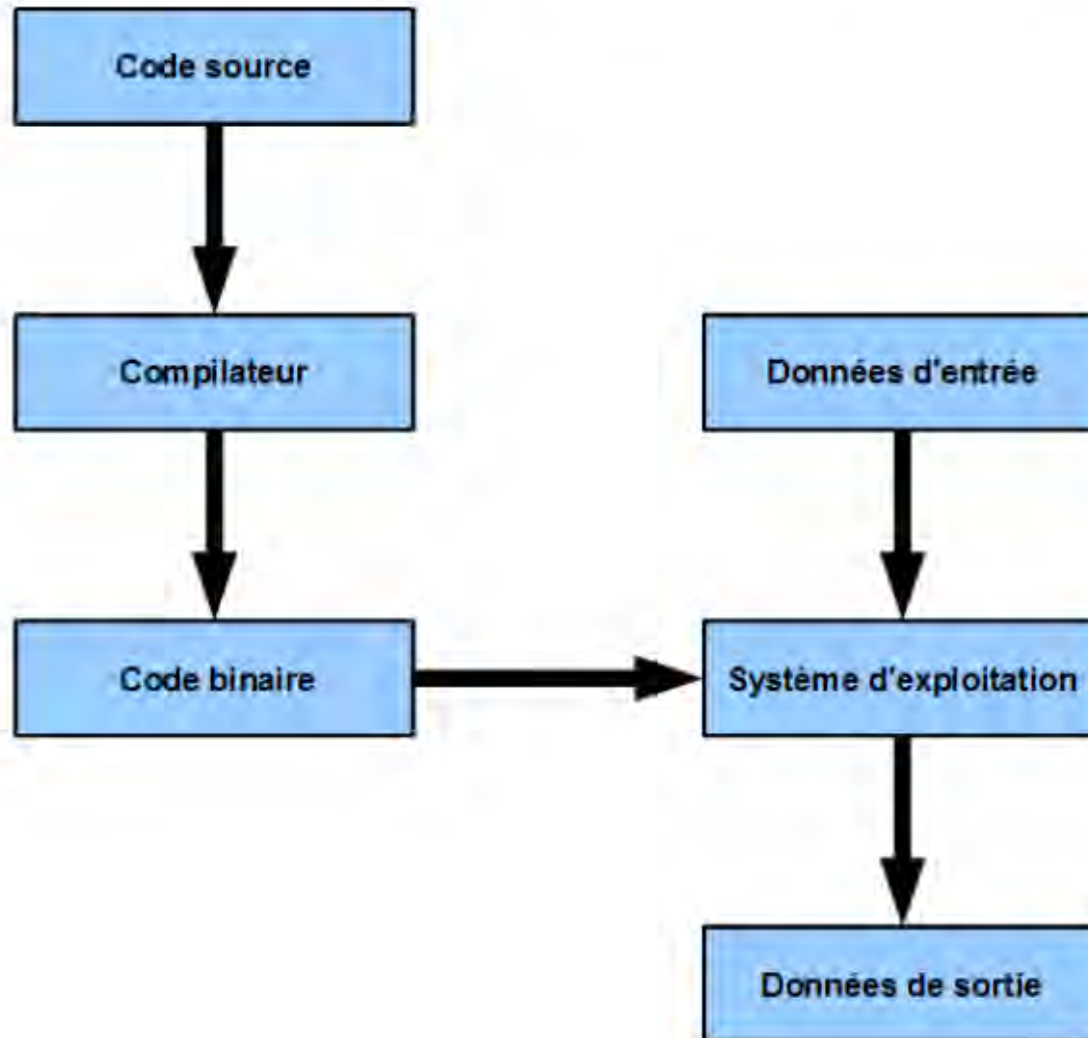
Dans ces langages, le code source (celui que vous écrivez) est interprété, par un logiciel qu'on appelle *interpréteur*. Celui-ci va utiliser le code source et les données d'entrée pour calculer les données de sortie :



L'interprétation du code source est un processus « pas à pas » : l'interpréteur va exécuter les lignes du code une par une, en décidant à chaque étape ce qu'il va faire ensuite.

Langages compilés

Dans ces langages, le code source (celui que vous écrivez) est tout d'abord compilé, par un logiciel qu'on appelle *compilateur*, en un *code binaire* qu'un humain ne peut pas lire mais qui est très facile à lire pour un ordinateur. C'est alors directement le système d'exploitation qui va utiliser le code binaire et les données d'entrée pour calculer les données de sortie :



Principales différences

- Dans un langage interprété, le même code source pourra marcher directement sur tout ordinateur. Avec un langage compilé, il faudra (en général) tout recompiler à chaque fois ce qui pose parfois des soucis.
- Dans un langage compilé, le programme est directement exécuté sur l'ordinateur, donc il sera en général plus rapide que le même programme dans un langage interprété.

L'association France-IOI a été créée en juin 2004 dans le but de développer la sélection et l'entraînement de l'équipe de France aux Olympiades internationales d'informatique (IOI). Les fondateurs de France-IOI s'occupaient jusqu'alors de cette activité au sein de l'association Prologin, en annexe de l'organisation du concours national d'informatique. À l'initiative d'[EPITA](#), le sponsor historique de l'association, France-IOI est alors créée pour prendre en charge cette activité. Dès sa création, l'association décide de faire profiter de son site d'entraînement à un large public, pour en faire un site d'apprentissage de la programmation et de l'algorithmique.

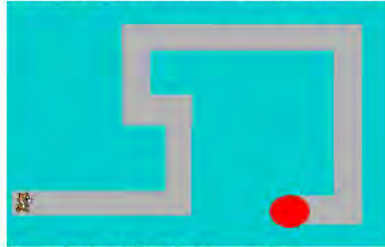
EXEMPLES DE FICHE

Labyrinthe

1. Introduction

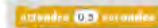
Lancer le logiciel Scratch qui est situé dans Logiciels/Maths/ et qui s'appelle scratch2-portable.
Attendre la mise en place des fichiers d'aide, annuler la mise à jour et le mettre en français en appuyant sur la mappemonde qui est en haut à gauche.
Lui faire ouvrir le fichier Labyrinthe2 qui est dans Commun/travail/Maths/Troisième/
L'enregistrer dans son dossier personnel.

2. **Première tâche** : Faire aller le chat jusqu'au rond rouge.



Pour cela, il va falloir choisir parmi les instructions de **Mouvement** (onglet Bleu) et les rajouter au centre du programme.

Après chaque déplacement, penser à rajouter



pour voir le déplacement.



3. **Deuxième tâche** :

Programmer pour que :

'appui sur la touche « f » fasse déplacer le chat vers la droite,

'appui sur la touche « d » fasse déplacer le chat vers la gauche,

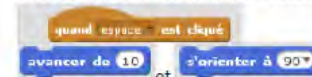
'appui sur la touche « r » fasse déplacer le chat vers le haut,

'appui sur la touche « c » fasse déplacer le chat vers le bas.

Garder le bloc ci-dessous comme point de départ



En se servant des instructions ci-dessous



et en finissant chaque « paquet » par



Bouton droit - Dupliquer

4. **Troisième tâche** :

Faire construire un nouveau labyrinthe en créant un nouvel arrière-plan.



D'abord Cliquer sur le crayon situé en bas à gauche puis sur **vectoriser** en bas à droite avant de commencer le dessin. Il vous suffit de tracer des rectangles et de les remplir de la même couleur par exemple.



Polygone régulier

Objectif :

Découvrir les instructions de déplacement de Scratch et les boucles. ici



Travail :

1. Tester et dire ce que fait le programme ci-contre

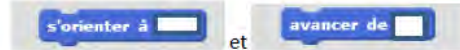


2. Transformer le à l'aide de



pour qu'il soit plus court et donne le même résultat puis enregistrer le sous un autre nom.

3. Obtenir le même résultat en se servant de



puis l'enregistrer sous un autre nom.

Quelle différence y a t'il entre



4. En s'inspirant de ce que vous venez de faire,

construire un triangle équilatéral.

5. En s'inspirant de ce que vous venez de faire,

construire un hexagone régulier.

6. En s'inspirant de ce que vous venez faire,

faire un programme qui demande le nombre de côtés du polygone régulier puis le faire construire.

Indication : Se servir de





Diviseur

Objectif :

Analyser et comprendre un programme puis s'en inspirer. Utiliser les listes de variables de Scratch.

Travail :

Voici le programme « PremierouPasBase » qui dit si un nombre est premier ou pas :

```

quand cliqué
  demander Donner un nombre et attendre
  mettre nombre à réponse
  mettre compteur à 2
  répéter jusqu'à nombre modulo compteur = 0
  ajouter à compteur 1
  dire regroupe compteur regroupe est un diviseur de nombre pendant 1 secondes
  si nombre = compteur alors
    dire Ce nombre est premier
  sinon
    dire Ce nombre n'est pas premier
  
```

1. Tester le
Noter les valeurs que vous avez essayé :
2. Expliquer comment il fonctionne en répondant à la question suivante

NB : `nombre modulo compteur` donne le reste de la division euclidienne de `nombre` par `compteur`

Noter les valeurs que prennent nombre et compteur si l'on a donné la valeur 15 au nombre.

Etape	<code>nombre modulo compteur</code>	Nombre	Compteur
1			
2			
3			

Puis reprendre le même travail si l'on avait choisi 35 comme valeur pour le nombre

Etape	<code>nombre modulo compteur</code>	Nombre	Compteur
1			
2			
3			
4			
5			

3. Pourquoi a-t-on mis `mettre compteur à 2` et pas `mettre compteur à 1` ?
.....
4. Pourquoi est-on sûr que le « répéter jusqu'à » va s'arrêter ?
.....
5. En s'inspirant du programme précédent, construire un programme qui demande un nombre et qui dit s'il est pair ou impair.
Sauvegarder le dans le répertoire de la classe/Travail/Maths en mettant votre nom dans le titre du fichier.
6. Construire un programme qui demande deux nombres (le plus grand d'abord) et dit si le deuxième est un diviseur du premier.
Sauvegarder le dans le répertoire de la classe/Travail/Maths en mettant votre nom dans le titre du fichier.
7. En se servant du programme « PremierouPasBase », le transformer pour obtenir tous les diviseurs d'un nombre.

Indications : il faudra créer une liste `Créer une liste` que l'on appellera `diviseur`,

On aura alors les commandes ci-contre :

Et il faudra modifier le répéter.

```

supprimer l'élément tout de la liste diviseur
ajouter thing à diviseur
  
```

8. Demander deux nombres et trouver leur plus grand diviseur commun (PGCD) et leur plus petit multiple commun (PPCM).