éduSCOL
Informer et accompagner les professionnels de l'éducation

Liberté • Égalité • Fraternité
RÉPUBLIQUE FRANÇAISE

MINISTÈRE DE L'ÉDUCATION NATIONALE ET DE LA JEUNESSE

VOIE GÉNÉRALE

2DE | 1RE | TLE

Enseignement scientifique

ENSEIGNEMENT COMMUN

THÈME 3
SOUS-THÈME 3-5 : MACHINES ET PROGRAMMES

Mots-clés

Ordinateur, machine universelle, mémoire, bit, octet, programme.

Références au programme

Jusqu'au début du XXe siècle, les machines traitant l'information sont limitées à une ou quelques tâches prédéterminées (tisser grâce à un ruban ou des cartes perforées, trier un jeu de carte perforées, séparer des cartes selon un critère, sommer des valeurs indiquées sur ces cartes...). Turing a été le premier à proposer le concept de machine universelle qui a été matérialisé 10 ans plus tard avec les premiers ordinateurs. Ceux-ci sont constitués à *minima* d'un processeur et d'une mémoire vive.

Savoirs

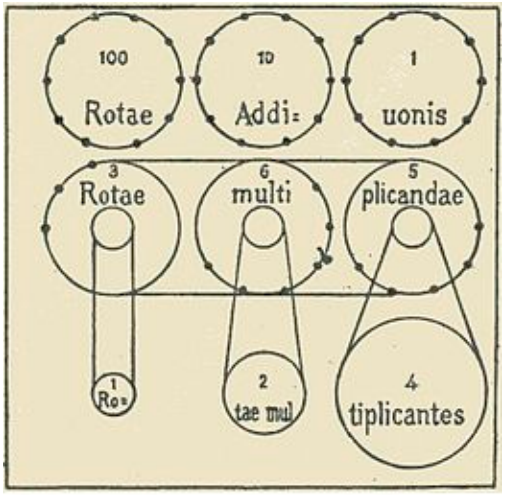
Un ordinateur peut manipuler des données de natures diverses une fois qu'elles ont été numérisées : textes, images, sons. Les programmes sont également des données : ils peuvent être stockés, transportés, et traités par des ordinateurs. En particulier, un programme écrit dans un langage de programmation de haut niveau (Python, Scratch…) peut être traduit en instructions spécifiques à chaque type de processeur.
Un programme peut comporter jusqu'à plusieurs centaines de lignes de code, ce qui rend très probable la présence d'erreurs appelées bogues (ou bugs).

Savoir-faire

Savoir distinguer les fichiers exécutables des autres fichiers sous un système d'exploitation donné.
Connaître l'ordre de grandeur de la taille d'un fichier image, son, vidéo.
Savoir calculer la taille en octets d'une page de texte (en ASCII et non compressé).
Étant donné un programme très simple, proposer des jeux de données d'entrée permettant d'en tester toutes les lignes.
Corriger un algorithme ou un programme bogué simple.

Histoire, enjeux, débats

Le premier calculateur mécanique à avoir effectivement fonctionné a été inventé par le mathématicien Blaise Pascal en 1642 afin d'alléger les calculs de son père, surintendant de la Haute-Normandie.



Dans les années 1830, le mathématicien anglais Charles Babbage invente une machine programmable grâce à des cartes perforées. Ses travaux découlent notamment des automates programmables de Jacquard (métier à tisser) et de la machine à calculer de Leibniz. Lady Ada Lovelace (1815-1852) publie le premier algorithme destiné à être programmé sur la machine analytique de Babbage.

À la fin du XIXe siècle, Herman Hollerith, inventeur de la mécanographie, invente un système de perforatrices, de trieuses et d'additionneuses à cartes, qui sera utilisée pour dépouiller rapidement les données du recensement des États-Unis de 1890. Les machines à cartes perforées se perfectionnent progressivement ; capables au début seulement d'additionner, elles peuvent bientôt soustraire puis multiplier et diviser ; ainsi la mécanographie sera utilisée à partir de la Seconde Guerre mondiale pour le calcul scientifique (tables de tir et conception de la bombe atomique).

Jusqu'au début du XXe siècle, les machines traitant l'information sont dédiées à une ou quelques tâches prédéterminées. Le concept de machine universelle apparaît dans les travaux d'Alan Turing en 1936.

Alan Turing est un mathématicien anglais connu en particulier pour deux faits marquants. D'une part, il a contribué à implémenter sur machine une méthode due à trois mathématiciens Polonais (Marian Rejewski, Jerzy Rózicki et Henryk Zygalski). Cette implémentation a permis de déchiffrer les messages allemands durant la seconde guerre mondiale. D'autre part, Turing est l'inventeur du concept de machine universelle qui a ouvert la voie à une part considérable de l'informatique théorique moderne.

En 1945, John Von Neumann, qui travaillait avec Eckert et Mauchly sur le projet du calculateur ENIAC, formule le principe de l'ordinateur à programme enregistré, dans lequel un programme est codé dans la mémoire du calculateur, de la même manière que les données.

Organisation d'un ordinateur

L'organisation générale d'un ordinateur, connue sous le nom *d'architecture de Von Neumann*, est composée d'un processeur et d'une mémoire. Cette structure n'a pas varié depuis les années 1940. Un programme décrit des calculs à effectuer sur des données. Il est codé dans la mémoire au même titre que les autres données. Le chargement d'un programme entraîne la création d'un processus dans le système d'exploitation et l'exécution du programme.

Unités informatiques

La mémoire est composée de plusieurs milliards de dispositifs électroniques permettant de maintenir deux états qu'on peut interpréter par 1 et 0. Cette unité de mémoire à deux états (0 ou 1) est appelée un *bit*.

Ces dispositifs électroniques sont organisés en agrégats de huit, seize, trente-deux, soixante-quatre bits, et parfois davantage, que l'on appelle des *cases mémoires*. Ces cases permettent de mémoriser les données et les programmes sous la forme de mots de huit, seize, trente-deux, soixante-quatre bits, et parfois davantage. Le nombre de ces cases définit la taille mémoire de l'ordinateur.

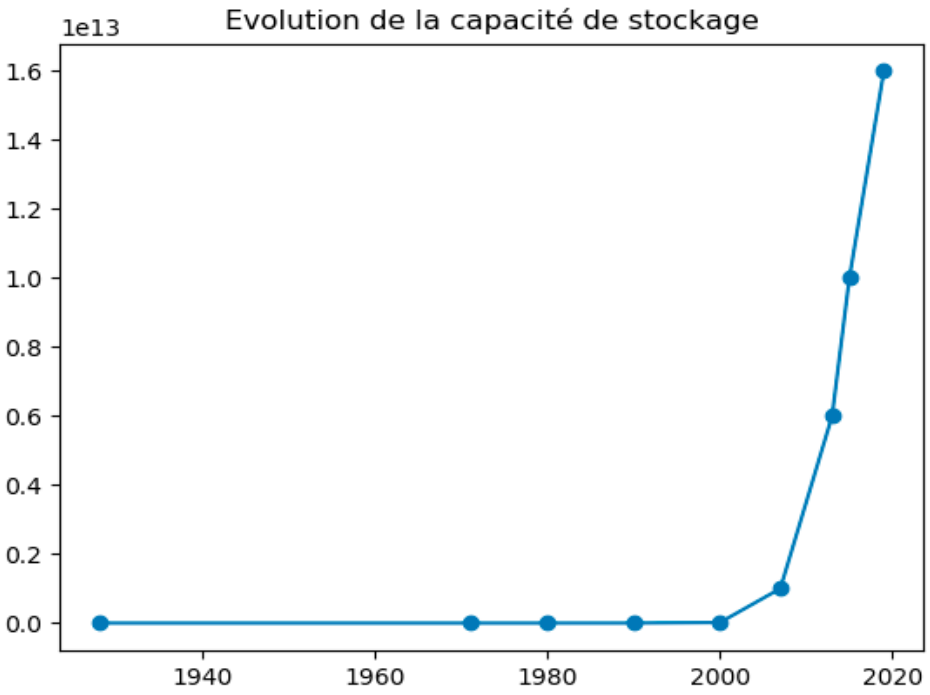
Une autre unité utilisée est l'octet. Un octet est un mot de 8 bits. Chaque bit pouvant prendre deux valeurs (0 et 1), un octet permet de coder $2^8 = 256$ valeurs différentes.

Les élèves ont déjà étudié, notamment dans l'enseignement SNT de seconde, les différentes unités utilisées en informatique (octet, kilooctet, mégaoctet, gigaoctet, téraoctet, pétaoctet), dont les expressions à l'aide des puissances de 10 pourront être rappelées.

Le tableau ci-dessous indique différents supports de stockage, leur année d'invention et les capacités de stockage associées.

Année d'invention	Support	Capacité de stockage
1928	Bande magnétique	50 octets
1971	Disquette	360 Ko
1980	Disquette	1,2 Mo
1990	CD-Rom	700 Mo
2000	Mémoire Flash	16 Go
2007	Disque dur ou SSD	1 To
2013	Disque dur ou SSD	6 To
2015	Disque dur ou SSD	10 To
2019	Disque dur ou SSD	16 To

La courbe ci-dessous illustre la rapidité de l'évolution de la capacité de stockage.



Les textes, les images, les sons, les vidéos, les programmes exécutables sont tous stockés en mémoire sous forme de séries de bits. Les codages relatifs aux différentes natures de fichiers sont appelés *formats*. Une extension est une petite chaîne de caractères (préfixée par un point) ajoutée en fin de nom du fichier afin que l'ordinateur puisse interpréter correctement ce fichier et que l'utilisateur puisse en reconnaître le format. Par exemple, un fichier nommé « maVideo.avi » a pour nom « maVideo », son extension est « avi », ce qui correspond à un format vidéo. Dans certains formats, les fichiers sont rangés de manière intelligente et les informations redondantes ne sont pas intégralement codées à chaque occurrence (par exemple un mot qui apparaît plusieurs fois dans un fichier texte). De même, un fichier audio au format mp3 ne code pas les sons que l'oreille humaine ne peut pas entendre. Les tableaux ci-dessous contiennent une liste non exhaustive de formats et, pour chacun, un ordre de grandeur de la taille d'un fichier de ce format, sachant que cette taille dépend en réalité de nombreux facteurs.

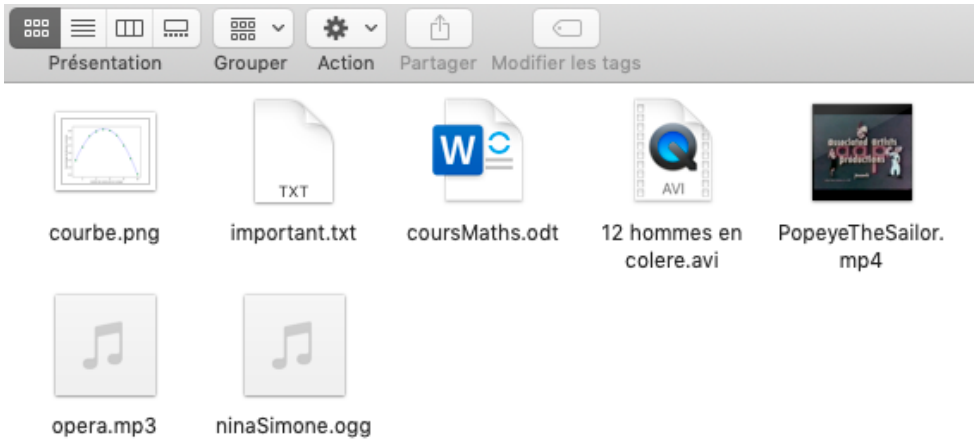
Format (donné par l'extension)	Nature
txt	Texte
doc	Texte
odt	Texte
jpeg	Image
png	Image
avi	Vidéo
mp4	Vidéo
mp3	Audio
ogg	Audio

Type de Fichier	Ordre de grandeur	Taille habituelle
Fichier texte	ko	< 20 ko
Fichier image	Mo	entre 1 et 20 Mo
Fichier audio	Mo	entre 3 et 30 Mo
Fichier Vidéo	Go	entre 100 Mo et 5Go

Propositions d'activités

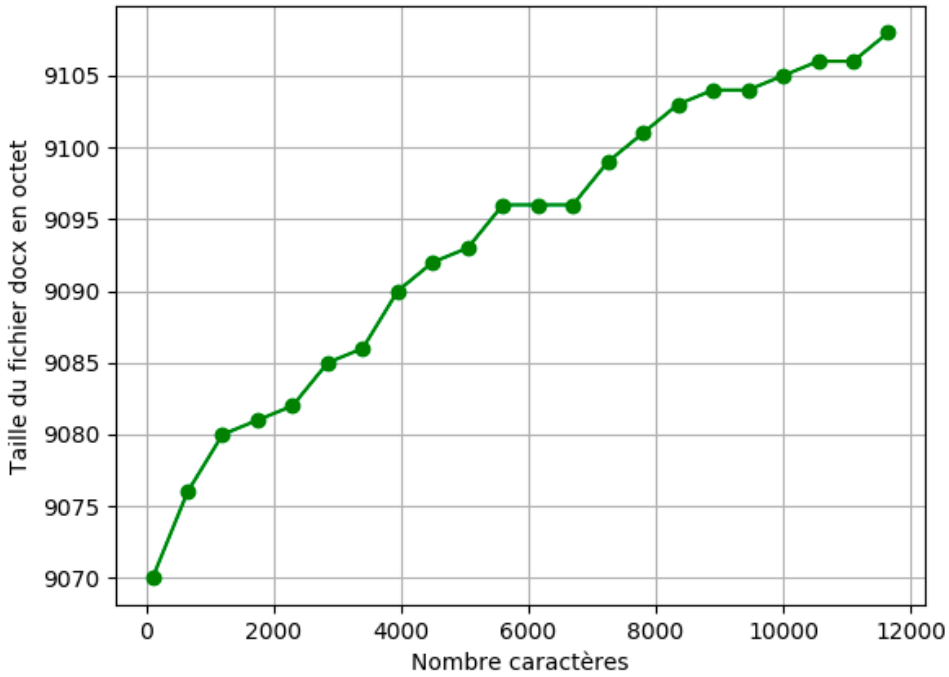
Activité 1

1. L'image suivante est une capture d'écran d'un logiciel de gestion de fichiers.



Déterminer la nature de chaque fichier.

2. Un fichier texte (brut, c'est-à-dire au format txt) nommé data.txt comporte 1 800 caractères. Dans ce format, chaque caractère occupe 1 octet de mémoire.
   a. Combien de bits sont utilisés en mémoire pour stocker ce fichier ?
   b. La courbe suivante représente l'évolution de la taille d'un fichier au **format docx** en fonction du nombre de ses caractères. Combien pèse le fichier précédent lorsqu'après l'avoir ouvert avec le logiciel Word, on l'enregistre au format docx ?



Activité 2 : analyse d'un programme considéré comme le premier programme de l'histoire

Ada Lovelace (1815-1852) est la fille du poète anglais Lord Byron. En 1833, elle rencontre Charles Babbage, qui est notamment l'inventeur d'une machine à calculer programmable connue sous le nom de machine analytique. Ada Lovelace traduit en anglais un article écrit en français sur la machine analytique de Babbage. Elle complète cet article de commentaires qui formalisent les idées de Babbage. L'algorithme des nombres de Bernoulli figure dans la note G. Il est considéré comme le premier programme informatique au monde, car écrit avec un formalisme destiné à une exécution par une machine. On ne manquera pas de faire remarquer qu'il a été écrit par une femme.

En 1977, le département de la Défense des États Unis dénomme un langage de programmation *Ada* (inspiré du langage Pascal) en l'honneur d'Ada Lovelace. Ce langage, très structuré, est utilisé pour des projets nécessitant un très haut niveau de fiabilité (domaines militaire, aéronautique, etc).

Le document ci-dessous décrit le programme de calcul des nombres de Bernoulli par Ada Lovelace.

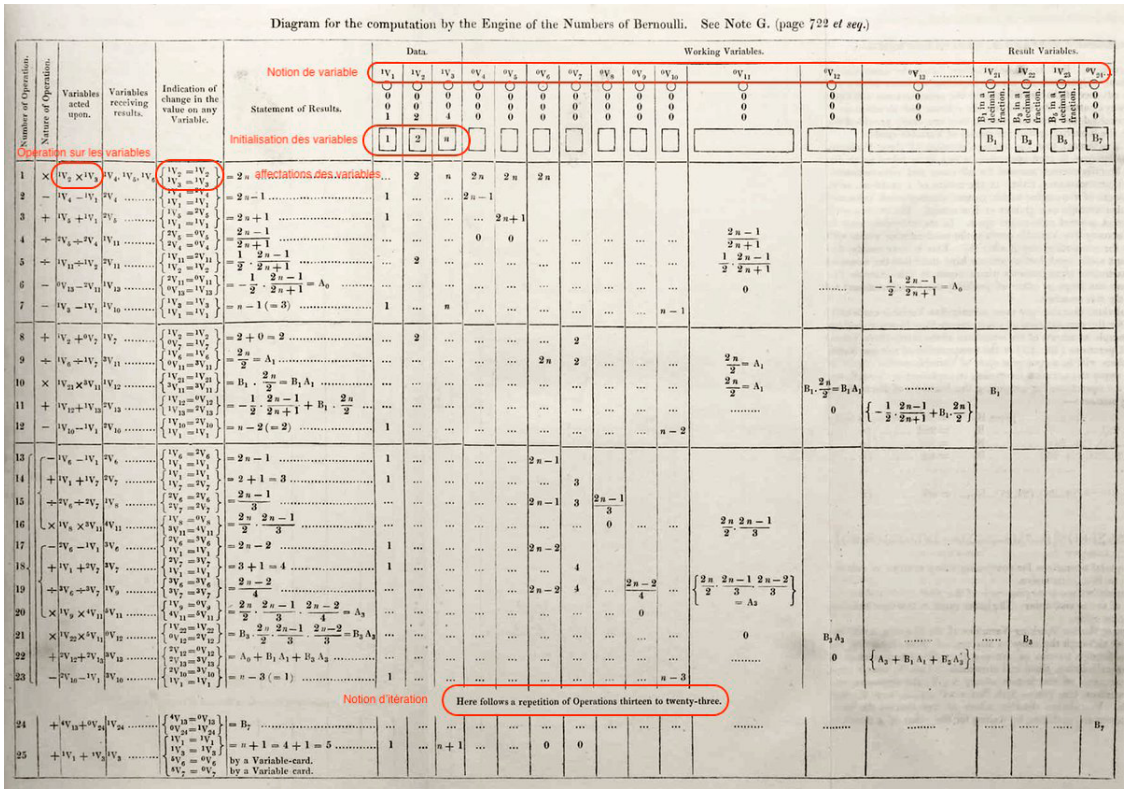


Diagram for the computation by the Engine of the Numbers of Bernoulli. See Note G. (page 722 *et seq.*)

Notion de variable

Initialisation des variables

Opération sur les variables

affectations des variables...

Notion d'itération

Here follows a repetition of Operations thirteen to twenty-three.

Il peut être demandé aux élèves d'analyser le contenu de ce programme.

Retrouvez éduscol sur

Cette analyse permet de comprendre la structure du programme et éventuellement de la comparer à celle d'un programme écrit en Python pour calculer les nombres de Bernoulli (voir leur définition ci-dessous).

- La première colonne du tableau contient les numéros des lignes pour pouvoir y faire référence. Cette fonctionnalité est présente aujourd'hui dans tous les éditeurs de code informatique.
- La seconde colonne fait référence à l'opération utilisée (addition, soustraction, multiplication, division). Cette information est importante pour la machine qui doit exécuter ces différentes opérations. Dans les ordinateurs modernes, ces opérations sont effectuées à l'aide des composants électroniques internes au processeur.
- La troisième colonne précise à quoi s'appliquent les opérateurs. Dans la première ligne, il est indiqué que la machine doit multiplier une fois le contenu de la variable V_2 par une fois le contenu la variable V_3, ce qui peut s'écrire $V_2 \times V_3$.
- La quatrième colonne identifie les variables qui vont être affectées par l'opération. Dans la première ligne, $1V_4$, $1V_5$, $1V_6$ signifie que la nouvelle valeur de la variable V_4 sera une fois le résultat de l'opération. Même chose pour V_5 et V_6. Pour réaliser l'opération, on doit modifier le contenu de certaines variables. Par exemple, en ligne 4, on veut réaliser $2V_5 \div 2V_4$. Le résultat obtenu est alors placé dans la variable V_{11}.
- La cinquième colonne indique ce qu'il advient des variables qui ont été utilisées pour le calcul. Pour calculer $2V_5 \div 2V_4$, on a utilisé les contenus des variables V_5 et V_4. Après le calcul de $2V_5 \div 2V_4$, on met à 0 le contenu de la variable V_5 (c'est ce qu'indique l'instruction $2V_5 = 0V_5$) et on met à 0 le contenu de la variable V_4 (c'est ce qu'indique l'instruction $2V_4 = 0V_4$).
- La sixième colonne donne le résultat de l'opération. Par exemple, ligne 1, l'opération est $V_2 \times V_3$, à ce moment-là $V_2 = 2$ et $V_3 = n$ donc le résultat est $2n$.
- La septième colonne donne le contenu des variables. Par exemple, au début du programme, les variables V_1, V_2 et V_3 sont telles que $V_1 = 1$, $V_2 = 2$ et $V_3 = n$.
- Entre la ligne 23 et 24, Ada Lovelace note qu'il faut répéter les lignes 13 à 23, ce qui correspondrait aujourd'hui à un bloc d'instructions placé dans une boucle for.

On constate à la fois que les notions de variable aléatoire et de boucle y sont déjà présentes.

Complément : les nombres de Bernoulli

Les nombres de Bernoulli peuvent être calculés de proche en proche de la manière suivante :

$B_1 = 1$

Pour déterminer les autres termes, on construit le triangle de Pascal

1 1
1 2 1
1 3 3 1
1 4 6 4 1
…

et on écrit des équations associées au début des lignes du triangle (on ne tient pas compte des chiffres 1 les plus à droite) de la manière suivante :

Deuxième ligne : $1 + 2B_1 = 0$
Troisième ligne : $1 + 3B_1 + 3B_2 = 0$
Quatrième ligne : $1 + 4B_1 + 6B_2 + 4B_3 = 0$
…

La résolution de ces équations permet de déterminer de proche en proche les nombres de Bernoulli.

Activité 3 : test et correction de programmes

Exemple 1

Un site internet propose des impressions de photos. Le prix dépend du nombre de photos commandées (tarif dégressif). Le détail de ces tarifs est donné dans le tableau ci-dessous.

Quantité	Prix par tirage
Entre 1 et 19	0,16€
Entre 20 et 499	0,14€
Entre 500 et 999	0,12€
Entre 1000 et 1499	0,10€
Plus de 1500	0,08€

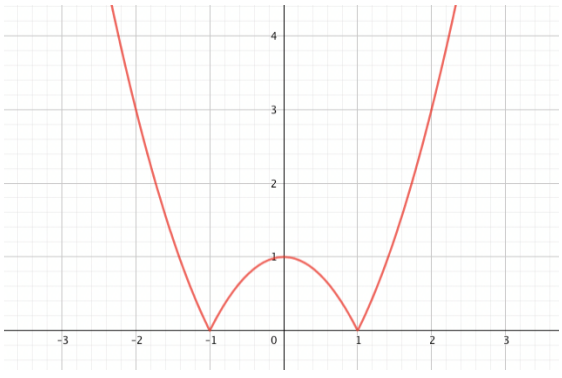
1. Écrire une fonction (en langage Python) qui, pour un nombre de photos donné, calcule le prix à payer.
2. Proposer un jeu de données permettant de tester le programme en balayant les différents cas.

Exemple 2

On considère la fonction informatique dont le code est le suivant :

```
In [7]: def f(x):
            if x<=-1:
                return x*x-1
            elif -1<=x and x<=1:
                return 1-x*x
            else:
                return x*x-1
```

La fonction ainsi définie a pour représentation sur [-3,3] la courbe suivante :



La courbe de la fonction f est constituée de trois portions qui correspondent aux 3 conditions « x inférieur à -1 », « x compris entre -1 et 1 », « x supérieur à 1 ».

1. Pour tester le programme, on peut proposer le jeu de données suivant.

Pour $x = -2$, la première condition est vérifiée, la fonction f renvoie la valeur 3.

Pour $x = 0$, la première condition n'étant pas vérifiée, on entre dans la deuxième condition qui est vérifiée car $-1 \leq 0$ et $0 \leq 1$. La fonction f renvoie alors la valeur 1.

Pour $x = 3$, la première et la deuxième condition n'étant pas vérifiées, on entre dans la troisième condition. La fonction f renvoie alors la valeur 8.

La structure if, elif, else est exécutée dans cet ordre. Cela signifie que, si une condition est réalisée, le programme effectue l'instruction qui la suit et n'examine pas les conditions suivantes, qu'elles soient ou non réalisées. Par exemple, pour $x = -1$, la condition qui suit l'instruction if et celle qui suit l'instruction elif sont toutes deux vérifiées. Cependant, le programme ne considère que la première condition.

2. Modifier le programme afin qu'aucun paramètre x ne puisse vérifier plusieursconditions.

3. Corriger un programme

Étant donné un nombre réel a compris entre 0 et 8, on cherche à déterminer à 0,1 près le plus grand nombre réel x appartement à $[\text{xmin}, \text{xmax}]$ et vérifiant $f(x) \leq a$. On propose plusieurs programmes qu'il s'agit de tester et de corriger.

a. Premier programme

```
In [5]: def solution1(a,xmin,xmax):
            x=xmin
            while f(x)<=a:
                x=x+0.1
            return x
```

Tester le programme pour $a = 0{,}5$; $\text{xmin} = -1$; $\text{xmax} = 0$.

Que constate-t-on ?

Corriger l'erreur.

L'erreur vient du fait que le programme ajoute une fois de trop 0,1 à x puisque la boucle s'arrête lorsque la condition n'est plus vérifiée.

b. Pour corriger ce problème, on modifier le programme de la manière suivante :

```
In [3]: def solution2(a,xmin,xmax):
            x=xmin
            while f(x)<=a:
                x=x+0.1
            return x-0.1
```

Tester le programme pour $a = 0{,}5$; $\text{xmin} = -1$, $\text{xmax} = 0$
puis $a = 0{,}5$; $\text{xmin} = -1$; $\text{xmax} = 1$,
enfin pour $a = 0{,}5$; $\text{xmin} = -1$; $\text{xmax} = 3$.

Que constate-t-on ? Corriger l'erreur.

c. Que penser du programme suivant :

```
In [4]: def solution3(a,xmin,xmax):
            x=xmax
            while f(x)>a:
                x=x-0.1
            return x
```

Bibliographie et sitographie

- La revue de culture scientifique en ligne Interstices : L'invention de la mécanographie
- Le site de l'INRIA propose plusieurs articles, dont un sur la machine universelle de Turing.