

DES PUISSANCES DE 2 AU CODE ANDROMEDE

--> préliminaire : Les puissances de 2

Ecrire les 11 premières puissances de 2.

Il faut connaître par coeur les 11 premières puissance de 2
 $1024 = ?$

exercice python :

- écrire un programme qui affiche les puissances de 2 de 2^0 à 2^{20}
- écrire un programme qui, connaissant une puissance de 2, renvoie la puissance (pour $m=2^n$, l'entrée c'est m, la sortie c'est n)

correction partielle code python :

```
>>> for i in range(0,21):  
...     print i,2**i
```

```
...  
0 1  
1 2  
2 4  
3 8  
4 16  
5 32  
6 64  
7 128  
8 256  
9 512  
10 1024  
11 2048  
12 4096  
13 8192  
14 16384  
15 32768  
16 65536  
17 131072  
18 262144  
19 524288  
20 1048576
```

--> système binaire

Dans notre système pour compter, dit décimal, on a :

$$1243 = 1000 + 200 + 40 + 3$$

soit :

$$1243 = 10^3 + 2*10^2 + 4*10^1 + 3*10^0$$

Nous sommes en base 10. On utilise alors les 10 symboles 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 pour noter les nombres.

Dans le système binaire, en base 2, nous aurons 2 symboles 0 et 1 pour noter les nombres.

Ainsi, tout nombre en binaire pourra s'écrire :

$$a = \text{somme } a_i * 2^i \text{ où les } a_i \text{ appartiennent à l'ensemble } \{0,1\}$$

Par exemple

$$1001 = 1*2^3 + 0*2^2 + 0*2^1 + 1*2^0$$

$$1001 = 9$$

vérification python :

```
>>> bin(1001)  
>>> '9'
```

décimal binaire

0	0
1	1
2	10
3	11
4	100
5	101

exercice python :

- écrire un programme qui affiche en binaire les entiers naturels de 1 à 20

décimal vers binaire

on effectue les divisions entières successives par 2 du nombre à transformer.

```
127 = 63*2 + 1
63 = 31*2 + 1
31 = 15*2 + 1
15 = 7*2 + 1
7 = 3*2 + 1
3 = 1*2 + 1
```

```
127 = (((((1*2 + 1)*2 + 1)*2 + 1)*2 + 1)*2 + 1)*2 + 1
127 = 1*2^6 + 1*2^5 + 1*2^4 + 1*2^3 + 1*2^2 + 1*2^1 + 1*2^0
```

D'où 127 s'écrit 1111111 en binaire.

vérification python :

```
>>> bin(1111111)
>>> '127'
```

exercice python :

- écrire un programme qui convertit en binaire un entier naturel.

--> un mot sur l'hexadécimal

Le système hexadécimal correspond à la base 16. On le rencontre entre autres dans le codage des couleurs.

On utilise les 16 symboles suivant pour représenter les nombres :

{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F}

A vaut 10, B vaut 11, etc..

Ainsi FF = $15*16^1 + 15*16^0 = 255$

Pour coder 127 en hexadécimal, il faut effectuer les divisions successives par 16 et procéder comme pour le binaire.

On peut s'aider de python pour le calcul

```
>>> 127/16
>>> 7
>>> 127%16
>>> 15
```

Donc $127 = 7*16 + 15$

$127 = 7*16^1 + 15*16^0$

127 s'écrit donc 7F (parce que 15 c'est le symbole F)

exercice python :

- écrire un programme qui convertit en hexadécimal un entier naturel, et vice-versa.

--> Codage d'un nombre

un bit c'est 0 ou 1

un octet c'est 8 bits

Le plus grand nombre entier naturel que l'on peut coder sur 8 bits est donc :

11111111

code python :

```
>>> bin(11111111)
>>> '255'
```

Quel est le plus grand entier que l'on peut coder sur 16 bits ?

En remarquant que :

```
>>> bin(65535)
'0b1111111111111111'
>>>
```

il faudra donc 16 bits pour coder 65535.

--> Codage d'un texte

On utilise le code ASCII (wikipedia) qui donne en hexadécimal le code ASCII des caractères du clavier.
Table fournie : [ASCII_Code_Chart.svg](#)

Exemple : A = 41 en hexadécimal = 65 en décimal

Pour coder le caractère A, on va donc coder en binaire le nombre 65 : 1000001

Il suffit de 7 bits pour coder les caractères de l'alphabet.

Décoder en français la suite binaire suivante pour connaître le message envoyé du futur dans le film :
The.Andromeda.Strain. (2008 American science-fiction film)
basée sur une nouvelle de Michael Crichton (1969) (http://en.wikipedia.org/wiki/The_Andromeda_Strain).

```
0110111 0110011 0111001 0110101 0110010 0111000
1000010 1000011 1000001 1001001 1001100 1001100
1010101 1010011 1001001 1001110 1000110 1000101
1010010 1001110 1010101 1010011
```

Voir l'extrait de film :

Python

- De quoi est fait un programme :
 - une suite d'instruct^o manipulent des données structurées
 - définit^o fonct^o
 - déclare^t diverses
- Une instruct^o :
 - une affectat^o
 - une boucle
 - une condit^o
 - une entrée/sortie
- Le langage est simple.
- syntaxe → il faut être très précises donc il faut bien analyser.
 - découper un gros probl^m en petits probl^m
 - trouver les bonnes structures de données pe modéliser les données du probl^m
 - savoir ce qu'on programme.

I - des Variables

int, float, long → ce sont les types numériques
bool → le bool et nous dit si c'est vrai ou faux "booléen"
str, list → ce sont les types de séquences

- type numériques : <, >, <=, >=, ==, !=
opérateur de comparaison différent addit^o multiplicat^o divis^o quotient modulo
+ * / // %
- type booléen : les booléens ne prennent que deux valeurs.
" les 2 écritures = true, false and or not
et logique ou logique non logique
- séquences : + concaténat^o, * concaténat^o multiple, [] opérateurs d'accès
str non mutable, list mutable

Exemple terminal pour type numérique :

```
a = 456
type(a)
type 'int' >

a = 456.0
type(a)
type 'float' >
```

type booléen :

```
a = False
type(a)
type 'bool' >
```

type séquence :

```
a = 'bonjour'
2^3
1
2x3
8
45128
0,57
'a
'bonjour'
type(a)
type 'str'
```

a = a + ' toto'
a
'bonjour toto'
a[0]
b
a[3:7]
'jour'

```
maListe = [11, 'azerty', 78, 0, [1, 2, 3], 'zaza']
'maListe'
maListe[0]
11
maListe[3]
'0'
```

II affectation

Instruction

- Variable est manipulée par son nom, un identificateur x .

• $x = 0$

maListe = [1, 6, 10, -3]

$x = x + 1$ affectation particulière appelée incrément, raccourci $x += 1$

nom = raw_input("Donnez votre nom : ")

est_bissextile = bissextile(demandeUneAnnée())

monAnni ^{Affectat°} $\text{moisNaissance} == \text{moisCourant}$ and $\text{jourNaissance} == \text{jourCourant}$

III. boucle for

si on veut taper à la main il faut faire l'espace.

for variable in sequence: instruction 1
instruction n

séquence est une liste.

```
mot = input("Donnez un mot : ")
for e in mot:
    if e in "aeiou":
        print "voyelle"
    else:
        print "consonne"
```

```
print i, i*i
print range(10)
```

```
mot = "agenthuige"
for i in range(len(mot)):
    print (mot[i])
```

Quand on ne sait pas comment exécuter le bloc d'instruction; on utilise "while" condition:

pour arrêter \rightarrow Keyboard Interrupt.

pour être infini \rightarrow true.

```
while x <= 0:
```

```
    x = input("un x > 0 s'il vous plaît")
```

input = entier strictement positif.

III condit°.

```
if bloc d'instruction si true
else " " " false
```

```
print ("Mon résultat est : " + mot)
```

```
n = input("Donnez un entier : ")
```


Des puissances.

ISN

• 11 premières puissances de 2

- $2^0 = 1$
- $2^1 = 2$
- $2^2 = 4$
- $2^3 = 8$
- $2^4 = 16$
- $2^5 = 32$
- $2^6 = 64$
- $2^7 = 128$
- $2^8 = 256$
- $2^9 = 512$
- $2^{10} = 1024$

$$* 10 = 1 \times 2^1 + 0 \times 2^0 = 2$$

$$* 1011 = 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0$$

$$= 8 + 0 + 2 + 1$$

$$= 11$$

• Écrire en binaire.

- $85 = 42 \times 2 + 1$
- $42 = 21 \times 2 + 0$
- $21 = 10 \times 2 + 1$
- $10 = 5 \times 2 + 0$
- $5 = 2 \times 2 + 1$
- $2 = 1 \times 2 + 0$
- $1 = 0 \times 2 + 1$

$$85 = 101\ 0101$$

	2^5	2^4	2^3	2^2	2^1	2^0
	1	0	1	0	1	0

$$101\ 0101 = 1 \times 2^6 + 1 \times 2^4 + 1 \times 2^2 + 1 \times 2^0$$

$$= 64 + 16 + 4 + 1$$

$$= 85$$

$$85 = (21 \times 2) \times 2 + 1$$

$$= ((10 \times 2 + 1) \times 2) \times 2 + 1$$

$$= \dots$$

• Hexadécimal

$$AF = 10 \times 16^1 + 15 \times 16^0 = 175$$

$$* 240 / 16 = 15$$

$$240 \% 16 = 0$$

$$240 = 15 \times 16^1 + 0 \times 16^0$$

$$= F0$$

```
Terminal < python
>>> for i in range (0, 11):
    print i
file "<stdin>", line 2
print i
...
0
...
10
for i in range (0, 11):
    print 2**i
...
1
...
1024
>>> for i in range (0, 11):
    print i, 2**i
...
0 1
...
10 1024
```

* j'ai noté les erreurs

* faire afficher une chaîne: '2^' + str(i) + '=' + 'i'

print '2^' + str(i) + '=' + 'i'

↓ ↓ ↓ ↓

affiche opérateur fond de la ligne

* le système décimal → base (10³; 10²; ...)

$$* 127 / 16 = 7$$

$$127 \% 16 = 15$$

$$127 = 7 \times 16 + 15$$

$$= 7 \times 16^1 + 15 \times 16^0$$

↓ ↓

7 F

Codeage d'un nombre / d'un texte B A C I L U S

	A	Z	E	R	T	Y	INFERNUS
hexadécimal	41	5A	45	52	54	59	57
décimal	65	90	69	82	84	89	

binaires 1000001 1011010 1000101 1010010 1010100 1011001

<https://groups.google.com/Forum/Forum/isi-ts2-lar>

0110111

- 55 = 32 + 16 + 4 + 2 + 1
- 51 = 32 + 16 + 2 + 1
- 57 = 32 + 16 + 8 + 1
- 53 = 32 + 16 + 4 + 1
- 50 = 32 + 16 + 2
- 56 = 32 + 16 + 8
- 66 = 64 + 2
- 67 = 64 + 2 + 1
- 65 = 64 + 1
- 73 = 64 + 8 + 1
- 76 = 64 + 8 + 4
- 76 = 64 + 8 + 4
- 85 = 64 + 16 + 4 + 1
- 83 = 64 + 16 + 2 + 1
- 73 = 64 + 8 + 1
- 78 = 64 + 8 + 4 + 2
- 70 = 64 + 4 + 2
- 69 = 64 + 4 + 1
- 82 = 64 + 16 + 2
- 78 = 64 + 8 + 4 + 2
- 85 = 64 + 16 + 4 + 1
- 83 = 64 + 16 + 2 + 1

	2 ⁶	2 ⁵	2 ⁴	2 ³	2 ²	2 ¹	2 ⁰
55	0	1	1	0	1	1	1
51	0	1	1	0	0	1	1
57	0	1	1	1	0	0	1
53	0	1	1	0	1	0	1
50	0	1	1	0	0	1	0
56	0	1	1	1	0	0	0
66	1	0	0	0	0	1	0
67	1	0	0	0	0	1	1
65	1	0	0	0	0	0	1
73	1	0	0	1	0	0	1
76	1	0	0	1	1	0	0
76	1	0	0	1	1	0	0
85	1	0	1	0	1	0	1
83	1	0	1	0	0	1	1
73	1	0	0	1	0	0	1
78	1	0	0	1	1	1	0
70	1	0	0	0	1	1	0
69	1	0	0	0	1	0	1
82	1	0	1	0	0	1	0
78	1	0	0	1	1	1	0
85	1	0	1	0	1	0	1
83	1	0	1	0	0	1	1

- 55 = 3x16 + 7 55 → 3x16' + 7x16° = 37 7
- 51 = 3x16 + 3 51 → 3x16' + 3x16° = 33 3
- 57 = 5x16 + 9 57 → 3x16' + 9x16° = 39 9
- 53 = 3x16 + 5 53 → 3x16' + 5x16° = 35 5
- 50 = 3x16 + 2 50 → 3x16' + 2x16° = 32 2
- 56 = 3x16 + 8 56 → 3x16' + 8x16° = 38 8
- 66 = 4x16 + 2 66 → 4x16' + 2x16° = 42 B
- 67 = 4x16 + 3 67 → 4x16' + 3x16° = 43 C
- 65 = 4x16 + 1 65 → 4x16' + 1x16° = 41 A
- 73 = 4x16 + 9 73 → 4x16' + 9x16° = 49 I
- 76 = 4x16 + 12 76 → 4x16' + 12x16° = 48 K (L)
- 76 = 4x16 + 12 76 → 4x16' + 12x16° = 48 K (L)
- 85 = 5x16 + 5 85 → 5x16' + 5x16° = 55 U
- 83 = 5x16 + 3 83 → 5x16' + 3x16° = 53 S
- 73 = 4x16 + 9 73 → 4x16' + 9x16° = 49 I
- 78 = 4x16 + 14 78 → 4x16' + 14x16° = 48 M (N)
- 70 = 4x16 + 6 70 → 4x16' + 6x16° = 46 F
- 69 = 4x16 + 5 69 → 4x16' + 5x16° = 45 E
- 82 = 5x16 + 2 82 → 5x16' + 2x16° = 52 R
- 78 = 4x16 + 14 78 → 4x16' + 14x16° = 48 M (N)
- 85 = 5x16 + 5 85 → 5x16' + 5x16° = 55 U
- 83 = 5x16 + 3 83 → 5x16' + 3x16° = 53 S

- Alimentat°
- Ventilateur
- Radiat°

- graveur
- lecteur disque
 - DVD
 - disquette
 - CD-ROM

- bande
- virtuelle
- RAM
- dur

Ordinateur

— Unité centrale

— carte mère

— mémoire

— pile

— carte

— chipset

— C.P.U

— BUS

— Disque

— SCSI
— IDE

- UC
- UAL
- autres cartes
- réseau
- graphique

carte mère. BIOS ROM
OS
File

Unité Cent. Mémoire morte
disque dur
DVD
Carte graphique
Son
addition. réseau

Conn.
entra./sortie

periph. —
Entrée
Sortie
ent./sortie

— Nappes

— clavier

— souris

— écran

— imprimante.

De quoi est composé un ordinateur ?

* partie principale → unité centrale → contient les éléments essentiels de l'ordinateur :

- avec unité centrale se branchent :

- le clavier & la souris → introduit des informations & exécute les ordres. (périphérique entrée)
- l'écran → visualise sous forme de texte, graphique des informations. (périphérique sortie)

- à l'intérieur de l'unité centrale il y a :

- x Alimentation
- x Ventilateur
- x 2^e disque dur
- x Carte réseau
- x Graveur
- x Lecteur DVD
- x " disquettes
- x Carte graphique
- x Disque dur
- x Nappe (relie D.Dur à carte mère)
- x Carte TV
- x Carte mère
- x Mémoire RAM.

* carte mère → constitue une pièce maîtresse de l'ordi & emplace au on peut ajouter d'autres cartes.

x carte graphique → gère les images... (jeux, images 3D, TV etc...)

x carte réseau → permet brancher ordi à un réseau → ligne téléphonique / TV.

* périphérique stockage → composant → gr^{de} quantité d'info peut être enregistrée sous forme de bits.

- disques durs : périphérique stockage
- lecteur disque : enregistre texte
- lecteur disquette : enregistre & lit info sur 1 disquette.
- lecteur CD/DVD : lire info sur CD-ROM CD = Compact Disc ROM: Read Only Memory
- lecteur DVD : lire images
- mémoire RAM : mémoire vive
- mémoire virtuelle = transfère par disque dur

* Alimentation → source d'énergie

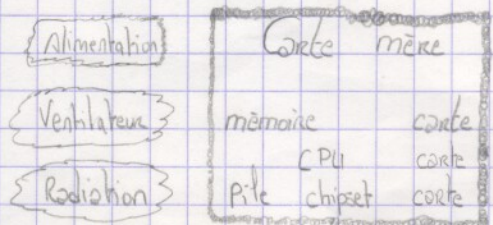
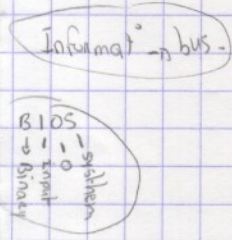
* Nappes connexion ; IDE ; SCSI

- disques IDE → Intelligent Drive Electronics → assure connex^o entre périph. stock. & processeur & transfère données

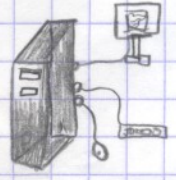
- nappes → relie avec carte mère au 2 périphérique stockage

- disques SCSI → Small Computer Systems Interface (scsi)

↳ interface entre disques & carte mère



UAL = unité arithmétique et logique (CPU)
UC = unité de contrôle (chipset)
mémoire : RAM - disque dur - bande mémoire



I ₁	I ₂	V	R
0	0	1	0
0	1	1	0
1	0	1	0
1	1	0	1

non et et

A	0	1
B	0	1
C	0	1

Porte non, NON (A ou B)

$$A \xrightarrow{1000} = 1 \times 2^4 + 0 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 0 \times 2^0$$

$$B \xrightarrow{0100} = 0 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 0 \times 2^0$$

$$A \times 2^2 + B \times 2^1 + C \times 2^0 = 2^1 + 2^0 + 2^1$$

ex: 101 = $1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0$

ex: 111 = $1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 = 4 + 2 + 1 = 7$

Langage logique des maths.

1. Vrai 0. Faux
TABLE DE VERITE

P	q
0	0
1	0
0	1
1	1

P	\bar{P}	p	q	$p \wedge q$	p	q	$p \vee q$
0	1	0	0	0	0	0	0
1	0	1	0	0	1	0	1
0	1	1	1	1	0	1	1
1	0	1	1	1	1	1	1

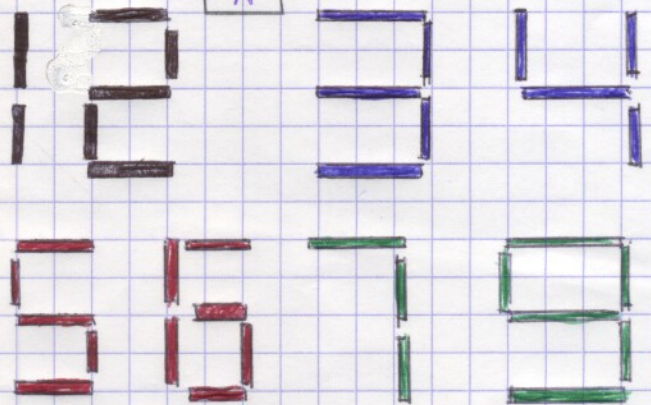
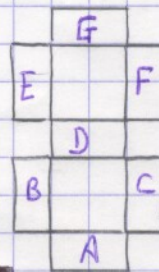
"et" → le seul cas où c'est vrai c'est 1 et 1 le reste est faux
 "ou" → " " " " faux " quand c'est dans aucun des cas.
 "ou exclusif" → il peut être soit ds un soit l'autre mais pas les deux.

p	q	$p \oplus q$
0	0	0
1	0	1
0	1	1
1	1	0

$$10000 = + 0 \times 2^0$$

Booléens...

	A	B	C	D	E	F	G
0	1	1	1	0	1	1	1
1	0	0	1	0	0	1	0
2	1	1	0	1	0	1	1
3	1	0	1	1	0	1	1
4	0	0	1	1	1	1	0
5	1	0	1	1	1	0	1
6	1	1	1	1	1	0	1
7	0	0	1	0	0	1	1
8	1	1	1	1	1	1	1
9	1	0	1	1	1	1	1



• $x = 10\ 000 \Rightarrow 16$

$$10\ 000 = 1 \times 2^4 + 0 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 0 \times 2^0 = 16$$

- si $0 < x \leq 9 \rightarrow$ afficher 0 sur D_1
- si $10 < x \leq 19 \rightarrow$ " 1 "
- " 20 " $29 \rightarrow$ " 2 "

memoriser la valeur decimale du 1^{er} digit
 \rightarrow multiplier la valeur du 1^{er} digit par 10 et enregistrer le resultat, soustraire ce resultat à la valeur
 \rightarrow Enregistrer ce nouveau resultat et l'utiliser pour afficher la valeur du deuxieme digit D_2 .

$\lfloor n \rfloor$ → aller à la ligne

```
a = 0
while (a < 7)
  a = a + 1
  print(a)
```

1
2
3
4
5
6
7

```
a = 2
b = 5
if (a == 5) & (b < 2):
  print "" & " signifie "et"; on peut aussi utiliser le mot "and"
```

```
c) a = 1
   if not a
     print 'gagné'
   elif a:
     print 'perdu'
```

```
a) a = 5
   b = 2
   if (a == 5) & (b < 2)
     print
```

isnbbs.fr → avoir une clé usb sous linux
/cle-isnb/

- 3). Écrire un pgm qui aff. 20 prem multiples de 7.
1) Écrire un pgm qui calcule le vol. d'un parallépipède.
2) Écrire un pgm qui affich. une suite de nb dont chq terme suit égale au triple du terme précédent.

```
1) p = input("entree la longueur : ")
   l = " " " longueur
   h = " " " hauteur
   v = p x l x h
   print v
```

```
3) a = 1
   for i in range(20):
     a = 3 * i
     print(i, a)
```

4)

```
2) a = 1
   for i in range(13):
     (tabulat) print(a)
     " " a = 3 * a
```

1
3
9
27
81
243
729
2187
6561
19683
59049
177147
531441

Python

Choice = input("Choisissez N")
 pour un entier rajouter int (à la fin)
 la syntaxe
 type booléen

if test : bloc d'instruct° si true
 else : bloc d'instruct° si non
 1er bloc → si test est évalué à true
 else → facultatif

non(A et B) = (non A) et (non B)
 non(A ou B) = (non A) et (non B)
 on s'arrête si une condit° est vérifiée, alors la condit° de la boucle est non(c)
 s'utilise quand on ne sait pas à priori le nombre d'itérat° de la boucle

afficher plusieurs valeurs par:
 par des virgules:
 >>> x + 2
 >>> print "HelloWorld", 4
 cette phrase est... "x + 2"
 print valeur

variable : input()
 print avant input ce qu'on attend de lui print "Donner un entier"
 n = input()
 attend que l'utilisateur tape qqch et appui sur entrée

boucle if:
 if est toujours suivi des :
 else :

boucle for
 For variable in sequence =
 instruct° 1
 ...
 séquence → liste, chaîne caractères ; n-uplet
 variable → 1 à 1 valeur élément sqe

Affectation
 valeur → 3, "hello world!", 10 + y, input (un entier)
 variable → x; a, liste, un élément
 chaîne de caractères ex: a = input("Entrez a = ")

Type numérique
 int
 float
 long
 + ; * ; / ; // ; %
 < , > , <= , >= , == , !=

Type booléen
 bool
 and et logique, or et logique, not non logique

séquence
 s[0] 1er élément de s; s[i] i+1ème élmt s[i:j] sous séq s[i:] ... s[j:]
 str, list
 + concaténation
 * \ multiple
 [] opérateur d'accès
 str non mutable, list mutable
 s → séquence (les éléments de s sont n° 0, 1 ... etc)
 n → la longueur de la séquence
 i ; j des entiers

I - Codage d'un nombre : Binaire - Décimal - Hexadécimal

Ecrire 135 en binaire. Justifier.

$135 = 67 \times 2 + 1$ $16 = 2 \times 8 + 0$ $3 \dots$ *à poursuivre*
 $67 = 33 \times 2 + 1$ ~~$135 = 1 \ 110$~~ 1000000
 $33 = 16 \times 2 + 1$

$$\frac{075}{20}$$

Ecrire 110111 en décimal. Justifier.

$110111 = 1 \times 2^5 + 1 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 = 55$
 $110111 = 100000 + 10000 + 1000 + 100 + 10 + 1$

Ecrire FF7A en décimal.

$FF7A = 15 \times 16^3 + 15 \times 16^2 + 7 \times 16^1 + 10 \times 16^0 = ?$ 65402

Ecrire 237 en hexadécimal.

ED

Adrien écrit un message binaire à Amélie en utilisant le code ASCII.

Décoder en français la suite binaire suivante pour connaître le message envoyé :

1010000 1001111 1010101 1010010 1010001 1010101 1001111 1001001

$1010000 = 80$ $1010101 = 85$ $1010001 = 81$ $1001111 = 79$
 $1001111 = 79$ $1010010 = 82$ $1010101 = 85$ $1001001 = 73$) oui.

→ ~~POUR QUOI~~
POUR QUOI

II - Python

1/ Remplir les pointillés :

Dans un terminal Python :

```
>>> 127/16
>>> 7.
>>> 127%16
>>> 15
```

2/ Ecrire la sortie obtenue avec le code suivant :

```
>>> for t in range(0,11):
...     print '2'+str(t)+' = ',2**t
...     Disp t
```

sortie obtenue :

```
---- if t > 11
---- return
```

$2^0 = 1$
 $2^1 = 2$
 $2^1 = 1024$

3/ Ecrire la sortie obtenue avec le code suivant :

```
>>> a, b = 7.3, 12
>>> y = 3*a + b/5
```

sortie obtenue :

$3 \times 7.3 + 12/5 = 24.3$ 23.9

4/ Ecrire la sortie obtenue avec le code suivant :

```
>>> M=2048
>>> x=2
>>> n=0
>>> y=1
>>> while (y<M):
...     y=x*y
...     n=n+1
...
>>> print(n)
```

sortie obtenue :

5/ Ecrire la sortie obtenue avec le code suivant :

```
>>> a, b, c = 1, 1, 1
>>> while c < 11 :
...     print b
...     a, b, c = b, a+b, c+1
```

sortie obtenue :

6/ Ecrire la sortie obtenue avec le code du programme suivant :

```
def volBoite(x1 =10, x2 =10, x3 =10):
# Volume d'une boîte parallélépipédique
return x1 * x2 * x3

print("Volume d'une boîte parallélépipédique")
print(volBoite())
print(volBoite(5))
print(volBoite(5, 3, 7))
```

sortie du programme :

7/ Ecrire la sortie obtenue avec le code du programme suivant :

```
def affiche_PuissanceDe2(a,b):
    for i in range(a,b):
        print(2**i)
affiche_PuissanceDe2(4,11)
```

sortie du programme :

III - Logique, table de vérité, booléens

1/ Compléter le tableau suivant :

p	q	non p	non q	p et q	p ou q	p et non q	non p ou q
0	0	1	1	0	0	0	0
0	1	1	0	0	1	0	1
1	0	0	1	0	1	1	0
1	1	0	0	1	1	0	1

2/ Remplir les sorties des codes suivants :

Dans un terminal Python :

```
>>> x=34
>>> y=15
>>> if (x<y or x==y) and x-y > y :
...     print("cas 1")
```



```
... else:
...     print("cas 2")
```

sortie :

```
.... 19 > 15
```

```
>>> x=361
>>> (x > 113) and (x < 227)
```

sortie :

```
.... return ("cas 1")
```

IV - Algorithmme

Albert a choisi un nombre compris entre 1 et 100, Bertrand doit le deviner. Bertrand fait des propositions et Albert répond "trop grand", "trop petit" ou "gagné". Le jeu s'arrête lorsque Bertrand a trouvé le nombre.

On décide d'appeler N le nombre caché choisi par Albert et R la réponse de Bertrand. Exemple de jeu : Albert a choisi 63. On obtient le tableau de jeu suivant :

N	R	Test	Réponse d'Albert
63	40	R < N	Trop petit
	80	R > N	Trop grand
	60	R < N	Trop petit
	65	R > N	Trop grand
	62	R < N	Trop petit
	63	R = N	Gagné !

*Il est du type nombre
R est du type nombre
Entee N
~~for N in range(1,100):~~
IF N > R
Disp "trop petit"
IF N < R
Disp "trop grand"*

*Else IF N = R
Disp "gagné"
End.*

*Co u bas p...
u u
algorithmme!*

On veut maintenant remplacer Albert par un ordinateur. Ecrire l'algorithmme à faire exécuter à l'ordinateur dans ce cas.

V - Page Web

Le code source suivant présente deux erreurs, lesquelles ?

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
<title> ISN TS2 </title>
<!-- Ceci est ma première page web -->
<meta name="author" content="Tata Tyty" />
<meta name="keywords" content="html,initiation" />
<body>
</head>
Ceci est <i> un texte </i> <b> simple </b>
<h1>Sommaire</h1>
<a href="#ancre1">Premier paragraphe<br>
<a href="#ancre2">Deuxième paragraphe<br>
<a href="#ancre3">Troisième paragraphe<br>
<h2 id="ancre1"> Premier niveau </h2>

<a href="http://fr.wikipedia.org/wiki/Marilyn_Monroe">

</a><i>Marilyn sur wikimedia commons</i>
<h2 id="ancre2"> Deuxième niveau </h2>
<br><br><br><br><br><br><br><br><br><br><br>
<h2 id="ancre3"> Troisième niveau </h2>
</html>
</body>
```

*de </head> doit être placé avant <body> ✓
de </body> doit être placé avant </html>. ✓*

~~1~~ 1 2 3 4 5 6 7 8 9 10
 A B C D E F
 11 12 13 14 15 16

	$237 = 14 \times 16 + 13$	
80	$= 5 \times 16 + 0$	50 P
89	$= 4 \times 16 + 15$	4F O
85	$= 5 \times 16 + 5$	55 U
82	$= 5 \times 16 + 2$	51 Q
81	$= 5 \times 16 + 1$	51 Q
85	$= 5 \times 16 + 5$	55 U
89	$= 4 \times 16 + 15$	4F O
83	$= 4 \times 16 + 9$	48 H

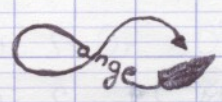
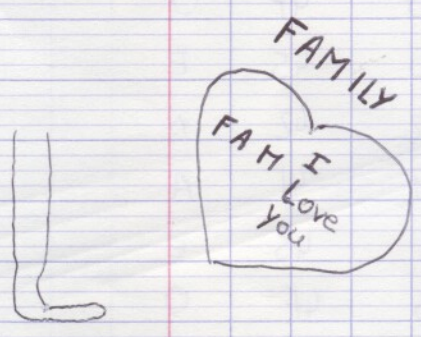


$$\begin{aligned}
 237 &= 2 \times 2^2 + 3 \times 2^1 + 7 \times 2^0 \\
 &= 8 + 6 + 7 = 21
 \end{aligned}$$

J'ai raté mon devoir d'ISN.

J'ai ratée mon devoir d'ISN.

Gwendoline Gwendoline Gwendoline Gwendoline



gwendoline

0 1 2 3 4 5 6 7 8 9 A B C D E F
 10 11 12 13 14 15

2^7 2^6 2^5 2^4 2^3 2^2 2^1 2^0

1 1 0 1 1 1

$= 1 \times 2^5 + 1 \times 2^4 + 1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0$
 $= 55$

FF7A = ~~$15 \times 16^3 + 15 \times 16^2 + 7 \times 16^1 + 10 \times 16^0$~~ $15 \times 16^3 + 15 \times 16^2 + 7 \times 16^1 + 10 \times 16^0 =$
 237 = $237 / 16 = 14$

2^7 2^6 2^5 2^4 2^3 2^2 2^1 2^0

1	0	1	0	0	0	0	0	$\rightarrow 64 + 16 = 80$
1	0	0	1	1	1	1	1	$\rightarrow 64 + 8 + 4 + 2 + 1 = 79$
1	0	1	0	1	0	1	1	$\rightarrow 64 + 16 + 4 + 1 = 85$
1	0	1	0	0	1	0	0	$\rightarrow 64 + 16 + 2 = 82$
1	0	1	0	0	0	1	1	$\rightarrow 64 + 16 + 1 = 81$
1	0	1	0	1	0	1	1	$\rightarrow 64 + 16 + 4 + 1 = 85$
1	0	0	1	1	1	1	1	$\rightarrow 64 + 8 + 4 + 2 + 1 = 79$
1	0	0	1	0	0	1	1	$\rightarrow 64 + 8 + 1 = 73$

$80 / 2 = 40$

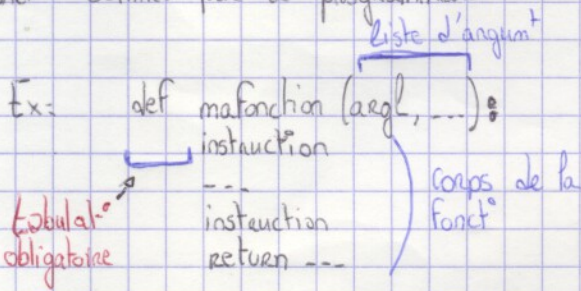
$80 = 40 \times 2 + 0$	@	$= 5 \times 16 + 0$	40	@
$79 = 39 \times 2 + 1$		$= 4 \times 16 + 15$	4F	0
$85 = 42 \times 2 + 1$	B	$= 5 \times 16 + 5$	45	E
$82 = 41 \times 2 + 0$	A	$= 5 \times 16 + 2$	42	D
$81 = 40 \times 2 + 1$	@	$= 5 \times 16 + 1$	41	A
$85 = 42 \times 2 + 1$	B	$= 5 \times 16 + 5$	45	E
$79 = 39 \times 2 + 1$		$= 4 \times 16 + 15$	4F	0
$73 = 36 \times 2 + 1$		$= 4 \times 16 + 9$	39	

Fonctions.

- petit ensemble d'instruct^o, mini programme
- 1 fonction = 1 tâche simple bien précise
- utilise une fonction = 1 appel à la fonction.
- Fonct^o prédéfinies
- Fonct^o définies par le programmeur

$$f(x) = x^2 \rightarrow f(2)$$

return = renvoie au principal



$$f(2) \text{ return } 2^2$$

Si une fonct^o juste d'affichage - def vide.
 def → introduit la définit^o de la fonct^o.

- argum^o dans la définit^o = paramètres formels.
- variable globale = visible ds le pgm princile.
- on ne modifie pas direct^{mt} une variable globale ds une fonct^o.

```
def afficheMontant (n, devise):
    print n, devise
    pu = 150
    print "at article coûte"
    afficheMontant "pu", "euro"
```

Tâche = demander un entier positif à l'utilisateur

1. demander un entier à l'utilisateur
2. vérifier que cet entier est positif sinon reprendre au 1.
3. mettre l'entier positif à disposit^o de celui qui a fait la demande.

```
def EntierPositif():
```

```
  a = -1
```

```
  while a < 0:
```

```
    a = input("entrez un nb entier positif")
```

```
  return a
```

```
print("Afficher le nombre entré par l'utilisateur : ", EntierPositif())
```

Run sur python IDLE test notre algo.

"\n" → retour à la ligne main = principale.

binéaire - booléen - lire et tracer les résultats.

• Programme pour calculer et un peu afficher le discriminant

```
a = input("entree a: ")
```

```
b = input("entree b: ")
```

```
c = input("entree c: ")
```

```
discriminant
```

Correction:

```
def entree les coefficients ():
```

```
    a = input("entree a: ")
```

```
    b = input("entree b: ")
```

```
    c = input("entree c: ")
```

```
    return a, b, c
```

```
def discriminant (a, b, c):
```

```
    discriminant = b*b - 4 * a * c
```

```
    return discriminant
```

```
def Affiche (d):
```

```
    print ("le discriminant est: ", d, "\n")
```

```
def main ():
```

```
    a, b, c = entree les coefficients ()
```

```
    affiche (discriminant (a, b, c))
```

```
main ()
```

ordinateur = BIOS - Carte mère Alimentat° - Carte - Perce

ajoute des images ...

```
def Entree nombre ():
```

```
    m = 0
```

```
    while m < 2 or m > 13:
```

```
        m = int(input("Entrez un entier compris entre 2 et 13: "))
```

```
    return m
```

```
def table Multiplic (n):
```

```
    for i in range (1, n):
```

```
        c = str(i) + " * " + str(n) + " = " + str(n * i) + "\n" # chaîne de l'affichage
```

```
        print(c)
```

```
box = Entree Nombre ()
```

```
TableMultiplic (box)
```

Arduino

* Installation: d'Arduino IDE:

- session Ubuntu sous Lycee7
- mdp: Lycee974
- ouvrir un terminal et entrez commande:
sudo apt-get install arduino
- + niveau mdp
- laissez changer.

* Carte Arduino:

Diode: Une diode sa ne laisse passer le courant que dans un sens.
LED: Diode Electro-Luminescente; laisse passer un courant très faible.

led → le sens est alphabétique anode & cathode. borne + grande → anode branche sur +.

Applicat^o. des LED.

repère 5V est mettre au niveau de anode.
↳ fil noir " " cathode - résistance.

Branchez par USB sur ordi.

Ouvrez Arduino 1.0

Open → Basic → Blink + (⊕) } il faut que sa clignote orange.
↳ (envoyer)

* langage Arduino:

- langage type (avant d'écrire il faut définir la variable), basé sur le langage de programmation "C".
[variables: boucle, Fonct^o].
 - séparé les + blocs → { }
 - // → commentaire
 - void setup → configurat^o
 - * / → Fonct^o
 - void loop → boucle infini
 - digitalWrite → écrire une sortie et la valeur
 - delay → attendre
 - void → fonct^o qui ne prend aucun paramètre (renvoie ∅ variable).
- digitalWrite (13, HIGH) ; digitalWrite (13, LOW)
delay (1000)



Applicat° :

Gregory

1^{er} projet:

```

void setup () {
  // initialise the digital pin as an output.
  // pin 13 has an LED connected on most Arduino boards:
  pinMode (13, OUTPUT);
  pinMode (5, OUTPUT);
}

void loop () {
  digitalWrite (13, HIGH); // set the LED on
  delay (1000); // wait for a second
  digitalWrite (13, LOW); // set the LED off
  digitalWrite (5, HIGH); // set the LED on
  delay (1000); // wait for a second
  digitalWrite (5, LOW); // set the LED off
}

```

2^{ème} projet:

ajouter une ligne pour LED

3^{ème} projet:

```

int ledR = 13; // on affecte la valeur entière 13 à la variable ledR
int ledV = 12; // " " " " " " " " ledV
int ledJ = 11; // " " " " " " " " ledJ

void setup () // Initialisat° de la carte
{
  pinMode (ledR, OUTPUT); // broche reliée à la LED rouge (13 en sortie)
  pinMode (ledV, OUTPUT); // " " " " " " verte (12 " " )
  pinMode (ledJ, OUTPUT); // " " " " " " jaune (11 " " )
}

void loop () // fonction principale, elle se répète à l'infini
{
  clign (ledR);
  clign (ledV);
  clign (ledJ);
}

void clign (int a) // fonct° de clignotement
{
  digitalWrite (a, LOW); // Allume la LED
  delay (1000); // * Attendre 1 seconde * de programmation
  digitalWrite (a, HIGH); // Eteindre la LED
}

```

3^e projet:

```

long temps; // variable qui stocke le temps
boolean etat_led;
void setup ()
{
  pinMode (13, OUTPUT);
  Serial.begin (9600);
}

void loop ()
{
  if ((millis () - temps) >= 1000)
  {
    (suite)
  }
}

```

```

(suite)
etat_led = !etat_led; // on inverse l'état de la
digitalWrite (13, etat_led); // on allume ou éteint
temps = millis (); // on stocke la nouvelle heure
Serial.println (temps);
Serial.println (etat_led);
}

```

Séance arduino N°3

Utilisation du moniteur série

1. En vous aidant de l'aide en ligne arduino commenter le code :

```
long temps; //variable qui stocke le temps.
boolean etat_led; //
void setup()
{
  pinMode (13, OUTPUT);
  Serial.begin(9600);
}
void loop()
{
  if((millis() - temps) >= 1000)
  {
    etat_led = !etat_led; //on inverse l'état de la LED.
    digitalWrite(13, etat_led); //on allume ou éteint.
    temps = millis(); // on stocke la nouvelle heuve
    Serial.println(temps);
    Serial.println(etat_led);
  }
}
```

2. Téléverser le programme sur une carte arduino.
3. Modifier le code de façon à faire clignoter la LED 5 fois par seconde. Surligner la partie du code à modifier et indiquer en rouge la valeur à remplacer.
4. Vérifier le fonctionnement du programme puis remettre un clignotement avec une période de 2 secondes.
5. Téléverser puis ouvrir le moniteur série. (Bouton en haut à droite).
 - Quelles sont les informations qui s'affichent dans le moniteur ?
6. Modifier le code pour que le moniteur série se comporte comme un chronomètre affichant le temps en secondes.
7. Protection du moteur d'un robot

Un robot avance jusqu'à ce qu'il rencontre un obstacle.

Lorsque le moteur du robot fonctionne un gyrophare (LED orange) clignote.

Lorsque le robot rencontre un obstacle (pression sur un bouton poussoir).

- On coupe l'alimentation du moteur. (on remplace le moteur par une LED verte).
 - La LED orange reste allumée.
8. Modifier le code précédent pour écrire un programme correspondant à ce comportement.
 9. En quoi l'utilisation de la commande « delay() » pour faire clignoter la LED peut-elle être problématique pour le moteur du robot ?

Utilisation d'une entrée analogique

Potentiomètre

Un potentiomètre est une résistance variable à trois bornes. Entre les bornes A et B la résistance R est constante. Un curseur permet de faire varier la position du point C entre les points A et B ce qui a pour effet de faire varier la valeur des résistances R_1 et R_2 entre 0 et R.

Remarque : On a toujours $R_1 + R_2 = R$

Si on applique une tension U entre les points A et B, le curseur du potentiomètre permet de faire varier les tensions U_{AC} et U_{BC} entre 0 et U.

<https://encrypted-tbn3.gstatic.com/im>

Montage

Relier :

- le point A du potentiomètre à la source 5 V de la carte arduino.
- le point B du potentiomètre à la masse GND.
- le point C du potentiomètre à l'entrée analogique A0.

1. Taper le code suivant :

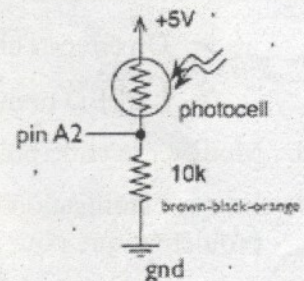
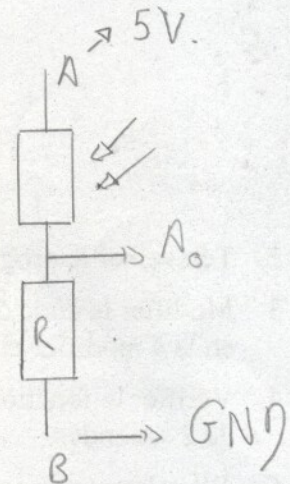
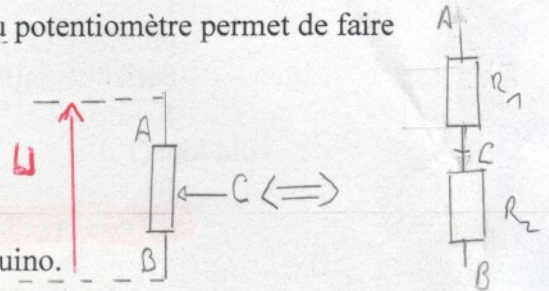
```
const int analogInPin = A0;
int sensorValue = 0;

void setup()
{
  Serial.begin(9600);
}

void loop()
{
  sensorValue = analogRead(analogInPin);
  Serial.println(sensorValue);
  delay(1000);
}
```

1. Téléverser le programme sur une carte arduino.
2. Ouvrir le moniteur série et manipuler le potentiomètre.
3. Que fait ce programme.
4. Commenter le code.
5. On remplace le potentiomètre par une photorésistance en série avec une résistance de 10 kΩ.

- Quel est l'intérêt de ce montage ?



ISN:

Surfer sur Internet : quels sont les risques et comment se protéger ?

Plan:

I. Surfer sur Internet.

ID₁: le monde de Internet.

ID₂: Explicat° de "surfer sur Internet"

II. des risques que nous pouvons avoir en surfant sur Internet.

ID₁: risques de l'utilisateur.

ID₂: Donner des exemples.

III. Comment pouvons nous, nous protéger ?

ID₁:

ID₂:

I. Internet

Def: * Internet est un système d'interconnexion de machines qui constitue un réseau informatique mondial, utilisant un ensemble de protocoles de transfert de données.

IP est composé de millions de réseaux aussi bien publics, privés, universitaires, commerciaux et gouvernementaux.

IP transporte un large spectre d'informatic° et permet l'élaborat° d'applicat° et de services variés. (courrier électronique, messagerie instantanée, World Wide Web.)

* L'accès à Internet peut-être obtenu grâce à un fournisseur d'accès à Internet.

* Adresse IP (internet protocol) identifie une machine sur le réseau permet la connectat° d'une machine à une autre.

* Réseau (passerelle, gateway) routeur, élément du réseau qui permet de passer d'un réseau à un autre.

* DNS (Domain Name Service) → serveurs de nom.

* Naviguer sur le Net: Internet sur ordinateur

:1021

déf: surfer? ou naviguer sur le web signifie visiter le monde électronique en se déplaçant sur le "World Wide Web" de site en site.

* avantages:

- C'est une source d'informati° (tous types de renseignements sur tous les sujets possibles)
- Sites favoris (sites concernant vos centres d'intérêt comme par ex: musique, déco, sport...)
- Contact personnels: (chat...)

II. Risques:

- publicité (jeunes considérés comme consommateurs, bcp msg publicitaires. entreprises connaissent le pt faible des jeunes).
- contenu nuisible (enfants → confrontés à des contenus préjudiciables; site racistes - sexistes) => informat° écartée image toxique (violence...)
- informat° non fiables: (tâches qui incombent à l'enseignem^t.)
- E. privacy: donnent volontairement & facilement données personnelles.
- vie privée:
 - informat° personnelles visibles par tous.

ex: Facebook. diffus° de toutes infos peut avoir des csq graves.

- accès à des données non appropriées.
- Virus. savoir faire la différence entre virus et canular.
- logiciel esp° compromettre confidentialité des données perso.
- risques liés au piratage.
- de phishing (courriel électronique).

III. Se protéger...

* de système de contrôle parentaux.

le contrôle parental est un type de système qui permet aux parents de restreindre automatiquement l'accès de leurs enfants à un média en le limitant à certaines catégories de contenus afin de les protéger contre (pédophilie, contenus choquants.)

* pare-feu (firewall)

* conseils:

- Éviter la communicat° d'info personnelle.
- " photos / vidéos perso si elles sont de nature à porter préjudice à soi ou à autrui.
- Refuser la publicat° en tagg.
- utiliser WOT → averti quand on va sur un site dangereux.
- avoir un pseudonyme
- antivirus.
- courriel : protéger vs échanges - Spam.

Bilan:

* Sur internet, nous courons des risques.

En effet, naviguer sur internet peut être un avantage mais aussi un risque.

D'un point de vue, Internet c'est un réseau "externe", public qui permet de nous informer (informat°...), c'est aussi un moyen de distract° pour les internautes (centr. internet, écolo, musique, etc.). Mais, naviguer sur internet c'est prendre des risques. En effet, quand nous surfons sur internet, nous sommes victimes des sites malveillants, contenu nuisible, virus. Mais nous nous étions victimes volontairement c'est à dire nous sommes tous sur des réseaux sociaux pour parler, partager des moments avec d'autres internautes. Par exemple, Facebook c'est un risque par le jeu car nous divulguons notre vie privée à travers des vidéos, photos, compte (non sécurisé) et souvent des jeux sont victimes de piratages.

Autre exemple de risques quand nous surfons sur Internet c'est les ~~jeux~~ sites qui attirent le plus les enfants. Les enfants eux aussi courent

* Ressources:

- Wikipédia:

- Webetaic: www.webetic.be

un grave danger en allant sur des sites non appropriés pour eux.

Et aussi la fausse information peut nuire à leurs études.

Pour cela, nous pouvons nous protéger de ces risques de navigation.

Nous pouvons mettre un système de contrôle parental, avoir un pseudonyme pour ne pas publier nos infos personnelles, protéger nos données, activités.

Algorithme

déf^o → est une suite finie et non ambiguë d'opé^o ou d'instruct^o permettant de résoudre un problème.

mode d'express^o

→ début fin.

lire - écrire

Si ... alors ... sinon

tant que ...

→ Lire a, b, c

Si $a \neq 0$ alors $\Delta = b^2 - 4ac$

sinon fin.

affichez a.

Si $\Delta < 0$ alors

affichage de "pas de racine"

sinon

x_1 prend la valeur $\frac{-b - \sqrt{\Delta}}{2a}$

x_2 prend la valeur $\frac{-b + \sqrt{\Delta}}{2a}$

affichez x_1

affichez x_2

organisat^o

→ • nom

• parties déclaratives

x constantes:

- définie.

x variables:

- nom

- type

- valeur

• partie exécutive:

x début - fin.

x commentaires:

- /* cmt plusieurs.

- lignes */

- // cmt se 1 seule ligne

• act^o bas:

x affectat^o:

- en prendo - code instruct^o x note $a \leftarrow 2$.

- incrémentat^o : $i++$; $i--$; $i \leftarrow i+1$

- une instruct^o d'affect^o ne modifie que ce qui est à gauche de l'opé^o.

• lecture - Écriture: **algèbre**

x structures alternatives:

- si condit° alors

sinon

fin si

x Expressions Booléennes:

- valeur 1: act° 1

valeur 2: act° 2

fin Selon

valeur n: act° n

fin Selon

autrement: act° 0

fin Selon

• structures répétitives:

- boucle de comptage

- compteur de boucle

- pour i allant de i₀ à i_n

faire

fin pour

- test en tête de boucle

→ Tant que (condit°)

faire

act° 1

act° 2

fin tant que

i ← 0
(ex: Pour i allant de 1 à 4)
faire
|
|
|
fin pour

faire en sorte que
soit juste -
définir les variables
si besoin

ex:

u ← 0

v ← 5

w ← 10

Tant que $u * v \geq w$

Pour que la boucle soit finie, il faut que l'exécution de l'act° modifie la valeur de l'express° conditionnelle évaluée par le test de sortie de la boucle.

Veiller à initialiser les variables intervenant vs la condit°.

applicat:

Variables:

a → début

b → fin

c → pas

a est du type nombre début ; b est du type nombre fin ; c est du type nombre pas.

Début pour:

$f(A)$ prend la valeur $A^5 - 3A^4 + 2A^3 - 7A + 1$

A prend la valeur $A + pas$

Si $A \geq B + pas$

alors

calculer $f(A)$

sinon

afficher A .

correction:

calcul du tableau de valeurs de la fonction $f(x) = x^5 - 3x^4 + 2x^3 - 7x + 1$

- Variables:

debut, fin, pas: entiers naturels

A, B, C: réels

- Début:

Lire debut, fin

A ← debut

B ← fin

C ← pas

} /* phase
d'initialisation

Faire

afficher fonction (A)

A ← A + pas

Tant que (A < B + pas)

Fin

Définir fonction (A):

fonction (A) ← $A^5 - 3A^4 + 2A^3 - 7A + 1$

* une négat° s'écrit
toujours inférieur stricte.

Calcul d'une suite $U_n = q \times n$

Variables:

raison, a_0 , 0 (compteur) : entiers naturels

R, A, I : réels

Debut:

Lire compteur, raison, a_0

I ← 0 (compteur)

A ← a_0

R ← raison

Faire:

A ← $R \times A$

I ← I + 1

Tant que (A < 100)

Faire

Afficher I

Afficher A

Si non

calculer a_n

Fin tant que

Afficher A.

Calcul des termes successifs
d'une suite géométrique

Variables:

a_0 , raison, réels

R, A : réels

i : entier naturel ^{l'indice de début}

Debut:

entree A_0 , raison

R ← raison

A ← a_0

i ← 0

Faire

i ← i + 1

A ← $R \times A$

Tant que (A > 100)

Afficher "de premier indice par lequel a_n dépasse 100 est:"

Afficher "la valeur du n^{e} terme dépassant 100 est de:" A

Fin

→ le string → str

str(b) → b → 14,2

le string permet d'afficher le chaîne de caractère.

Exercices:

Recherche de limite approchée de suite géométrique convergente.

Variables:
R: réels // raison
U₀: réels // terme initial
N, B: réel // terme successifs de la suite
I: entier // compteur
debut
R = 2
Tant que R > 1
Entrez R
A ← U₀
I ← 0 // initialiser du compteur
faire
I ← I + 1
B ← A
A ← R * A
Tant que |A - B| >= 10⁻³
Afficher I
Afficher "u_{n-1} =", B
Afficher "u_n =", A
Afficher "Ecart =", B - A
fin

faire pour exo 2.
un module pour

- * - , coding: utf-8 - * -

limite de suite géométrique convergente
from math import *

R = 2

while R > 1:

R = float(input("raison = "))

u₀ = float(input("u₀ = "))

a = u₀

i = 0

while True:

i + 1

b = a

a = R * a

if fabs(b - a) <= 10⁻³:

break

print("indice courant n =", str(i))

print("u_{n-1} =", str(b))

print("u_n =", str(a)).

ISN: Exercices.

Choix = ↑ input ("Choisir N")
inv()

Algo devoirs:

```

Choisie N
Initialise R
R → 0
R → réponse de Bertrand
Si R < N alors
Afficher "trop petit"
Si R > N alors
Afficher "trop grand"
Si non
Afficher gagné

```

Python:

```

"Choisir N"
R = input("réponse de Bertrand")
if R < N:
    print("trop petit")
    break
if R > N:
    print("# trop grand")
    break
print("# gagné")

```

① Variables:
R, A, entiers naturels
I: réels

```

Début:
R ← réponse
A ← b0
I ← 0
Faire I ← I + 1
B ← A // sauvegarde bn-1
A ← R * A
Tant que |A - B| < 10-3
Afficher I (n0)
Afficher B (bn-1)
Afficher B - A
fin

```

Input (R, A, entiers naturels, I)

```

R = réponse
A = b0
I = 0
I = I + 1
B = A # "sauvegarde bn-1"
A = R * A
While |A - B| < 10-3
Print I (n0)
Print B (bn-1)
Print B - A

```

② Variables
A, I, réels

```

Début
A ← U1
I ← 1
Faire
I ← I + 1
A ← 1 + 1/A
afficher I, A
Si I = I FIN
fin

```

```

A = "U1"
I = 1

```

```

I = I + 1
A = 1 + 1/A
Print I, A
if I = I
fin

```

Line N
 $S = 0$
 input (i, 1, n)
 $S = S + i$
 print S

③

Variables
 $S, n = \text{réels}$
 $E; \text{réels}$

debut

Line N

 $S \leftarrow 0$

Pour i allant de 1 à n

 $S \leftarrow S + i$

fin pour

Afficher S

fin.

④

I, n, k, a = réels

 $U_0 \leftarrow 0$

Lire n, k

Pour I allant de 1 à n

 $I \leftarrow I + 1$ $U_n \leftarrow \frac{1}{k^{na}} + U_n$

fin pour

Afficher U_n $U_0 = 0$

Input n, k

Input i = (1; n)

 $I = i ++$ $U_n = 1/k^{na} + U_n$

= Break

Print U_n

⑤

Calculer la somme des éléments d'un tableau

Créer une page web

• Le langage HTML :

HTML: Hyper text mark up language

- HTML (le squelette).
 - DOCTYPE se trouve sur w3c (consensus internationale) où on trouve tout les doctype.
 - Langage à Balise c'est ce qu'on met en `< >`. sa indique le début. **obligatoire**
ex: `<style > ... </style >`
 - une feuille de style → définit les titres, arrière plan ...
dans le "head" il faut l'auteur, le titre, les mots clés adéquats à notre page et définit sa page.
 - commentaire : commence par `<!--` et se termine par `-->`
 - body `<body >` c'est le contenu de la page.
 - Une balise ouvrante `<html >` puis `</html >`. Tout ce qui est dans le `<head >` n'apparaît pas dans la page (référentiel) mais si ce qui a été le body apparaît.
 - ` ... ` permet d'avoir une énumération [ex (le sommaire)]
 - `<table > ... </table >` **tableau**
 - insère une image `<tr > <td > `
 - écriture en italique `<i > ... </i >` ou gras ` ... `
le World Wide Web (www) → hypertexte public
HTML → (langage dit de marquage)
balise → indique la façon dont doit être présentée le document.
- pour écrire une page web il faut : éditeur de texte → `<html > <head > </head > <body > </body > </html >`.

- balise = titre on met dans head. (`<title >`)
- paragraphe `<p > ... </p >` et le retour à la ligne est `
`
- mix en forme de texte (balise `<hr >`)
- hypertexte (balise `<a >`)

• Les feuilles de style utilise un langage CSS.

- permettre de personnaliser écrit grâce aux balises HTML.
- les paramètres de body (permet de modifier principalement) → `color` ...

```
ex: h1 {  
    color : # 999 ;  
    font-family :  
    ...  
}  
h2 {  
    }  
}
```

• Syntaxe:

crée un lien mp3 = ` ... `

crée une image = ``

il faut cliquer sur site du zéro

" un lien vers un autre site = ` Site du Zéro `

" lien vers une balise: ` la cuisine <br / >`

" lien vers une autre page: ` la page 2 de mon site `

Créer un lien vers une autre ^{balise:} `<h2 id="cuisine">de cuisine </h2>`

1^{er} page web: avec un titre dans le body insérer une image l'ien externe contenu de la page

Sm: H: gpa: gpa: ab

2^e page web: appelé la 1^{er} page

appelé une idée `<h2 id=" ... </h2>`

`<meta http-equiv="content-Type" content="text/html; charset=UTF-8" />`
↳ pareil écrit en caractère

`<title> ...`

`<meta name="author" content="mon nom" />`

`<meta name="Keywords" content="html, initiation" />`

`<meta http-equiv="content-Type" content="text/html; charset=UTF-8" />`

- ma 1^{er} page s'appelle obligatoirement `index.html`
- crée un document image, music

``

``

``
``
``

avoir un lien en cliquant sur l'image

220

System

Exposés.

gp. 1.

la commission nationale l'informatique et des libertés 28/11

CNIL: autorité administrative. afin que le citoyen n'est pas atteinte: identité, vie privée, liberté individuelles & publiques.

21 mars 1974: projet: (Système automatisé par les fichiers administratifs & répertoire des indiv.) SARIAR
CNIL accorde librement ces collaborations.

Missions & Fonct:

informer; réguler, sanctionner (le qui ne respecte pas la loi), protéger (banque, droit)
contrôler (si respect des fichiers info). anticiper:

exemple d'intervent:

- utilisation abusive des fichiers info:

(liste fichier de ces clients). → fichier client doit déclaré à la CNIL.

→ CNIL: protéger perso. consulte et projette loi. lutte contre Hchgmt illégaux.

↳ CNIL assure la protect^{sur} des personnes. de leur vie privée & des Hchgmt illégaux.

des 4 types de licences et les droits et devoirs des utilisateurs.

- licence: autorisat° délivrée à l'utilisateur sous forme de contrat.

↳ ex: Showmax.

- licence libre: respecte le droit d'auteur.

↳ ex: Stackware. LGPL - Creative Commons - FDL.

- les droits et devoirs des utilisateurs.

Internet → espace liberté.

droits = code civil et individus possèdent un droit et respect à sa vie privée.

devoirs = (plusieurs devoirs → respect l'auteur).

gp. 3

2009 x

gp 5.

la persistance des données.

1. 99

- > réseaux sociaux -> danger pour la vie privée.
- privé -> navigat^r -> paramètres.
- les moteurs recherche disposent tout autre d'intérêt.
- en fin^r -> localisat^r -> publicitaires.
- https = confidentialité fb.
- x VPN: Virtual Private Network.
- x cookies: repère les act^r de l'internaute.
- x https: chiffré. -> fichiers -> navigateur privée.
- VPN: sécurité logique privée.
- ✓ adresses IP -> cachée pour un adresse à l'étranger.

Dropbox.

des compétences d'évaluat° pour ISN:

- C1: Décrire, expliquer une situat°, système ou un pgeme.
- C2: Concevoir et réaliser une solut° informatique en rpe à un pb.
- C3: Collaborer efficacement au sein d'une équipe ds le cadre
- C4: Communiquer à l'écrit et à l'oral
- C5: Faire un usage responsable

* Page Web: problème.

index.html savoir les liens - des images: ex: a href /images /image 1. png'
renvoie respect. image - média - index - page2 - page3 - style

modifier: éditem : restructura , remplacer

* pygal:

recherche pygal sur internet. (documentat - interpolat°.

dans un terminal.

installer pygal:

```
$ sudo apt-get install python-pip
```

```
$ sudo pip install pygal.
```