

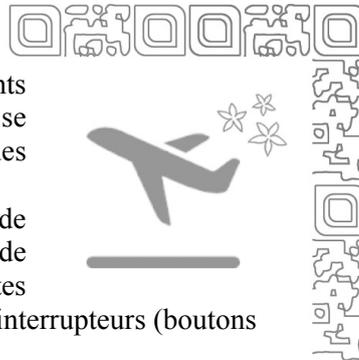


Les enfants à l'aéroport

1- Enoncé

Dans le cadre d'une sortie scolaire, 28 enfants visitent la tour de contrôle de l'aéroport qui se trouve proche de leur école. Ils portent des brassards numérotés de 1 à 28.

Un tableau de commande dans cette tour de contrôle permet l'allumage de 196 feux de signalisation pour les différentes pistes d'atterrissage. Ces feux sont commandés par des interrupteurs (boutons poussoirs) numérotés de 1 à 196.



Afin de passer le temps, les enfants appuient sur les interrupteurs à tour de rôle. L'enfant N°1 appuie sur tous les interrupteurs dont les numéros sont des multiples de 1, c'est-à-dire tous les interrupteurs.

L'enfant N°2 appuie ensuite sur tous les interrupteurs dont les numéros sont des multiples de 2 : 2, 4, 6, 8, etc.

L'enfant N°3 appuie ensuite sur les interrupteurs 3, 6, 9, 12, etc.

Tous les enfants appuient donc sur les interrupteurs à tour de rôle jusqu'au dernier, l'enfant N°28, qui appuie alors sur les interrupteurs 28, 56, 84, 112, etc.

Avant l'arrivée des enfants, tous les feux de signalisation étaient éteints.

2- Travail demandé

Niveau ☀

Ecrire un algorithme puis un programme permettant d'afficher l'état des feux de signalisation après le passage des 28 enfants. Pour cela, afficher un vecteur ligne comportant des 0 (pour les feux éteints) et des 1 (pour les feux allumés). Afficher également le nombre de feux de signalisation allumés.

Niveau ☀☀

Modifier le programme afin qu'il soit possible de choisir le nombre d'enfants, E , et le nombre de feux de signalisation, N . Le programme doit afficher l'état des feux ainsi que la liste des numéros des feux allumés.

Lancer le programme avec $N = 196$ et $E = 28$. Quelle remarque peut-on faire à propos des numéros des feux allumés ?

Niveau ☀☀☀

Modifier le programme afin de répondre au problème suivant :

Lorsque $N = 196$, représenter le nuage de points du nombre de feux allumés après le passage des enfants en fonction du nombre d'enfants, E .

Remarque : afin de limiter les temps de calcul, on prendra des multiples de 28 pour le nombre d'enfants : 28, 56, 84, ..., 196.

3- Les fonctions à utiliser

3-1. Fonctions Scilab

input() ; zeros() ; linspace() ; modulo() ; int() ; sum() ; printf() ; disp() ; plot()

3-2. Fonctions Python

input() ; zeros() ; linspace() ; a%b ; int() ; sum() ; print() ; plot() ; show()

4- Solution du niveau ☀

4-1. L'algorithme

Entrées

Aucune

Traitement

```
Définir le vecteur F
for e = 1:28
    for k = 1:196
        if modulo(k,e) == 0 then
            if F(1,k) == 0 then
                F(1,k) = 1
            else
                F(1,k) = 0
            end
        end
    end
end
end
```

NumF prend la valeur sum(F)

Sorties

Afficher le vecteur F
Afficher le nombre NumF

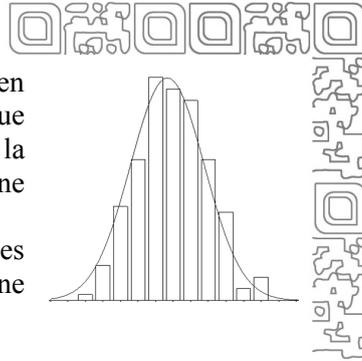


Répartition des notes à un examen

1- Énoncé

Un nombre K d'étudiants composent à un examen dont le sujet comporte 20 questions. Chaque question rapporte un point. On considère que la probabilité qu'un étudiant réponde juste à une question est égale à $1/3$.

Il s'agit d'établir une prévision des notes obtenues à cet examen en effectuant la simulation d'une expérience aléatoire.



2- Travail demandé

Niveau ☼

Ecrire un algorithme puis un programme permettant à l'utilisateur de saisir le nombre K d'étudiants et d'afficher l'histogramme des effectifs des notes obtenues suite à cette simulation.

Niveau ☼☼

Modifier le programme afin d'afficher l'histogramme des fréquences des notes obtenues.

Niveau ☼☼☼

Tracer la courbe de la loi normale $N(\mu = n\mu', \sigma = \sqrt{n}\cdot\sigma')$ sur l'histogramme des fréquences des notes obtenues avec $n = 20$, $\mu' = 1/3$ et $\sigma' = 0,471$. On rappelle la fonction densité de fréquence de la loi normale :

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

Remarque : $\mu' = 1/3$ et $\sigma' = 0,471$ sont la moyenne et l'écart type pour la loi de probabilité X de la note obtenue à une question. On a les probabilités suivantes pour cette loi : $P(X = 0) = 2/3$ et $P(X = 1) = 1/3$.



Calcul approché d'une intégrale

1- Enoncé



On considère l'intégrale $I = \int_0^1 f(x).dx$, où la fonction

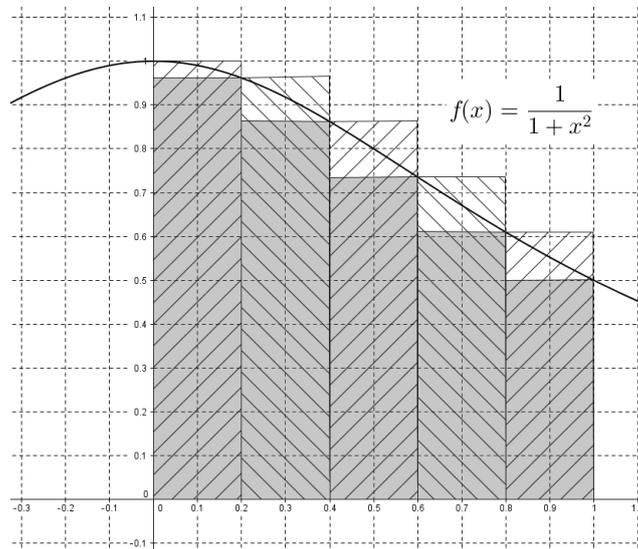
$$I = \int_0^1 f(x).dx$$

f est définie, pour tout réel x par : $f(x) = \frac{1}{1+x^2}$



Il s'agit de déterminer un encadrement de cette intégrale I en utilisant la méthode dite des rectangles.

La figure ci-dessous correspond au cas où $N = 5$.



• Méthode utilisée

On choisit un nombre de rectangles, N , de même largeur, h , avec $h = \frac{1-0}{N} = \frac{1}{N}$

La valeur de l'intégrale I est minorée par la somme des aires des rectangles grisés et majorée par la somme des aires des rectangles hachurés (voir la figure ci-avant).

On établit alors l'encadrement suivant :

$$h \times f(1) + \sum_{k=1}^{k=N-1} h \times f(k \times h) < I < h \times f(0) + \sum_{k=1}^{k=N-1} h \times f(k \times h)$$

Si on pose $K(h) = \sum_{k=1}^{k=N-1} h \times f(k \times h)$, on obtient :

$$h \times f(1) + K(h) < I < h \times f(0) + K(h)$$

On en déduit alors l'expression de l'amplitude d de cet encadrement :

$$d = h \times [f(0) - f(1)] = \frac{h}{2}$$

2- Travail demandé

Niveau ☼

Ecrire un algorithme puis un programme permettant de déterminer un encadrement de l'intégrale I . Le programme doit permettre la saisie du nombre N de rectangles à considérer et il doit également afficher l'amplitude obtenue pour cet encadrement.

Niveau ☼☼

Modifier le programme afin de permettre à l'utilisateur de saisir l'amplitude d maximale de cet encadrement. Ce programme doit afficher le nombre N de rectangles utilisés ainsi que l'amplitude réelle obtenue.

Niveau ☼☼☼

On donne le résultat suivant : $I = \int_0^1 \frac{dx}{1+x^2} = \frac{\pi}{4}$

Modifier le programme afin de déterminer un encadrement de la valeur de π . Le programme doit permettre la saisie de l'amplitude δ maximale de cet encadrement. Il doit également afficher le nombre N de rectangles utilisés ainsi que l'amplitude réelle obtenue.

Informations complémentaires :

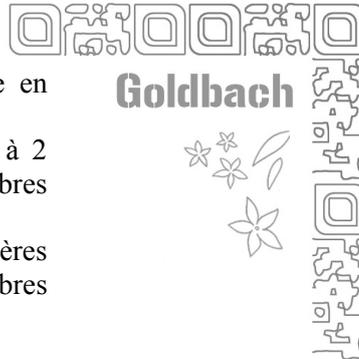
Relations entre les différentes grandeurs : $\delta = 4 \times d = 2 \times h$

On donne les premières décimales de Pi : $\pi = 3,141592653589793\dots$



Goldbach et les sommes de nombres premiers

1- Enoncé



La conjecture de Christian Goldbach, formulée en 1742, s'énonce ainsi :

"Tout nombre entier pair strictement supérieur à 2 peut s'écrire comme la somme de deux nombres premiers".

Par exemple le nombre 26 peut s'écrire de 3 manières différentes comme la somme de deux nombres premiers : $3 + 23$, $7 + 19$ et $13 + 13$.

On dira donc que le nombre 26 génère 3 sommes de deux nombres premiers.

2- Travail demandé

Niveau ☼

Ecrire un algorithme puis un programme permettant de déterminer si un nombre n est premier. Le programme doit permettre la saisie du nombre n et afficher le résultat. On pourra créer une fonction nommée *estpremier(n)* qui renvoie « Vrai » si le nombre est premier et « Faux » sinon.

Niveau ☼ ☼

Ecrire un algorithme puis un programme permettant de déterminer toutes les sommes de deux nombres premiers égales à N . Le programme doit permettre la saisie du nombre N et afficher tous les couples possibles.

Niveau ☼ ☼ ☼

Modifier le programme afin de déterminer le nombre K ($K < 500$) qui génère le plus de sommes de deux nombres premiers.

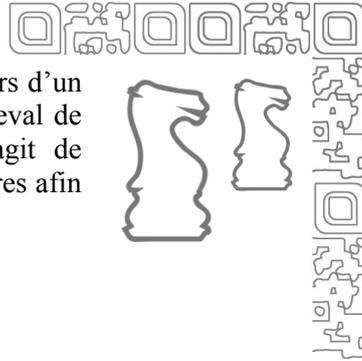
Remarque : le théorème suivant permet de réduire le temps d'exécution de la fonction *estpremier(n)* en ne considérant que les entiers inférieurs à \sqrt{n} .



Les petits chevaux

1- Enoncé

Au jeu des petits chevaux, il faut obtenir un 6 lors d'un lancer de dé équilibré pour pouvoir sortir un cheval de son enclos. A l'aide d'une simulation, il s'agit de déterminer le nombre moyen de lancers nécessaires afin de sortir un cheval de son enclos.



2- Travail demandé

Niveau ☼

Ecrire un algorithme puis un programme permettant d'effectuer la simulation de ce jeu. Le programme doit afficher le nombre N de tentatives qui ont été nécessaires pour sortir un cheval de son enclos. Lancer plusieurs fois le programme.

Niveau ☼☼

Modifier le programme afin d'estimer le nombre moyen de tentatives nécessaires pour sortir un cheval de son enclos. Ce programme doit permettre la saisie du nombre K de simulations.

Niveau ☼☼☼

On note X la variable aléatoire qui associe le nombre de tentatives nécessaires pour sortir un cheval de son enclos. On donne la formule de calcul de l'espérance de X :

$$E(X) = \sum_{k=1}^{\infty} k \times p \times (1-p)^{k-1}$$

avec : $p = \frac{1}{6}$

Ecrire un programme permettant d'afficher une valeur approchée de $E(X)$ en calculant la somme ci-avant pour k allant de 1 à 150. $E(X)$ correspond au nombre moyen de tentatives nécessaires pour sortir un cheval de son enclos.

Chapitre 5. Les instructions et fonctions du langage Scilab

ABS

La fonction `abs()` renvoie la valeur absolue d'un réel ou le module d'un nombre complexe.

Exemple :

```
-->abs(-.35)
ans =
```

```
0.35
```

```
-->abs(3 + %i*4)
ans =
```

```
5.
```

ASCII

La fonction `ascii()` convertit une chaîne de caractères en codes `ascii` ou un vecteur de codes `ascii` en chaîne de caractères. Exemple : `ascii("c")` renvoie 99 et `ascii(99)` renvoie "c".

```
Editeur :
u = "ciel";
v = ascii(u);
disp(v)
w = ascii(v);
disp(w)
```

```
Console :
99.    105.    101.    108.

ciel
```

BAR

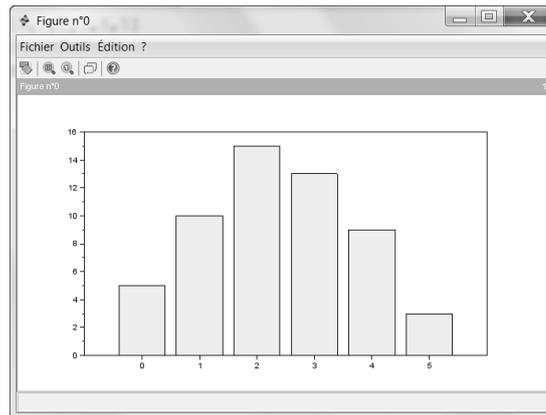
La fonction `bar()` permet l'affichage d'un histogramme à partir d'un couple de vecteurs `X` et `Y`.

```
Editeur :
X = [0,1,2,3,4,5]
Y = [5,10,15,13,9,3]

bar(X,Y,'y')
```

'y' correspond à la couleur de remplissage des barres (yellow).

Résultat dans la fenêtre graphique :



CLF

La fonction `clf()` efface le contenu de la fenêtre graphique courante.

Remarque : On peut saisir dans l'éditeur : `clf` ou `clf()`. Dans le cas de plusieurs fenêtres graphiques, l'instruction `clf(n)` efface le contenu de la fenêtre graphique dont l'identificateur est `n`.

DEFF

La fonction `deff()` permet de définir une fonction utilisateur.

```
deff(' [z] = mamoyenne(x,y)', 'z = (x + y)/2')
```

Les paramètres sont deux chaînes de caractères (que l'on définit à l'aide de l'apostrophe ' ou du guillemet ").

`x, y` : variables d'entrée.

`z` : variable de sortie.

Remarque : Cette fonction est utilisée pour définir une nouvelle fonction depuis la console Scilab. Cependant elle permet également de définir une fonction au cours d'un programme lorsque l'expression de celle-ci doit être saisie par l'utilisateur.

Exemple :

```
Editeur :  
deff(' [z] = mamoyenne(x,y)', 'z = (x + y)/2')
```

```
Console :  
mamoyenne(6,8)  
ans =  
7
```

DISP

La commande disp() permet d'afficher un résultat dans la console Scilab.

```
Editeur :  
a = 5;  
disp(a)
```

```
Console :  
5.
```

EVSTR

Renvoie le résultat de l'évaluation de la matrice de chaînes de caractères. Chaque élément de la matrice doit être une expression Scilab valide.

```
Editeur :  
a = 1;  
b = 2;  
Z = ['a', 'b'];  
mat = evstr(Z);  
disp(mat);
```

```
Console :  
1. 2.
```

EYE

Renvoie une matrice identité (ou matrice unité).

Exemple :

```
-->M = eye(4,4)  
M =
```

```
1. 0. 0. 0.  
0. 1. 0. 0.  
0. 0. 1. 0.  
0. 0. 0. 1.
```

Chapitre 6. Les instructions et fonctions du langage Python

ABS

Renvoie la valeur absolue d'un réel ou le module d'un nombre complexe.

Exemple :

Console :

```
>>> abs(-5)
5
>>> c = complex(3,4)
>>> abs(c)
5.0
```

a%b

a%b renvoie le reste de la division euclidienne de a par b.

Exemple :

Console :

```
>>> 7%2
1
```

Effectivement : $7 = 2*3 + 1$

BAR

La fonction bar() permet d'effectuer le tracé d'un histogramme à partir d'un couple de vecteurs X et Y.

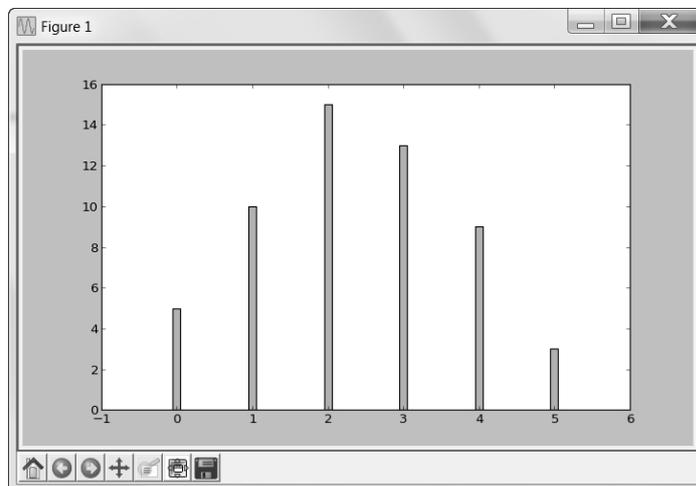
Exemple :

Editeur :

```
import matplotlib.pyplot as plt
```

```
x = [0,1,2,3,4,5]
y = [5,10,15,13,9,3]
```

```
plt.bar(x,y, align='center', width = 0.1, color = 'y')
plt.show()
```



BREAK

L'instruction break permet de sortir d'une boucle.

```
Exemple :
Editeur :
for i in range(1,1000) :
    print(i)
    if i == 3 :
        break
```

```
Console :
>>>
1
2
3
```

CHR

L'instruction chr() renvoie le caractère correspondant au code ascii.

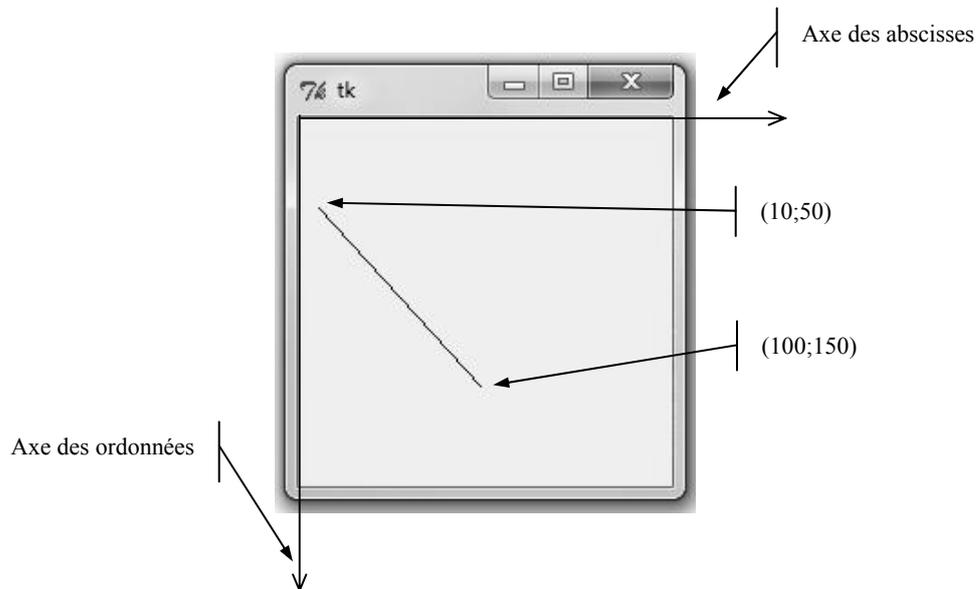
```
Exemple :
Console :
>>> chr(65)
'A'
>>> chr(66)
'B'
```

CREATE_LINE

La fonction create_line(x1,y1,x2,y2) trace le segment de droite [A1A2] avec A1(x1,y1) et A2(x2,y2).

```
Exemple :
Editeur :
from tkinter import *
fen = Tk()
cv = Canvas(fen,width = 200, height = 200)
cv.pack()
cv.create_line(10, 50, 100, 150)
fen.mainloop()
```

Remarque : L'origine du repère se trouve dans le coin supérieur gauche de la fenêtre.



DEF

Déclaration d'une fonction personnalisée.

Exemple :
 Editeur :

```
def f(x) :
    y = x**2 + 1
    return y
```

Par exemple, $f(2)$ renvoie 5.

EVAL

Evalue une expression de type chaîne de caractères.

Exemple :
 Expression incorrecte qui provoque une erreur :
 Console :

```
>>> "15" + 5
Traceback (most recent call last):
  File "<interactive input>", line 1, in <module>
TypeError: Can't convert 'int' object to str implicitly
```

Expression correcte :
 Console :

```
>>> eval("15") + 5
20
```