

Recherche d'empreinte digitale numéro 01

Le but de cette activité est de construire un algorithme pour rechercher une empreinte digitale dans un fichier nationale des empreintes.

On considère qu'une base de données nationale des empreintes digitales est constituée d'une suite de fichiers informatiques contenant l'image des empreintes et le nom du porteur de l'empreinte. Tous les fichiers ont un nom constitué de 0 chiffres puis de l'extension png.

Exemple : 1022547866.png

Pour comparer des empreintes digitales, une idée simple est d'oublier la majorité des détails de celle-ci pour se concentrer sur les points les plus caractéristiques que l'on nomme **minuties**. Il s'agit de bifurcations, d'îles, de nœuds, de lignes qui disparaissent etc ... Chaque chiffre de notre nom de fichier, correspond à une minutie particulière.



Exemple : Si le chiffre 2 correspond à un nœud, on peut dire que l'empreinte de l'exemple 1022547866.png a 2 nœuds. Lorsque nous avons trouvé une empreinte sur une scène d'enquête, il faut chercher le nom de fichier dans la base de données nationale, qui se rapproche le plus de celle trouvée.

On étudiera, en MPS, les deux critères suivants, pour déterminer l'empreinte qui se rapproche le plus.

- Le nombre de chacune des minuties doit être le même.
- Si F est la première empreinte et G la deuxième alors pour tout i dans [1,10] on doit avoir

$$L'écart entre F(i) et G(i) \leq 1$$

Partie I : Saisie de la base de donnée nationale

Avant de pouvoir utiliser un fichier nationale d'empreintes, il va falloir saisir tous les noms de fichier dans une base de donnée qui pour nous sera juste une liste (tableau d'une ligne) ou chaque cellule comportera le nom d'un fichier.

On note une liste L_i et chaque cellule sera nommé $L_i(j)$ où j est l'emplacement de la cellule dans la liste.

Exemple :

L_1	L_2	L_3
$L_1(1)$	$L_2(1)$	$L_3(1)$
$L_1(2)$	$L_2(2)$	$L_3(2)$
$L_1(3)$	$L_2(3)$	$L_3(3)$
$L_1(4)$	$L_2(4)$	$L_3(4)$

On notera **Long** la longueur des tableaux (dans l'exemple $\text{Long}(L_1)=4$)

On considère que la base de données nationale contient les empreintes digitales de 30 personnes.

1. Voici un algorithme permettant de faire la saisie. Ecrire le programme correspondant en langage Ti.

Algorithme SaisieEmpreintes01
 Effacer la liste L_1
 Pour i allant de 1 à 30
 • Demander E
 • $E \rightarrow L_1(i)$
 Fin du pour

Program SaisieE01
 :
 :
 :
 :
 :

2. Réécrire cet algorithme puis le programme correspondant, en utilisant la boucle " Tant que "

Algorithme SaisieEmpreintes02
 Effacer la liste L_2
 $I \leftarrow 1$
 Tant que
 • Demander E
 • $E \rightarrow L_2(i)$
 •
 Fin du pour

Program SaisieE02
 :
 :
 :
 :
 :
 :
 :

Faite fonctionner les deux programmes, avec les empreintes de la dernière page. Attention à bien entrer les empreintes lignes par ligne et de la gauche vers la droite.

Venez montrer le résultat à votre enseignant, pour pouvoir continuer ou pas le TP.

Partie II : Recherche dans la base de donnée nationale

Maintenant que l'on sait saisir une base de données d'empreintes digitales, il faut savoir rechercher l'empreinte la plus proche de celle trouvée sur la scène de crime, avec les deux critères fixés.

1. Création d'un algorithme qui compte le nombre de minutes

Exemple : algorithme qui compte le nombre de 2 dans un nombre de 20 chiffres.

Algorithme CompteChiffre2
 Demander E
 $N \leftarrow E$
 $C \leftarrow 0$
 Pour i allant de 1 à 10
 • Si ($10 \times$ Partie Décimale de $N/10=2$)
 alors $c \leftarrow c+1$
 • $N \leftarrow$ (Partie Entière de $N/10$)
 Fin du pour
 Afficher C

Program Compte2
 :
 :
 :
 :
 :
 :
 :
 :
 :
 :
 :

a) Si E=0215236422
 Compléter le tableau suivant en simulant le fonctionnement de l'algorithme :

i	N	C	10*Partie décimale (N/10)	Partie entière (N/10)
	0215236422	0		
1	021523642	1	2	021523642
2	02152364	2	2	02152364
3	0215236	2	4	021523
4				
5				
6				
7				
8				
9				
10				

- b) Ecrire un programme Ti (Dans le cadre ci-dessus) traduisant cet algorithme et tester son fonctionnement. Les fonctions « partie décimale » et « partie entière » sont dans **Maths / Nbre**
- c) Ecrire un algorithme puis le programme correspondant qui compte le nombre de 0 puis de 1 puis de 2 Dans un nombre de 20 chiffres. On pourra utiliser une Liste L₃ pour stocker les résultats sachant que la cellule L₃(i) doit contenir le nombre de chiffre i dans le nombre entré.

Venez montrer le résultat à votre enseignant, pour pouvoir continuer ou pas le TP.
 Vous pouvez demander l'aide 01 en cas de blocage.

3. Programme avec Algobox qui permet la saisie et la recherche.

- a. Ecriture d'un programme Ti qui permet de réunir toutes les étapes ci-dessus, c'est-à-dire :
 - i. Permettre la saisie des 30 noms de fichier de la base de données nationale
 - ii. Lire le nom du fichier de l'empreinte trouvée sur les lieux du crime
 - iii. Détermine tous les noms des fichiers de la base de données qui vérifient les deux critères demandés
 - iv. Affiche le nom des fichiers qui répondent aux deux critères.

Compléter les programmes pour qu'ils fonctionnent puis les entrer dans votre calculatrice pour les tester avec les données de l'annexe.

Program :Minuties

```

:0→Q
:Effliste (L2,L3)
:For(I,1,10)
:0→L2(I)
:0→L3(I)
:End
:For(K,0,9)
:F→E
:L→G
:For(I,1,10)
:if 10×partDéc (.....)=k
:Then
:L2(K+1)+1 → L2(.....)
:End
:if 10×partDéc (.....)=k
:Then
:L3(K+1)+1 → L3(.....)
:End
:partEnt (E/10)→E
:partEnt (G/10)→G
:End
:End
:For(I,1,10)
:if L2(I)= L3(I)
:Then
:Q+1→Q
:End
:End

```

Program :Ecart

```

:0 → R
:F→E
:L→G
:For(I,1,10)
:if (10×partDéc(E/10)- 10×partDéc(G/10).....1)
et (10×partDéc( E/10)- 10×partDéc(G/10).....-1)
:Then
:R+1→R
:End
:partEnt(.....)→E
:partEnt(.....)→G
:End

```

Program Empreinte

```

:Saisie
:Input "F=",F
:For(I,1,30)
:L1(I)→L
:prgmMinuties
:prgmEcart
:if Q=..... et R=.....
:Then
:Disp "L(",I,")=",L
:Pause
:End
:End

```

Program Saisie

```

: Effliste(L1)
: For(I,1, ..... )
: Input " N= ", N
: ..... → L1(I)
: End

```

Partie III : Et si on change les critères ?

On étudiera les deux critères suivants, pour déterminer l'empreinte qui se rapproche le plus.

- La fréquence de chacune des minuties doit être identique à 0.025 près.
- Il ne doit pas y avoir plus de trois chiffres différents.

1. Ecrire un algorithme qui permet de classer les chiffres, d'un nom de fichier, dans l'ordre croissant.
2. Ecrire un nouveau programme avec Algobox qui permet de prendre en compte ces nouveaux critères.

Annexe

Base de données nationale :

26231472443532192001	13230462447522219103	22231474446532191003	23231472446532191002
13241482436532201092	32233472446512191002	23331472444532191002	23231460447532191002
23231471446532291001	13220462147532294103	32235462447032292101	12230462447532291103
13230962447522241103	22130461549432271203	32230462447532291103	12230462447532291103
32211362457442291103	23231472446532191002	22230462447531293103	12230462447532291103
32230462447532291103	12230462447532294103	12530462447532291103	13320462449522273103
12232462447532291103	12230462445732291103	12230462447539221103	12530462447232291103
12230462417532294103	14230462427532291103		

Empreinte trouvée sur la scène de crime :

12230462447532291103

Program :Minuties

```
:Input "N=",N
:EffListe L3
:For(I,1,10)
:O→L3( ..... )
:End
:For(K,0,9)
:N→E
:For(I,1,20)
:If 10×partDéc (E/10)= .....
:Then
:L3(K+1)+1 → L3(K+1)
:End
:partEnt (E/10)→ .....
:End
:End
:For(K,0,9)
:Disp "NBR de ",K,"=", .....
:Pause
:End
```