

Unité d'Enseignement Mathématiques/Informatique
Année 2020-2021

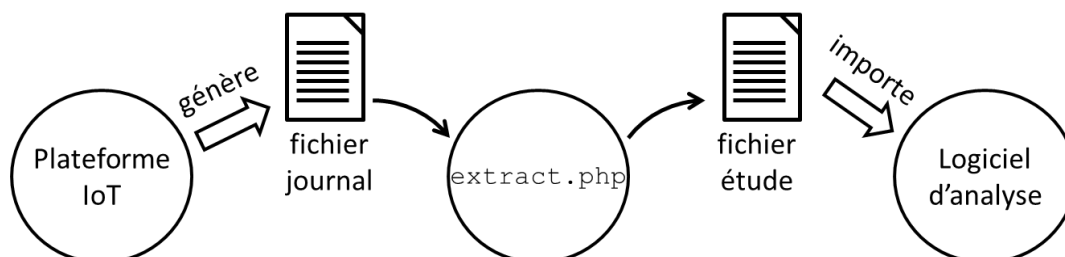
2^e année ESTIA – Apprentis
Examen de Système d'Information – Première session du 7 mai 2021
 (durée 2h00, aucun document autorisé)

-- CORRECTION --

Remarque : Ce document ne présente que des éléments de correction. Les variantes possibles sont nombreuses.

Exercice 2 : Connecteur plateforme Internet of Things (7 points sur 10)

Notre entreprise vend des produits et des garanties à ses clients automobilistes (professionnels et particuliers). Elle utilise une plateforme IoT (*Internet of Things*) hébergée dans le cloud pour gérer les données envoyées par les voitures des clients. Lors de chaque trajet, la plateforme IoT génère des lignes dans des fichiers journaux (*log files*). Ces lignes de données semi-structurées sont difficilement exploitables en l'état. Nous voulons en extraire des informations pertinentes dans le but de pouvoir conduire des analyses de données (*machine learning, business intelligence, aide à la décision, ...*). Pour alimenter ces analyses, nous voulons automatiser l'extraction et la transformation des données vers une forme exploitable par un logiciel d'analyse. Le but de cet exercice est d'écrire le script `extract.php` qui automatisera ce traitement.



Chaque ligne du fichier journal d'un service de données est constituée des éléments suivants :

1. La date de début du trajet, suivie d'un espace, suivi de l'heure ;
2. L'adresse MAC de la voiture connectée concernée ;
3. Le numéro d'identification de l'utilisateur authentifié (*id*) ;
4. Trois valeurs décrites chacune par la clé de la donnée, suivie d'un espace, suivie de la valeur de la donnée :
 - La position géographique de départ (*geo*) ;
 - La position géographique d'arrivée (*geo*) ;
 - La distance kilométrique mesurée par le compteur de la voiture (*kms*).

Sachant que chaque élément sera suivi du caractère point-virgule « ; ».

Exemple de ligne :

```
2019-04-03 07:35:10;f0:de:f1:39:7f:18;1;geo 43.396104 -1.458206;geo 43.447645 -1.555190;kms 15.688;
```

Pour aider notre bureau d'étude et notre service marketing à adapter les produits et les offres, le nouveau Data Scientist recruté par notre entreprise doit mieux comprendre le nombre de

kilomètres parcourus selon les jours de la semaine. L'objectif central des produits et services vendus par notre entreprise restant de contribuer à diminuer les émissions de gaz carbonique, d'oxyde d'azote et de particules fines. Pour conduire cette étude, le logiciel d'analyse devra recevoir (uniquement) le nombre total de kilomètres parcourus pour chaque jour de la semaine sur un mois et une année donnés, en structurant chaque ligne comme suit :

1. Le numéro du jour de la semaine en représentation numérique ISO-8601 (c.-à-d. 1 pour lundi, 2 pour mardi, 3 pour mercredi, 4 pour jeudi, 5 pour vendredi, 6 pour samedi et 7 pour dimanche) ;
2. Le nombre total de kilomètres parcourus par les automobilistes.

Sachant que chaque élément sera suivi du caractère point-virgule « ; ».

Exemple de ligne :

1;3;15.688;

1) Écrivez le script PHP `extract.php` qui lit le fichier `journal.txt` et pour chaque ligne extrait et transforme les données utiles et produit le résultat des calculs pour le mois de mai 2021 dans le fichier `etude.txt`. Ci-après un exemple d'exécution de ce script et un mémento des fonctions PHP utiles pour ce travail.

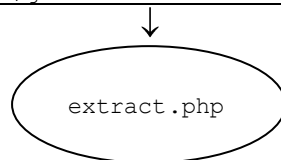
```

extract.php
1  <?php
2
3  $f1 = fopen ('journal.txt', 'r') ;
4
5  if (!$f1) {
6      die ('Erreur ouverture fichier en lecture'."\n") ;
7  }
8
9  $f2 = fopen ('etude.txt', 'w') ;
10
11 if (!$f2) {
12     die ('Erreur ouverture fichier en ecriture'."\n") ;
13 }
14
15 for ($i=1 ; $i <= 7 ; $i++) {
16     $count[$i] = 0 ;
17 }
18
19 $year = 2021 ;
20 $month = 5 ;
21
22 while (!feof($f1)) {
23     $line = trim(fgets ($f1)) ;
24     if ($line != '') {
25         $stab = explode (';', $line) ;
26
27         $date = explode ('-', $stab[0]) ;
28         $y = $date[0] ;
29         $m = $date[1] ;
30
31         if ($y == $year && $m == $month) {
32             $time = strtotime($stab[0]) ;
33             $day = date('N', $time) ;
34             $id = $stab[2] ;
35             $kms = substr($stab[5], 4) ;
36             $count[$day] += $kms ;
37         }
38     }
39 }
40
41 for ($i=1 ; $i <= 7 ; $i++) {
42     fputs ($f2, $i.';'.$count[$i].';'."\r\n") ;
43 }
44
45 fclose ($f1) ;
46 fclose ($f2) ;
47
48 ?>

```

Exemple d'exécution :

journal.txt										
2021-04-30	18:50:46	f0	de:f1:39:7f:18;1	geo	43.396104	-1.458206	geo	43.447645	-1.555190;kms	15.692;
2021-05-03	07:40:46	f0	de:f1:39:7f:18;1	geo	43.396104	-1.458206	geo	43.447645	-1.555190;kms	15.692;
2021-05-03	18:10:10	f0	de:f1:39:7f:18;1	geo	43.447645	-1.555190	geo	43.396104	-1.458206;kms	15.730;
2021-05-04	06:45:00	f0	de:f1:39:7f:19;2	geo	43.486585	-1.555899	geo	43.175398	-1.401778;kms	76.392;
2021-05-04	07:41:32	f0	de:f1:39:7f:18;1	geo	43.396104	-1.458206	geo	43.447645	-1.555190;kms	15.698;
2021-05-04	18:05:30	f0	de:f1:39:7f:18;1	geo	43.447645	-1.555190	geo	43.396104	-1.458206;kms	15.702;
2021-05-04	18:16:00	f0	de:f1:39:7f:19;2	geo	43.175398	-1.401778	geo	43.342216	-1.444841;kms	57.891;
2021-05-04	17:38:00	f0	de:f1:39:7f:19;2	geo	43.342216	-1.444841	geo	43.486585	-1.555899;kms	62.710;
2021-05-05	07:35:10	f0	de:f1:39:7f:18;1	geo	43.396104	-1.458206	geo	43.447645	-1.555190;kms	15.688;
2021-05-05	18:15:03	f0	de:f1:39:7f:18;1	geo	43.447645	-1.555190	geo	43.394214	-1.457210;kms	15.772;
2021-05-05	17:45:03	f0	de:f1:39:7f:18;1	geo	43.394214	-1.457210	geo	43.396104	-1.458206;kms	0.268;



etude.txt	
1;	31.422;
2;	228.393;
3;	31.728;
4;	0;
5;	0;
6;	0;
7;	0;

MÉMENTO : Voici quelques éléments de PHP pour vous aider à mener à bien ce travail

- La fonction **fopen** permet d'obtenir un descripteur (type `resource`) d'un fichier `$filepath` ouvert en lecture quand `$mode` vaut "r" et en écriture quand `$mode` vaut "w". La valeur retournée par `fopen` est le descripteur qui a été ouvert, ou faux si l'ouverture a échoué (par exemple en cas de fichier inexistant).

```
resource fopen ( string $filepath , string $mode ) ;
```

- La fonction **fclose** permet de libérer le descripteur de fichier `$fid`.
- La fonction **feof** permet de tester si le descripteur de fichier `$fid` a atteint la fin du fichier. La valeur retournée est vrai si la fin de fichier a été atteinte, faux sinon.

```
bool fclose ( resource $fid ) ;
```

```
bool feof ( resource $fid ) ;
```

- La fonction **fgets** permet de lire une ligne dans le fichier décrit par le descripteur `$fid`. La valeur retournée est une chaîne de caractères contenant la ligne du fichier.

```
string fgets ( resource $fid ) ;
```

- La fonction **fputs** permet d'écrire dans le fichier décrit par le descripteur `$fid`.

```
void fputs ( resource $fid , string $txt ) ;
```

- La fonction **explode** permet de décomposer une chaîne de caractères en plusieurs morceaux dans un tableau selon un délimiteur spécifié.

```
array explode ( string $delimiter, string $line ) ;
```

- La fonction **strtotime** permet calculer le « *timestamp* » d'une date (c.-à-d. le nombre de secondes écoulées entre le 1^{er} janvier 1970 à 00h00.00 UTC et la date).

```
int strtotime ( string $date ) ;
```

Par exemple, après l'appel suivant :

```
$t = strtotime ( "2018-04-12 10:45:58" ) ;
```

La variable `$t` contiendra 1523522758

- La fonction **date** permet de calculer la représentation d'une date à partir d'une valeur de *timestamp* et du format souhaité.

```
string date ( string $format , string $timestamp ) ;
```

Par exemple, après l'appel suivant :

```
$d1 = date ( "h:i a, l, F jS, Y" , 1523522758 ) ;
```

La variable `$d1` contiendra "10:45 am, Thursday, April 12th, 2018"

Et, par exemple, en ce qui nous concerne plus particulièrement, après l'appel suivant :

```
$d2 = date ( "N" , 1523522758 ) ;
```

La variable `$d2` contiendra "4", car le formatage "N" demande de produire le jour numéro du jour de la semaine en représentation numérique ISO-8601 (c.-à-d. de 1 pour lundi à 7 pour dimanche).

- La fonction **substr** permet de créer une chaîne de caractères de longueur `$length` à partir de la position `$start` de la chaîne de caractères `$line`. Si le paramètre `$length` est omis, alors le résultat sera une chaîne de caractères à partir de la position `$start` jusqu'à la fin de `$line`.

```
string substr ( string $line, int $start, int $length ) ;
```