G. Hunault
L3 Informatique

Angers, mars 2019

Examen en Production Automatique de Graphiques, Statistiques et Documents

Question 1 : description de matrices colorisées

On appelle MC – *matrice colorisée* – une matrice (au sens mathématique du terme) d'entiers positifs ou nuls, c'est-à-dire un tableau à deux dimensions, nommées classiquement lignes et colonnes. Pour de telles matrices, on invente le langage SMC suivant à base d'instructions et de commentaires défini comme suit :

- une instruction commence par l'un des mots matrice, dim ou case ;
- il ne peut y avoir qu'une seule instruction matrice, il ne peut y avoir qu'une seule instruction dim mais il peut y avoir autant d'instructions case que l'on veut ;
- toute instruction s'écrit sur une seule ligne et on peut avoir des lignes vides ;
- un commentaire commence par le symbole dièse (#) ; il peut être mis soit seul sur une ligne soit à la suite d'une instruction ;
- matrice est la première instruction ; elle est suivie d'un seul mot, le nom de la matrice ;
- dim est la deuxième instruction ; elle est suivie de deux entiers qui sont respectivement le nombre de lignes et le nombre de colonnes de la matrice ;
- toute instruction case doit être suivie de trois entiers, un numéro de ligne, un numéro de colonne et un numéro de couleur.

Pour ce qui suit, on admet que les couleurs sont définies par les codes suivants : 0=blanc, 1=noir, 2=bleu, 3=rouge.

On admettra aussi que par défaut toutes les cases ont la couleur blanche.

Question 1.1

Sachant que la MC ci-dessous

exemple	C001	C002	C003
L001			
L002			
L003	■		
L004	■		
L005		■	

correspond au fichier .smc suivant

```
matrice exemple

# question 1.1 de l'examen

dim 5 3
case 3 1 1 # noir
case 4 1 1
case 5 2 1
```

donner le contenu du fichier france.smc qui permet de définir le drapeau français, comme la matrice MC de nom FR, avec 1 ligne et 3 colonnes, dont la case en ligne 1, colonne 1 est bleue et dont la case en ligne 1, colonne 3 est rouge.

Question 1.2

Donner le code complet en PHP conceptuel, d'une fonction PHP nommée smc qui admet un seul paramètre nommé $smcFile avec la chaine vide comme valeur par défaut et dont le comportement est le suivant, avec des fonctions verifieMC(), construitMC() et afficheMC() supposées déjà définies :

- si le paramètre est la chaine vide, on écrit un titre de niveau 2 pour le dire et on renvoie 1 ;

- si le fichier qui correspond au paramètre n'existe pas, on le dit dans un titre de niveau 2 et on renvoie 1 ;

- si l'appel de la fonction verifieMC($smcFile) renvoie autre chose que 0, on dit que le fichier est invalide dans un titre de niveau 2 et on renvoie 1 ;

- sinon :
  - on affiche un titre de niveau 1 qui rappelle le nom du fichier passé en paramètre ;
  - on appelle la fonction construitMC() avec le nom du fichier passé en paramètre et on met dans la variable $mc le résultat de cette fonction ;
  - on appelle la fonction afficheMC() en lui passant $mc comme paramètre ;
  - on renvoie 0.

Question 2 : création d'une matrice MC

Donner le code complet de la fonction construitMC() qui admet comme seul paramètre le nom d'un fichier SMC valide. Cette fonction doit créer et renvoyer un tableau au sens de PHP (array) qui comporte $nbl + 1$ lignes où nbl est le premier entier qui suit le mot dim dans le fichier SMC et $nbc + 1$ colonnes où nbc est le deuxième entier qui suit le mot dim. La ligne 0 contient le nom de la matrice en colonne 0 puis les numéros de colonne formatés comme dans la copie d'écran. Pour les autres lignes, la colonne 0 contient les numéros de ligne formatés comme dans la copie d'écran.

La matrice doit être initalisée avec la valeur 0 partout puis doit être éventuellement mise à jour avec les valeurs entières lues dans les instructions case.

Vous discuterez le choix de la lecture du fichier en PHP avec l'instruction file, l'instruction file_get_contents ou avec les instructions fopen, fgets, fclose.

Vous pouvez implémenter la matrice comme un tableau de lignes ou comme un tableau de colonnes.

Vous pouvez inventer des fonctions complémentaires pour que le code soit plus lisible.

Question 3 : affichage d'une matrice MC

Donnez le code complet de la fonction afficheMC() qui admet pour seul paramètre un tableau comme celui renvoyé par la fonction construitMC() et qui fournit un rendu XHTML de la matrice via un élément table. On supposera que chaque ligne (tr) et chaque case (th ou td) utilise toujours au moins une classe CSS nommée bord-gris. On supposera de plus qu'il y a un tableau de correspondance entre numéro de couleur et classe CSS, comme par exemple fond_blanc pour la valeur 0, fond_noir pour la valeur 1...

Vous utiliserez forcément du PHP conceptuel.

Question 4 : améliorations du langage SMC

Essayer d'inventer un maximum d'autres instructions pour enrichir le langage SMC afin de réduire le nombre d'instructions à écrire pour une matrice donnée. On trouvera en annexe quelques matrices MC à produire en un minimum d'instructions.

Vous détaillerez la syntaxe de chaque nouvelle instruction.

Question 5 : Discussion

Répondez à la question suivante :

> *Avec toutes les possibilités qu'offre SVG, le langage DOT va-t-il disparaitre ?*

Votre réponse devra mettre en évidence votre culture sur la qualité des documents produits par programme ainsi que votre recul et votre esprit de synthèse en matière de traitement de l'information et de programmation.

Votre réponse devra faire 10 lignes au minimum, sans limite de maximum. On utilisera au moins 3 mots de 4 syllabes ou plus.

ANNEXE

xmp04	C001	C002	C003	C004	C005	C006	C007	C008
L001	■	■	■	■	■	■	■	■
L002	■	■	■	■	■	■	■	■
L003								
L004								
L005								
L006								
L007								
L008								

xmp05	C001	C002	C003	C004	C005	C006	C007	C008
L001						■		■
L002						■		■
L003						■		■
L004						■		■
L005						■		■
L006						■		■
L007						■		■
L008						■		■

xmp06	C001	C002	C003	C004	C005	C006	C007	C008
L001	■	■	■					
L002	■	■	■					
L003	■	■	■					
L004								
L005								
L006								
L007								
L008								

xmp07	C001	C002	C003	C004	C005	C006	C007	C008
L001								
L002								
L003								
L004								
L005								
L006						■	■	
L007						■	■	
L008								

xmp08	C001	C002	C003	C004	C005	C006	C007	C008
L001								
L002								
L003								
L004								
L005								
L006	■						■	■
L007	■						■	■
L008								

xmp09	C001	C002	C003	C004	C005	C006	C007	C008
L001	■	■	■	■	■	■	■	■
L002	■	■	■	■	■	■	■	■
L003	■	■	■	■	■	■	■	■
L004	■	■	■		■	■	■	■
L005	■	■	■	■	■	■	■	■
L006	■	■	■	■	■	■	■	■
L007	■	■	■	■	■	■	■	■
L008	■	■	■	■	■	■	■	■

xmp10	C001	C002	C003	C004	C005	C006	C007	C008
L001	■							
L002		■						
L003			■					
L004				■				
L005					■			
L006						■		
L007							■	
L008								■

xmp11	C001	C002	C003	C004	C005	C006	C007	C008
L001	■							
L002	■	■						
L003	■	■	■					
L004	■	■	■	■				
L005	■	■	■	■	■			
L006	■	■	■	■	■	■		
L007	■	■	■	■	■	■	■	
L008	■	■	■	■	■	■	■	■

ESQUISSE DE SOLUTION

Question 1 : description de matrices colorisées

Question 1.1

```
matrice drapeauFR
# un exemple simple
dim 1 3
case 1 1 2 # bleu
case 1 3 3 # rouge
```

Question 1.2

Dans la mesure où toutes les actions à exécuter dans la fonction smc() sont indiquées, le code PHP est simple à écrire et correspond à ce que l'on a déjà vu en cours. Mettre return(1) permet de ne pas avoir de if en cascade.

Toutes les fonctions sont définies dans le fichier des sous-programmes nommé `smc_inc.php`.

```
<?php

include("std.php") ;
include("smc_inc.php") ;

############################################################

function smc($smcFile="") {

############################################################

if ($smcFile=="") {
  h2("No .smc file given. Stop.","grouge") ;
  return(1) ;
} # fin si
```

```
if (!file_exists($smcFile)) {
  h2("File $smcFile not found. Stop.","grouge") ;
  return(1) ;
} # fin si

$vmc = verifieMC($smcFile) ;
if (!$vmc==0) {
  h2("Error in the SMC file $smcFile. Stop.","grouge") ;
  return(1) ;
} # fin si

h1("Colored matrix for file $smcFile") ;

$mc = construitMC($smcFile) ;
afficheMC($mc) ;

return(0) ;

} # fin de fonction smc

?>
```

Question 2 : création d'une matrice MC

Avec un peu de réflexion, les instructions SMC sont gérables via le concept de mot.

Comme le fichier `strfun.php` utilisé par `std.php` définit une fonction nommée **mot** et dont la syntaxe est **mot(phrase,numéroDeMot)**, la construction de la matrice consiste sans doute à appeler pour chaque mot numéro 1 (instruction) un sous-programme paramétré dont les paramètres sont les mots numéros 2, 3...

Pour le format SMC fourni, deux fonctions, nommées initMatrice() et coloriseCase() sont sans doute suffisantes.

Le code fourni ci-dessous utilise d'autres fonctions, liées aux améliorations possibles. On notera que l'ordre dans lequel on traite les instructions est important.

On lit le fichier avec file, ce qui permet un traitement ligne par ligne. On utilise l'instruction continue dès qu'on a traité l'instruction SMC afin de passer à l'itération suivante.

```
<?php

############################################################

function construitMC($smcFile) {

############################################################

$smc = file($smcFile) ; # car on a déjà testé la présence dans smc($smcFile)

$nbi = 0 ;  # nombre d'instructions
$nom = "" ; # nom de la matrice

foreach ($smc as $ins) { # on traite chacun des lignes obtenues grâce à file(...)

  $nbi++ ;
  $ins = trim($ins) ;
  # on ignore les lignes vides et les lignes commentaires
  if ($ins=="") { continue ; } ;
  if (substr($ins,0,1)=="#") { continue ; } ;
  $m1 = mot($ins,1) ;
  $m2 = mot($ins,2) ;
  $m3 = mot($ins,3) ;
  $m4 = mot($ins,4) ;
  $m5 = mot($ins,5) ;
  $m6 = mot($ins,6) ;

# instructions du format SMC fourni

  if ($m1=="matrice")   { $nom= $m2 ; continue ; } ;
  if ($m1=="dim")       {
      $m = initMatrice($nom,$m2,$m3) ; $nbl = $m2 ; $nbc = $m3 ; continue ; } ;
  if ($m1=="case")      {
      $m = coloriseCase($m,$m2,$m3,$m4) ; continue ; } ;

# améliorations

  if ($m1=="couleur")   {  $m = coloriseMatrice($m,$m2) ; continue ; } ;
  if ($m1=="triangle")  {  $m = coloriseTriangle($m,$m2,$m3) ; continue ; } ;
  if ($m1=="diagonale") {  $m = coloriseDiagonale($m,$m2) ; continue ; } ;
  if ($m1=="ligne")     {  $m = coloriseLigne($m,$m2,$m3) ; continue ; } ;
  if ($m1=="colonne")   {  $m = coloriseColonne($m,$m2,$m3) ; continue ; } ;
  if ($m1=="bloc")      {  $m = coloriseBloc($m,$m2,$m3,$m4,$m5,$m6) ; continue ; } ;
```

```
  p() ;
  echo "ligne $nbi mot $m1 non reconnu" ;
  finp() ;

} # fin pour chaque

return($m) ;

} # fin de fonction construitMC

?>
```

Question 3 : affichage d'une matrice MC

L'énoncé guide beaucoup l'écriture de la solution.

Compte tenu de HTML, on génère l'élément `<table>` ligne par ligne, en séparant bien la ligne zéro et la colonne zéro.

Puisque les styles CSS sont déjà définis, on initialise chaque case du tableau avec le style à *bord-gris* et on rajoute le style associé à la couleur.

```
<?php

############################################################

function afficheMC($m) {

############################################################

$nbl = count($m)    - 1 ;
$nbc = count($m[0]) - 1 ;

table(1,10,"collapse") ;

 # la ligne zéro correspond aux entêtes de colonnes

 tr("bord-gris") ;
    th("","bord-gris") ; echo $m[0][0] ; finth() ;
    for ($idc=1;$idc<=$nbc;$idc++) {
      th("","bord-gris") ; echo $m[0][$idc] ; finth() ;
    } ; # fin pour idc
 fintr() ;
```

```
for ($idl=1;$idl<=$nbl;$idl++) {

    tr("bord-gris") ;
    td("L","bord-gris") ;
    echo $m[$idl][0] ;
    fintd() ;
    for ($idc=1;$idc<=$nbc;$idc++) {
         $fond = "bord-gris " ;
         if ($m[$idl][$idc]==0) { $fond .= "fond_blanc" ;  } ;
         if ($m[$idl][$idc]==1) { $fond .= "fond_noir" ;  } ;
         if ($m[$idl][$idc]==2) { $fond .= "fond_bleu" ;  } ;
         if ($m[$idl][$idc]==3) { $fond .= "fond_rouge" ;  } ;
         td("C",$fond) ; nbsp() ; fintd() ;
    } ; # fin pour idc
    fintr() ;

} ; # fin pour idl

fintable() ;

} # fin de fonction afficheMC

?>
```

Sous-programmes utilisés

Afin que la solution soit complète, nous fournissons ci-dessous le code des autres fonctions utilisées, mais sans explication.

```
<?php

## SOUS-PROGRAMMES POUR LE FORMAT SMC

#############################################################

function coloriseCase($m,$lig,$col,$coul) {

#############################################################

   $m[$lig][$col] = $coul ;
   return($m) ;

} # fin de fonction coloriseCase
```

```
############################################################

function initMatrice($nom,$nbl,$nbc) {

############################################################

$m= array() ;

for ($idl=1;$idl<=$nbl;$idl++) {
  $m[$idl] = array( range(0,$nbc ) ) ;
    for ($idc=0;$idc<=$nbc;$idc++) {
      $valeur = 0 ;
      if ($idc==0) { $valeur = "L".sprintf("%03d",$idl) ; } ;
      $m[$idl][$idc] = $valeur ;
    } ; # fin pour idc
} ; # fin pour idl

for ($idc=1;$idc<=$nbc;$idc++) {
  $m[0][$idc] = "C".sprintf("%03d",$idc) ;
} ; # fin pour idc

$m[0][0] = $nom ;

return($m) ;

} # fin de fonction initMatrice


############################################################

function coloriseLigne($m,$lig,$coul) {

############################################################

$nbl = count($m)    - 1 ;
$nbc = count($m[0]) - 1 ;
for ($idc=1;$idc<=$nbc;$idc++) {
   $m = coloriseCase($m,$lig,$idc,$coul) ;
} ; # fin pour idc

return($m) ;

} # fin de fonction coloriseLigne

############################################################

function coloriseColonne($m,$col,$coul) {

############################################################
```

```
$nbl = count($m)    - 1 ;
$nbc = count($m[0]) - 1 ;
for ($idl=1;$idl<=$nbl;$idl++) {
   $m = coloriseCase($m,$idl,$col,$coul) ;
} ; # fin pour idl

return($m) ;

} # fin de fonction coloriseColonne

############################################################

function coloriseBloc($m,$ligDeb,$colDeb,$ligFin,$colFin,$coul) {

############################################################

# pour debug :
#  p() ; echo "coloriseBloc(m,$ligDeb,$colDeb,$ligFin,$colFin,$coul\n"; finp() ;

for ($idl=$ligDeb;$idl<=$ligFin;$idl++) {
   for ($idc=$colDeb;$idc<=$colFin;$idc++) {
     $m = coloriseCase($m,$idl,$idc,$coul) ;
   } ; # fin pour idc
} ; # fin pour idl

return($m) ;

} # fin de fonction coloriseBloc

############################################################

function coloriseMatrice($m,$coul) {

############################################################

$nbl = count($m)    - 1 ;
$nbc = count($m[0]) - 1 ;

for ($idl=1;$idl<=$nbl;$idl++) {
   for ($idc=1;$idc<=$nbc;$idc++) {
     $m = coloriseCase($m,$idl,$idc,$coul) ;
   } ; # fin pour idc
} ; # fin pour idl

return($m) ;

} # fin de fonction coloriseMatrice
```

```
############################################################

function coloriseDiagonale($m,$coul) {

############################################################

$nbl = count($m)    - 1 ;
$nbc = count($m[0]) - 1 ;
#p() ; echo "coloriseBloc(m,$ligDeb,$colDeb,$ligFin,$colFin,$coul\n"; finp() ;
for ($idv=1;$idv<=min($nbl,$nbc);$idv++) {
     $m = coloriseCase($m,$idv,$idv,$coul) ;
} ; # fin pour idl
return($m) ;

} # fin de fonction coloriseDiagonale

############################################################

function coloriseTriangle($m,$mode,$coul) {

############################################################

$nbl = count($m)    - 1 ;
$nbc = count($m[0]) - 1 ;
#p() ; echo "coloriseBloc(m,$ligDeb,$colDeb,$ligFin,$colFin,$coul\n"; finp() ;
for ($idl=1;$idl<=min($nbl,$nbc);$idl++) {
   for ($idc=1;$idc<=min($nbl,$nbc);$idc++) {
     if ($mode=="inférieur") { if ($idl>=$idc) { $m = coloriseCase($m,$idl,$idc,$coul) ; }
     if ($mode=="supérieur") { if ($idl<=$idc) { $m = coloriseCase($m,$idl,$idc,$coul) ; }
   } ; # fin pour idl
} ; # fin pour idl
return($m) ;

} # fin de fonction coloriseTriangle

?>
```

Question 4 : améliorations du langage SMC

Voici des idées d'amélioration. NCOUL désigne un numéro de couleur, NLIG désigne un numéro de ligne, NCOL désigne un numéro de colonne.

- couleur NCOUL

  on colorise toute la matrice avec la couleur indiquée.

- triangle MODE NCOUL

  si MODE est INF, on remplit la matrice triangulaire inférieure avec la couleur indiquée, si MODE est SUP, on remplit la matrice triangulaire supérieure avec la couleur indiquée.

- diagonale NCOUL

  on colorise la diagonale de la matrice avec la couleur indiquée.

- ligne NLIG NCOUL

  on colorise la ligne correspondante de la matrice avec la couleur indiquée.

- colonne NCOL NCOUL

  on colorise la colonne correspondante de la matrice avec la couleur indiquée.

- bloc LIGDEB LIGFIN COLDEB COLFIN NCOUL

  on colorise la sous-matrice correspondante de la matrice avec la couleur indiquée.