



POLYTECHNIQUE
MONTRÉAL

Questionnaire
examen final +
corrigé

INF1005A

Sigle du cours

Identification de l'étudiant(e)		
Nom :	Prénom :	
Signature :	Matricule :	Groupe :

Sigle et titre du cours		Groupe	Trimestre
INF1005A – Programmation procédurale		Tous	20161
Professeur		Local	Téléphone
Martine Bellaïche, responsable Valérie Danielle Justafort, Salah-Eddine Benbrahim, Marième Diallo et Luiza Gheorghe, chargés de cours		M-3414	4679 / 5193
Jour	Date	Durée	Heures
Mercredi	20 avril 2016	2 h 30	9 h 30 à 12 h 00

Documentation	Calculatrice	
<input checked="" type="checkbox"/> Aucune <input type="checkbox"/> Toute <input checked="" type="checkbox"/> Voir directives particulières	<input checked="" type="checkbox"/> Aucune <input type="checkbox"/> Toutes <input type="checkbox"/> Non programmable	Les cellulaires, agendas électroniques ou téléavertisseurs sont interdits.

Directives particulières	
<ul style="list-style-type: none">✓ Ne recopiez pas les déclarations ni les instructions déjà fournies dans le questionnaire.✓ Vous n'avez pas à écrire de commentaires ni d'en-têtes.✓ Au besoin, utilisez le verso de chaque page pour vos calculs ou si vous manquez d'espace pour vos réponses.✓ Le code écrit doit être complet, c'est-à-dire qu'il est interdit d'utiliser les points de suspension (...) ou une phrase comme « Cette portion de code est répétée x fois ».✓ Ne faites que les validations d'entrées demandées.✓ L'utilisation de fclose('all') et feof() est interdite.	

Important	Cet examen contient <input type="text" value="4"/> questions sur un total de <input type="text" value="20"/> pages (excluant cette page)
	La pondération de cet examen est de <input type="text" value="40"/> %
	Vous devez répondre sur : <input checked="" type="checkbox"/> le questionnaire <input type="checkbox"/> le cahier <input type="checkbox"/> les deux
	Vous devez remettre le questionnaire : <input checked="" type="checkbox"/> oui <input type="checkbox"/> non

L'étudiant doit honorer l'engagement pris lors de la signature du code de conduite.

Fonctions interdites

`bin2dec()`, `break()`, `cell2mat()`, `continue()`, `dec2bin()`, `exist()`,
`exit()`, `feof()`, `max()`, `numel()`, `quit()`, `return()`, `sort()`, `strjoin()`,
`sscanf()`, `strsplit()`, `struct`, `sum()`, `textscan()`, `textscan()`,
`textread()`, `true()`, `unique()`.

Aussi interdite `fclose('all')`

Question 1

(2 points)

Encercler la bonne réponse qui s'applique à chaque énoncé.

1.1 Le nombre $(110)_{10}$ est

- a) $(0110\ 0110)_2$
- b) $(106)_8$
- c) $(6E)_{16}$
- d) $(167)_{10}$

1.2 Soit le nombre réel ci-dessous représenté en binaire en simple précision en respectant la norme IEEE754 :

1	1000 0001	1100 0000 0000 0000 0000 000
---	-----------	------------------------------

- a) Le nombre dans la base 10 est un nombre négatif.
- b) Le nombre dans la base 10 est un nombre positif.
- c) La partie décimale du nombre dans la base 10 est 0.0005.
- d) Le nombre décimal est 7.

1.3 Soient les nombres binaires `int8` : $(0100\ 0111)_2$ et $(0000\ 0000)_2$, après l'addition de ces deux nombres, on obtient :

- a) Un débordement.
- b) Un résultat correct de l'addition.
- c) Une retenue.
- d) La réponse est un nombre négatif.

1.4 Le nombre $(-256)_{10}$ peut être représenté en MATLAB en utilisant :

- a) `int8`
- b) `uint8`
- c) `uint16`
- d) `int16`

Solution (2 pt) ->

1 – c
 2 – a
 3 – b
 5 – d

Question 2**(2 points)**

Soient les données suivantes :

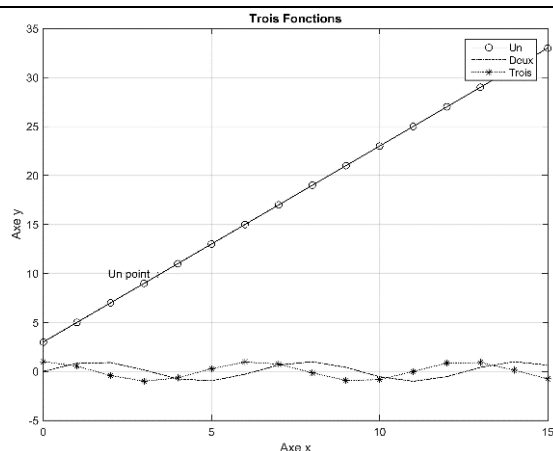
```
x=0:1:15;
y1=2*x+3;
y2=sin(x);
y3=cos(x);
```

Pour chaque numéro (correspondant à une partie de code), choisir la bonne lettre qui correspond au graphique tracé avec le code.

1)

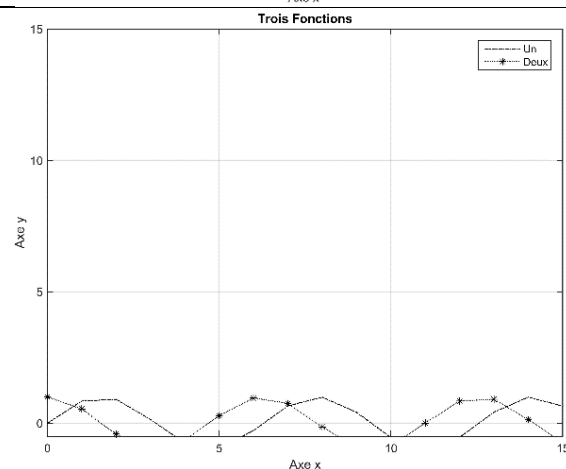
```
plot(x, y1, 'o-', x, y2, '-.',
x, y3, ':*')

xlabel('Axe x');
ylabel('Axe y');
title('Trois Fonctions');
legend('Un', 'Deux', 'Trois',
1);
```

**a)****2)**

```
plot(x, y1, 'o-')
plot(x, y2, '-.')
hold on
plot(x, y3, ':*')
hold off
grid on

axis ([-inf inf -0.5 15]);
xlabel('Axe x');
ylabel('Axe y');
title('Trois Fonctions');
legend('Un', 'Deux', 'Trois',
1);
```

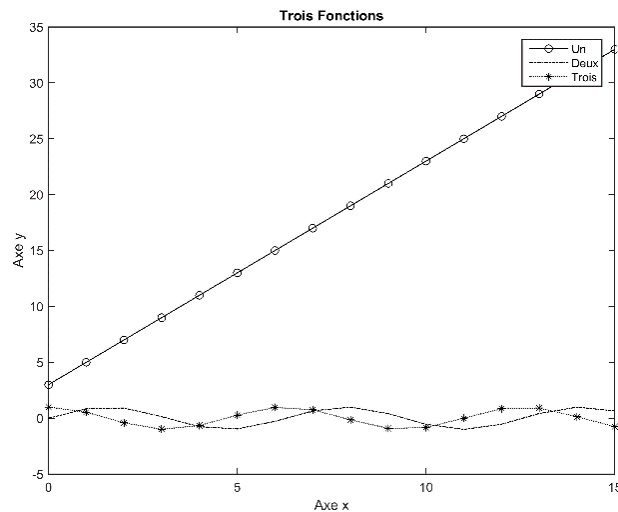
**b)**

3)

```
plot(x, y1, 'o-', x, y2, '-.',
x, y3, ':*')

axis ([-1 inf -0.5 15]);
xlabel('Axe x');
ylabel('Axe y');
title('Trois Fonctions');
legend('Un', 'Deux', 'Trois',
1);
```

c)

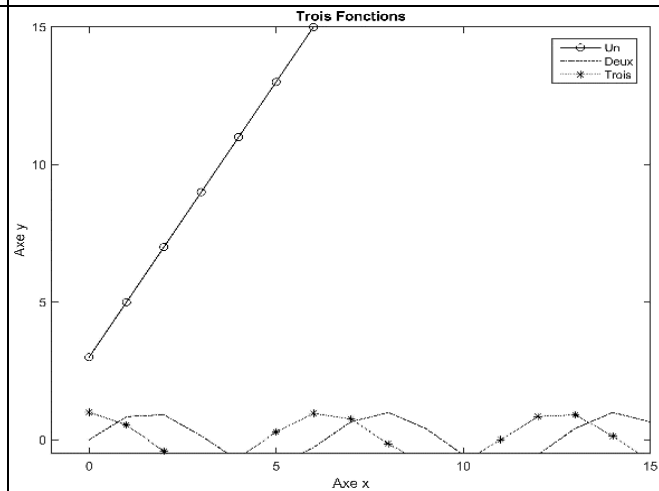


4)

```
plot(x, y1, 'o-', x, y2, '-.',
x, y3, ':*')
grid on;

xlabel('Axe x');
ylabel('Axe y');
title('Trois Fonctions');
legend('Un', 'Deux', 'Trois',
1);
text(3.5, 10, 'Un
point\rightarrow',
'HorizontalAlignment','right')
```

d)



Code	Graphique
1	
2	
3	
4	

Solution question 2 : (2 pt):

1-c

2-b

3-d

4-a

Question 3**(3 points)**

Le fichier binaire `etud.bin` contient des informations sur plusieurs étudiants et a la structure suivante :

Taille nom étudiant_1	Nom étudiant_1	Matricule étudiant_1	Moyenne étudiant_1	. . .	Taille nom étudiant_n	Nom étudiant_n	Matricule étudiant_n	Moyenne étudiant_n
-----------------------------	-------------------	-------------------------	-----------------------	----------	-----------------------------	-------------------	-------------------------	-----------------------

où

`Taille nom étudiant` : un entier - le nombre de caractères du nom d'un étudiant.

`Nom étudiant` : une chaîne de caractères - le nom de l'étudiant.

`Matricule` : un entier très grand – le matricule de l'étudiant.

`Moyenne` : un nombre réel – la moyenne de l'étudiant.

Le code ci-dessous lit et sauvegarde dans l'ensemble de cellules `etudiants`, les informations sur les étudiants. Malheureusement, le code contient 6 erreurs. Identifier les lignes et les erreurs des lignes. Écrire les bonnes instructions des lignes erronées.

```

1  idl = fopen('etud.bin', 'w');
2  if idl == -1
3      disp('erreur ouverture')
4  else
5      taille = fread(idl, 1, 'uint8');
6      i=1;
7      while ~ischar(taille)
8          etudiants{i, 1} = fread(idl, 1, 'char');
9          etudiants{i, 2} = fread(idl, [1 7], 'int32');
10         etudiants{i, 3} = fread(idl, 1, 'float32');
11         i=i+1;
12         taille = fread(idl, 1, 'uint8');
13     end
14     fclose('etud.bin');
15 end

```

Question 3 (suite)

Solution question 3 (3 pt) ->

Lignes: 1, 7, 8, 8, 9, 14

```
1  idl = fopen('etud.bin', 'w'); %doit écrire 'r'
2  if idl == -1
3      disp('erreur ouverture')
4  else
5      taille = fread(idl, 1, 'uint8');
6      i=1;
7      while ~ischar(taille) % doit écrire isempty() au lieu de
ischar
8          etudiants{i, 1} = fread(idl, 1, '*char'); %doit écrire
[1, taille] ou taille et transposée aussi il manque l'étoile pour
ASCII
9          etudiants{i, 2} = fread(idl, [1, 7], 'int32'); %doit
écrire 1
10         etudiants{i, 3} = fread(idl, 1, 'float32');
11         i=i+1;
12         taille = fread(idl, 1, 'uint8');
13     end
14     fclose('etud.bin');%doit écrire identificateur du fichier idl
15 end
```

Question 4**(13 points)**

La ligue interuniversitaire de Hockey Cosom désire utiliser un système automatique afin d'afficher la programmation des quarts de finale, entre les 8 meilleures équipes de chaque université et de chaque catégorie comme suit :

Tournoi de Hockey Cosom - Quarts de finale

```
1- Poly
*****
Bac                Cycles Supérieurs

F vs E             M vs U
C vs H             I vs P
A vs B             J vs O
D vs G             K vs Q
```

```
2- Concordia
*****
```

Bac Cycles Supérieurs

...

Vous avez été sollicité(e) pour réaliser ce travail et le client vous fournit, pour chaque université, le nom, l'établissement et un tableau d'enregistrements regroupant les informations sur les différentes équipes (bac et cycles supérieurs) contenant les champs suivants :

- `nom` : nom de l'équipe (une chaîne de caractères) ;
- `cycle` : cycle d'études auquel appartiennent les membres de l'équipe (un caractère : 'B' pour bac, 'M' pour maîtrise et 'D' pour doctorat) ;
- `parties` : vecteur-ligne de 3 entiers [G N P] (G, N et P étant respectivement le nombre de parties gagnées, nulles ou perdues tout au long de la saison régulière).

Le client vous explique aussi que plusieurs fonctions ont déjà été définies et qu'il s'agit pour vous de compléter le travail.

4.1 Soit la fonction `split()` qui sépare les informations des équipes du bac de celles des équipes aux cycles supérieurs.

Entrée	Sorties
Un tableau d'enregistrements, equipes , contenant les champs : nom , cycle et parties , relatives à toutes les équipes d'une université donnée.	Deux tableaux d'enregistrements, Bac , Sup , chacun avec 2 champs : nom et parties . Bac : informations sur les équipes du bac. Sup : informations sur les équipes aux cycles supérieurs.

Question 4 (suite)

4.1.1 Donner le prototype de la fonction `split()`.**Solution 4.1.1**

```
function [Bac, Sup] = split(equipes)
```

4.1.2 Compléter le code suivant afin de retourner, dans les enregistrements appropriés, les informations pertinentes, et vérifier le nombre d'entrées et de sorties.

Prototype...

```
Bac.nom = ''; Sup.nom = '';
%Nombre d'enregistrements dans
Bac et Sup
BacCount = 0; SupCount = 0;
```

```
if À compléter 1
    error('Nombre d''arguments
incorrects!');
end
```

```
if isempty(equipes)
    error('Aucune equipe...');
end
```

```
for i = 1:length(equipes)
```

... **À compléter 2...**

```
end
```

Solution 4.1.2

```
1) nargin ~= 1 || nargin ~= 2
```

2)

```
if equipes(i).cycle == 'B'
    BacCount = BacCount + 1;
    Bac(BacCount).nom = equipes(i).nom;
    Bac(end).parties = equipes(i).parties;
else
    SupCount = SupCount + 1;
    Sup(SupCount).nom = equipes(i).nom;
    Sup(end).parties = equipes(i).parties;
end
```

Question 4 (suite)

4.2 Soit la fonction `calculerPoints()` qui calcule le total des points accumulés par chaque équipe : 3 points par partie gagnée, 1 point par partie nulle et 0 point sinon.

Entrée	Sortie
Un tableau d'enregistrements, enr , contenant 2 champs, nom et parties , regroupant les informations relatives aux équipes d'une même catégorie d'une université donnée.	Un ensemble de cellules, Cell de taille $[N \times 2]$, N étant le nombre d'équipes dans l'enregistrement en entrée. Chaque ligne de Cell est organisée comme suit : <ul style="list-style-type: none"> • colonne 1 : le nom de l'équipe • colonne 2 : les points accumulés.

Soit le début d'implémentation suivant

```
function [Cell] = calculerPoints(enr)

Cell = {};
if nargin ~= 1 || nargout ~= 1 || isempty(enr)
    error('Erreur');
end
```

Quelle séquence d'instructions permet de réaliser le calcul de points par équipe ? (donnez la lettre correspondante).

```
a) for i = 1:size(enr,1)
    Cell{i,1} = enr(i).nom;
    Cell{i,2} =
sum(enr(i).parties .*[3 1 0]);
end
```

```
c) for i = 1:length(enr)
    Cell{i,1} = enr(i).nom;
    Cell{i,2} =
sum(enr(i).parties .*[3 1 0]);
end
```

```
b) for i = 1:length(enr)
    Cell{i,1} = enr.(i)nom;
    Cell{i,2} =
all(enr.(i)parties .*[3 1 0]);
end
```

```
d) for i = 1:length(enr)
    Cell{1,i} = enr(i).nom;
    Cell{2,i} =
sum(enr(i).parties .*[3 1 0]);
end
```

Solution 4.2

c)

4.3 Soit la fonction `trierEquipes()` qui trie les équipes en ordre décroissant du nombre de points accumulés

Entrée	Sortie
Un ensemble de cellules, Cell , de taille [N X 2]. Chaque ligne de Cell est organisée comme suit : <ul style="list-style-type: none"> • colonne 1 : le nom de l'équipe • colonne 2 : les points accumulés. 	Un ensemble de cellules, cellTrie , de même format que Cell , où les équipes sont triées.

et l'implémentation

```
function [cellTrie] = trierEquipes(Cell)
cellTrie = {};

if nargin ~= 1 || nargout ~= 1 || isempty(Cell)
    error('Erreur');
end

for i = 1:size(Cell,1)
    pos = 1;
    for j = 1:size(Cell,1)
        if Cell{j,2} > Cell{pos,2}
            pos = j;
        end
    end

    À compléter... ;
end
```

Question 4 (suite)

À l'aide du symbole d'affectation (=), combiner les 5 éléments suivants pour produire 3 instructions et ordonner ces dernières afin de compléter le programme ci-dessus.

Cell{pos,2}	cellTrie{i,2}	Cell{pos,1}
cellTrie{i,1}	-realmax	

Solution 4.3 (1.5 pt)

```
cellTrie{i,1} = Cell{pos,1};
cellTrie{i,2} = Cell{pos,2};
Cell{pos,2} = -realmax;
```

4.4 Soit la fonction `choisirFinalistes()` qui sélectionne les 8 meilleures équipes.

Entrée	Sortie
Un ensemble de cellules, Cell de taille [N X 2]. Chaque ligne de Cell est organisée comme suit : colonne 1 : le nom de l'équipe colonne 2 : le nombre de points accumulés. Les équipes sont triées en ordre décroissant du nombre de points accumulés	Un ensemble de cellules, finalistes de taille [8 X 1] contenant le nom des 8 meilleures équipes.

Donner le prototype de cette fonction.

Solution 4.4

```
function [finalistes] = choisirFinalistes(Cell)
```

Question 4 (suite)

4.5 Soit la fonction `planifierRencontre()` qui détermine le calendrier des rencontres, c'est-à-dire, les 8 équipes devant s'affronter en quart de finale. Le calendrier est basé sur le classement comme suit : 1^{er} vs 8^e, 2^e vs 7^e, etc.

Entrée	Sortie
Un tableau d'enregistrements, enr , contenant 2 champs, nom , et parties , relatives aux équipes d'une même catégorie d'une université donnée	Un ensemble de cellules, programmation de taille [4 X 1]. Chaque ligne de programmation contient une chaîne de caractères de la forme 'nomEqX vs nomEqY' (voir exemple au début de la question)

Soit le code suivant :

```
function [programmation] = planifierRencontre(enr)

programmation = {};
if nargin ~= 1 || nargout ~= 1 || isempty(enr)
    error('Erreur');
end

% calculer les points de chaque équipe,
% trier les équipes en ordre décroissant du nombre de points
% sélectionner les 8 meilleures équipes
```

À compléter 1... ;

```
for i = 1:4

    programmation{i,1} = À compléter 2... ;
```

end

En vous aidant de certaines fonctions définies précédemment, compléter les deux sections de code.

Solution 4.5 (1.5 pt)

```
1) eqPts = calculerPoints(enr);

   eqTriées = trierEquipes(eqPts);

   meillEq = choisirFinalistes(eqTriées);

2) [meillEq{i} ' vs ' meillEq{8 - i + 1}];
```

4.6 Soit la fonction `quartFinale()` qui, pour chacune des universités en entrée (au moins une université), détermine la programmation des quarts de finale, par catégorie (bac et cycles supérieurs).

Entrée	Sortie
Un nombre quelconque de tableaux d'enregistrements (un tableau par université) contenant les champs nom , cycle et parties , relatives à toutes les équipes d'une université donnée.	Pour chacune des universités, la fonction retourne un enregistrement contenant 2 champs, Bac et Sup . Chaque champ représente un ensemble de cellules de taille [4 X 1] contenant la programmation des quarts de finale pour la catégorie en question.

En vous aidant des fonctions `split()` et `planifierRencontre()` définies précédemment, écrire la fonction `quartFinale()` (prototype et code Matlab). Il faut uniquement vérifier le nombre d'entrées et de sorties et signaler tout problème avec `error()`.

Solution 4.6 (1.75 pt)

```
function [varargout] = quartFinale(varargin)
varargout = {};
if nargin < 1 || nargin ~= nargout
    error('Erreur');
end
for i = 1:length(varargin)
    [bac, sup] = split(varargin{i});
    varargout{i}.bac = planifierRencontre(bac);
    varargout{i}.sup = planifierRencontre(sup);
end
```

end

Question 4 (suite)

4.7 Soit le programme principal ci-dessous qui prend en entrée les informations de 4 différentes universités, afin d'afficher le calendrier des quarts de finale. Les données sur les universités sont sauvegardées dans un enregistrement contenant les champs suivants :

- nom : le nom de l'université
- equipes : un tableau d'enregistrements contenant les champs : nom, cycle et parties, relatives aux équipes d'une université donnée (toutes catégories confondues).

```
% Programme principal %

% Information des équipes des 4 universités
for i = 1:4
    fprintf('Entrez les informations de la %i ème université\n', i);
    Univ{i} = input('');
end

% Déterminer la programmation des quarts de finale

    À compléter 1... ;

U = {U1; U2; U3; U4};

% Affichage de la programmation des quarts de finale

fprintf('Tournoi de Hockey Cosom - Quarts de finale\n\n');

for i = 1:4
    fprintf('%i-%s\n*****\n',i, Univ{i}.nom);
    fprintf('Bac\t\t\tCycles Supérieurs\n\n');
    for j = 1:4
        fprintf('%s\t\t\t%s\n', À compléter 2 ...);
    end
    fprintf('_____ \n');
end
```

4.7.1 En une seule affectation (une étape), déterminer la programmation des matchs des quarts de finale des 4 universités afin d'assigner une valeur aux variables U1, U2, U3 et U4 (à compléter 1).

Question 4 (suite)**Solution 4.7.1 (0.5 pt)**

```
[U1, U2, U3, U4] = quartFinale(Univ{1}.equipes, Univ{2}.equipes,
Univ{3}.equipes, Univ{4}.equipes);
```

4.7.2 Choisir l’instruction qui permet de compléter l’affichage de la programmation pour les quarts de finale au bac et aux cycles supérieurs (à compléter 2)?

a) $U\{j\}.\text{bac}\{i\}, U\{j\}.\text{sup}\{i\}$
b) $U\{i\}.\text{sup}\{j\}, U\{i\}.\text{bac}\{j\}$
c) $U\{i\}.\text{bac}(j), U\{i\}.\text{sup}(j)$

d) $U\{i\}.\text{bac}\{j\}, U\{i\}.\text{sup}\{j\}$

Solution 4.7.2 (0.25 pt)

d)

4.8 Votre employeur a trouvé le fichier texte `parties.txt` avec plusieurs équipes et parties qui ont été jouées entre toutes les universités canadiennes et demande d’écrire un programme pour sauvegarder ces informations et le réutiliser avec les fonctions déjà écrites. La structure du fichier (dont le nombre de lignes (équipes) n’est pas connu à l’avance) est :

nom d’une université : chaîne de caractères
 nom de l’équipe : un mot, chaîne de caractères
 cycle d’étude : un caractère (`'B'`, `'M'` ou `'D'`)
 parties : 3 entiers correspondant à G N P

```
université_1
nomEquipe    cycle
parties
université_2
nomEquipe    cycle
parties
université_1
nomEquipe    cycle
parties
université_3
nomEquipe    cycle
parties
université_2
nomEquipe    cycle
parties
. . .
```

Question 4 (suite)

4.8.1 Compléter le code suivant afin de lire le fichier `parties.txt` et le sauvegarder dans l’ensemble de cellules `stat`. L’ensemble de cellules a la taille $N \times 2$ où N est le nombre d’équipes écrites dans le fichier texte à lire. Sur chaque ligne, la première colonne est le nom d’une université, la deuxième colonne est un enregistrement ayant les champs suivants :

- `nom` : nom de l’équipe (une chaîne de caractères) ;

- `cycle` : cycle d'études auquel appartiennent les membres de l'équipe (un caractère : 'B' pour bac, 'M' pour maîtrise et 'D' pour doctorat) ;
- `parties` : vecteur-ligne de 3 entiers [G N P] (G, N et P étant respectivement le nombre de parties gagnées, nulles ou perdues tout au long de la saison régulière).

```
%ouverture fichier

id =   à compléter 1 ...

% verification ouverture

if     à compléter 2 ...

    error('ouverture echouee')

else

    i=1;
% lecture fichier texte, sauvegarde dans l'ensemble de cellules stat

        à compléter 3 ...

        i=i+1;
        à compléter 4 ...
    end

        à compléter 5 ...
end
```

Solution 4.8.1 (2.25 pt)

```
1) id=fopen('parties.txt', 'rt')

2) id==-1

3)
ligne = fgetl(id)
while ischar(ligne)
stat{i,1} = ligne
ligne = fgetl(id) %accepter aussi la solution avec fscanf().
stat{i,2}.nom = ligne(1 :end-2) % aussi = fscanf(id, '%s', 1)
stat{i,2}.cycle = ligne(end) % aussi = fscanf(id, '%s', 1)
stat{i, 2}.parties = fscanf(id, '%i %i %i', [1 3]);

% i=i+1;
4) ligne = fgetl(id)

5) fclose(id)
```


Question 4 (suite)

4.8.2 Compléter le code suivant afin d'écrire l'information sauvegardée dans l'ensemble de cellules `stat`. Toute statistique concernant une université doit être écrite dans un fichier texte dont son nom est le nom de l'université (i.e. toutes les équipes de Poly doivent être écrites dans le fichier `Poly.txt`). Le format de chaque fichier texte est :

```
nom           % nom de l'équipe
cycle         % cycle d'étude : un caractère ('B', 'M' ou 'D')
parties       % 3 chiffres, entiers correspondant à G N P
nom           % nom d'une autre équipe
cycle
parties
...
```

Les noms des fichiers doivent être créés à partir de l'information existante dans l'ensemble de cellules `stat`. Le programme affiche à l'écran un message chaque fois, qu'un fichier ayant le nom d'une université n'a pas été trouvé parmi les fichiers textes créés.

```
% répétition pour parcourir l'ensemble de cellules
for i=1:size(stat,1)

    uni = stat{i,1}

    % vérification si le fichier existe, affichage message s'il n'existe
    pas

        à compléter 1 ...

    % écriture dans le fichier texte

        à compléter 2 ....

end
```

Solution 4.8.2 (1.25 pt)

```
1) % vérification si le fichier existe, affichage message s'il
n'existe pas
    idv= fopen([uni '.txt'], 'rt')
    if idv == -1
        disp('le fichier n''existe pas')
    else
        fclose(idv)
    end

2) % écriture dans le fichier texte
    ide = fopen([uni '.txt'], 'at')
    if ide == -1
        disp('erreur')
```

```

else
    fprintf(ide, '%s\n%s\n%i\t%i\t%i\n', stat{i,2}.nom,
        stat{i,2}.cycle, stat{i,2}.parties(1), stat{i,2}.parties(2),
        stat{i,2}.parties(3))
    fclose(ide)
end
end
end

```

Question 4 (suite)

4.8.3 Réécrire le code écrit en 4.8.2 pour la partie écriture, mais pour un fichier binaire. Vous devez vous assurer que le fichier binaire puisse être relu. N'oubliez pas que dans un fichier binaire, il n'y a pas de saut de ligne, donc toute l'information sera écrite sur une seule ligne.

% écriture dans le fichier binaire
à compléter ...

Solution 4.8.3 (1 pt)

```

(P) % écriture dans le fichier binaire
ideb = fopen([uni '.bin'], 'a')
if ideb == -1
    disp('erreur')
else
    fwrite(ideb, length(stat{i,2}.nom), 'int16')
    fwrite(ideb, (stat{i,2}.nom), 'char')
    fwrite(ideb, stat{i,2}.cycle, 'char')
    fwrite(ideb, stat{i,2}.parties(1), 'int16')
    fwrite(ideb, stat{i,2}.parties(1), 'int16')
    fwrite(ideb, stat{i,2}.parties(1), 'int16')
    fclose(ideb)
end
end
end

```