

Question 1 (3 points)

Veillez cocher la bonne réponse.

	Vrai	Faux
a) L'addition de deux nombres <code>uint8</code> peut donner une retenue, un débordement alors que la réponse est correcte.		
b) La représentation IEEE754 simple précision du -12,25 en base 10 est : 1 1000 0010 1000 1000 0000 0000 0000 000.		
c) La représentation <code>int32</code> du -12,25 en base 10 est : 1 1000 0010 1000 1000 0000 0000 0000 000.		
d) <code>etime(t1, t2)</code> donne le même résultat si <code>t1</code> et <code>t2</code> sont calculés avec les fonctions <code>clock()</code> ou <code>cputime()</code> .		
e) Le nombre 1010 en base 2 est la lettre A en base hexadécimal.		
f) Le nombre 129 en base 10 peut être représenté en utilisant <code>uint8</code> .		
g) La fonction <code>error()</code> affiche un message d'erreur, mais n'arrête pas le programme.		
h) Dans les fichiers binaires, il n'y a pas de fin de ligne.		
i) L'identificateur d'un fichier (texte ou binaire) donne la taille des données écrites dans le fichier.		
j) <code>nargin()</code> et <code>length(varargin)</code> donnent toujours le même résultat.		

Solution :

	Vrai	Faux
a) L'addition de deux nombres <code>uint8</code> peut donner retenue, débordement alors que la réponse est correcte. (0.25 point)		X
b) La représentation IEEE754 simple précision du -12,25 en base 10 est : 1 1000 0010 1000 1000 0000 0000 0000 000 (0.5 point)	X	
c) La représentation <code>int32</code> du -12,25 en base 10 est : 1 1000 0010 1000 1000 0000 0000 0000 000 (0.25 point)		X
d) Soient : <code>etime(t1, t2)</code> donnent le même résultat si <code>t1</code> et <code>t2</code> sont calculés avec les fonctions <code>clock()</code> ou avec <code>cputime()</code> . (0.25 point)		X
e) Le nombre 1010 en base 2 est la lettre 'A' en hexadécimal. (0.5 point)	X	
f) Le nombre 129 en base 10 peut être représenté en utilisant <code>uint8</code> . (0.25 point)	X	
g) La fonction <code>error()</code> affiche un message d'erreur mais n'arrête pas le programme. (0.25 point)		X
h) Dans les fichiers binaires il n'y a pas de fin de ligne. (0.25 point)	X	
i) L'identificateur d'un fichier (texte ou binaire) donne la taille des données écrites dans le fichier. (0.25 point)		X
j) <code>nargin()</code> et <code>length(varargin)</code> donnent toujours le même résultat. (0.25 point)		X

Question 2 (3 points)

Le programme ci-dessous devrait tracer 3 graphiques dans la même figure : `x1i` en fonction de `ti`, `x2i` en fonction de `ti` et `x3i` en fonction de `ti1`. Le code aurait dû aussi exporter le graphique dans un fichier `.jpeg` ayant le nom « `mon_graphique` ». Malheureusement plusieurs erreurs se sont glissées dans le code. **Identifiez** seulement 6 erreurs, **indiquez** dans le tableau après le programme la ligne, et **expliquez** l'erreur trouvée.

```

1  clear all; clc
2
3  w = 1;
4
5  ti = 0:0.01;7;
6  ti1 = -pi:0.01:3*pi;
7  x1i = x0*log(ti) ;
8  x2i = -w*x0*(ti1);
9  x3i = 2*sin(ti1);
10
11 figure(1)
12
13 plot(ti,x1i, '-r*', 'LineWidth', 2)
14
15 plot(ti,x2i, ':bo', 'LineWidth', 2)
16
17 labelx('t(sec)')
18
19 ylabel('x1, x2')
20
21 titre('Solution')
22
23 export -djpeg -r300
24
25 hold off;

```

Ligne	Erreur

Solution (0.50 point/réponse correcte)

2 – valeur x0

4 - ; dans la matrice de valeurs

8 - x2i en fonction de ti et non ti1

14 – hold on ;

16 – il manque le plot pour xi3

17 – xlabel au lieu de labelx()

19 – il manque ‘ pour la chaine de caracteres a l’interieur de ylabel

21 – title au lieu de titre

23 – print au lieu du export

23 – il manque le titre du fichier jpeg.

Question 3 (7 points)

On vous engage pour mettre sur pied, un système de gestion de bulletins qui affiche le menu suivant :

SYSTÈME DE GESTION DES BULLETINS

- 1 - Ajouter un étudiant
- 2 - Produire le bulletin d'un étudiant
- 3 - Quitter

Ce système manipule 2 types de fichiers :

- Un fichier texte, `donnees.txt`, déjà créé, regroupant les informations de **tous** les étudiants. Chaque ligne du fichier texte contient le matricule, le nom et le prénom d'un étudiant. On considère que le nom et le prénom sont chacun un mot.
- Un fichier binaire contenant, pour chaque étudiant, le matricule (chaîne de 7 caractères), et les notes (`float32`) des 5 cours suivis au cours de la session.

Notes :

- Le fichier `donnees.txt` contient les informations de tous les étudiants et il ne peut pas être modifié. Le fichier binaire peut être modifié.
- Il n'est pas nécessaire de vérifier les entrées et sorties des fonctions.
- Il ne faut pas utiliser la fonction `error()`.

- L'utilisation de fonctions `textscan()`, `importdata()`, `uiimport()`, `stringsplit()`, `feof()` est interdite.

Soit le programme principal :

```

clc;
clear all

% Création du fichier binaire
bFile = input('Entrer le nom du fichier binaire(sans son extension): ','s');
notes = input('Entrer les matricules et notes des étudiants: ');
creerFichierBinaire(bFile, notes);

CONTINUE = 1;

menu = sprintf('SYSTÈME DE GESTION DES BULLETINS\n\n\t1- Ajouter un
étudiant\n\t2- Produire le bulletin d''un étudiant\n\t3- Quitter\n\n');

while CONTINUE
    disp(menu);
    choix = input('Votre choix: ');

    switch choix
        case 1
            matr = input('Entrer le matricule de l''étudiant: ', 's');

            ajouterEtudiant(); % à écrire

            if ~ajoute
                fprintf('L''étudiant existe déjà dans le système\n\n');
            else
                fprintf('L''étudiant a été ajouté dans le système\n\n');
            end
        case 2
            matr = input('Entrer le matricule de l''étudiant: ', 's');

            produireUnBulletin(); % à écrire

            if ok
                fprintf('Bulletin généré...\n\n');
            else
                fprintf('Impossible de générer le bulletin\n\n');
            end
        case 3
            CONTINUE = 0;
        otherwise
            disp('Choix invalide');
    end
end
end

```

Soit la fonction déjà implémentée:

```
function [trouve, index] = chercherEtudiant(bFile, matr)
```

qui prend en entrée :

- le nom d'un fichier binaire (sans son extension) `bFile`
- le matricule d'un étudiant `matr`.

La fonction parcourt le fichier et recherche le matricule spécifié. Si ce dernier a été trouvé, la fonction retourne :

- une variable booléenne, `trouve` 1 si l'étudiant a été trouvé, 0 sinon.
- la position, `index`, du matricule dans le fichier, si l'étudiant a été trouvé, -1 sinon. La position `index` correspond au nombre d'octets à lire avant de pouvoir lire le matricule spécifié.

En vous aidant de la fonction `chercherEtudiant()`, écrire les fonctions suivantes (prototypes et codes MATLAB):

3.1 `ajouterEtudiant()` qui prend en entrée :

- le nom d'un fichier binaire (sans son extension) `bFile`
- le matricule d'un étudiant `matr`.

La fonction vérifie d'abord si l'étudiant (son matricule) figure déjà dans le fichier. Si l'étudiant n'est pas présent dans le fichier binaire, la fonction demande les 5 notes de l'étudiant sous le format suivant [`note1 note2 note3 note4 note5`], et ajoute l'étudiant (matricule et notes) à la fin du fichier binaire. On suppose que les étudiants suivent toujours 5 cours.

La fonction retourne `ajoute`, qui est égal à 1 si l'étudiant a été ajouté au fichier binaire et 0 sinon.

3.2 `chercherInfos()` qui prend en entrée le matricule d'un étudiant `matr`.

La fonction parcourt le fichier texte `donnees.txt` pour rechercher le nom et le prénom de l'étudiant dont le matricule est spécifié en entrée.

La fonction retourne les variables `nom` et `prénom`, correspondant au nom et prénom de l'étudiant recherché. On suppose que le matricule recherché figure toujours dans le fichier `donnees.txt`.

3.3 `produireUnBulletin()` qui prend en entrée :

- le nom d'un fichier binaire (sans son extension) `bFile`
- le matricule d'un étudiant `matr`.

La fonction vérifie d'abord si l'étudiant (son matricule) figure déjà dans le fichier. Si l'étudiant est présent dans le fichier binaire, la fonction fait ensuite appel à la fonction `chercherInfos()` pour récupérer les nom et prénom de l'étudiant en question.

Par la suite, la fonction `produireUnBulletin()` lit les notes de l'étudiant dans le fichier binaire (la fonction `chercherEtudiant()` peut vous être très utile!).

Enfin, la fonction `produireUnBulletin()` crée un nouveau fichier texte, ayant pour nom `X_Y_Z.txt`, où `nom`, `X`, `Y` et `Z` correspondent au nom, prénom et matricule de l'étudiant en question. Dans ce fichier, les informations de l'étudiant sont sauvegardées comme suit :

```
nom   prenom   matricule   moy
```

où la `moy` est la moyenne des 5 cours suivis et est affichée avec 2 décimales.

La fonction retourne `ok`, qui est égal à 1 si le bulletin a été généré et 0 sinon.

Réponse :

3.1 - 2.25 points

```
function [ajoute] = ajouterEtudiant(bFic, matricule)%0.25 prototype

ajoute = 0;

trouve = chercherEtudiant(bFic, matricule); %0.25 appel chercheretudiant

if ~trouve %0.25 verification condition
    fid = fopen([bFic '.bin'], 'a'); %1 ouverture, nom correct, verification,
fermeture
    if fid ~= -1
        fwrite(fid, matricule, 'char'); %0.75 ecriture, input, valeur de
retour
        notes = input('Entrez les 5 notes de l''étudiant sous forme de
vecteur ligne: ');
        for i = 1:length(notes)
            fwrite(fid, i, 'float32');
        end
        ajoute = 1;
        fclose(fid);
    end
end
```

3.2 - 1.75 points

```
function [nom, prenom] = chercherInfos(matricule) %0.25 prototype

trouve = 0;
fid = fopen('Donnees.txt', 'rt'); %0.25 ouverture avec 'rt'

if fid ~= -1
    mat = fscanf(fid, '%s', 1); %1.25 lecture preliminaire, while, lecture
suivante, structure pour trouver l'etudiant, valeurs de retour

    while ~isempty(mat) && ~trouve;
        if strcmpi(mat, matricule)
            nom = fscanf(fid, '%s', 1);
            prenom = fscanf(fid, '%s', 1);
            trouve = 1;
        end
    end
end
```

```

        else
            fgetl(fid);
            mat = fscanf(fid, '%s', 1);
        end
    end
    fclose(fid);
end
end

```

3.3 - 3 points

```

function [ok] = produireUnBulletin(bFic, matricule) %0.25 prototype

ok = 0;
notes = zeros(1,5);
[trouve, pos] = chercherEtudiant(bFic, matricule); % 0.25 appel
chercheretudiant()

if trouve %0.25 condition etudiant trouvee
    [nom, prenom] = chercherInfos(matricule); % 0.25 appel chercherinfos()

    fid = fopen([bFic '.bin'], 'r'); % 0.5 ouverture, validation, fermeture
    (0.5 si 3, 0.25 si 2/3, 0 si 1 de 3)

    if fid ~= -1
        fseek(fid, pos + 7 , 'bof'); %0.50 trouver etudiant, lire notes
        notes = fread(fid, [1 5], 'float32');
        fclose(fid);
    end

    fid = fopen([nom '_' prenom '_' matricule '.txt'], 'wt'); % 0.5 creation
nom fichier, ouverture 'wt'
    if fid ~= -1
        fprintf(fid, '%s\t%s\t%s:\t%0.2f/20',nom, prenom, matricule,
mean(notes)); % 0.5 écriture dans le fichier texte
        ok = 1; % 0.25 valeur de retour
        fclose(fid);
    end
end
end

```

Question 4 (7 points)

Ecrivez la fonction (prototype et code MATLAB) `ConvBin()` qui fait la conversion de nombres binaires, entiers non-signés (`uint8`).

La fonction reçoit en entrée un nombre inconnu (indéfini) d'ensembles de cellules. Chaque ensemble de cellules reçu en entrée peut contenir une ou deux cellules.

- Si un ensemble de cellules reçu en entrée lors de l'appel de la fonction contient une seule cellule, la cellule doit être un nombre binaire (une chaîne de caractères de '0' et '1').
- Si un ensemble de cellules reçu en entrée lors de l'appel de la fonction contient deux cellules, la première cellule doit être le nombre binaire à convertir (une chaîne de caractères de '0' et '1'), la deuxième cellule doit être la base dans laquelle le nombre binaire doit être converti (un caractère : 'b' pour base 2, 'd' pour base 10 et 'x' pour base 16).

La fonction retourne :

- Les nombres :
 - convertis dans la base choisie si la fonction reçoit en entrée, lors de l'appel, des ensembles des cellules ayant la taille 2 et le caractère pour la base de conversion est 'd' ou 'x'.
 - binaires reçus en entrée si la taille de l'ensemble de cellules est 1 ou, la taille est 2 et le caractère pour la base de conversion est différent de 'd' ou 'x'.
- Le nombre de conversions faites. On considère qu'une conversion correspond seulement au cas où un nombre binaire a été converti dans une autre base que la base 2.
- Le temps nécessaire pour exécuter le programme.

Vous devez :

- Lire les exemples d'exécution de la fonction.
- Afficher une erreur (le programme s'arrête) si :
 - Le nombre de paramètres d'entrée et valeurs de retour lors de l'appel de la fonction n'est pas respecté.
 - Les tailles des ensembles de cellules reçues en entrée lors de l'appel de la fonction ne sont pas valides.
 - Les tailles de chaînes représentant les nombres binaires `uint8` à convertir ne sont pas valides ou les chaînes contiennent d'autres caractères que '0' et '1'.
- Afficher un avertissement (le programme continue l'exécution mais indique à l'écran les indices des arguments et les raisons des avertissements) si la fonction reçoit en entrée:
 - des ensembles des cellules ayant la taille 1.
 - des ensembles de cellules ayant la taille 2, mais la deuxième cellule est différente de 'b', 'd' ou 'x'.
- Utiliser au moins une fois la structure `switch/case`.

Note :

Il est interdit d'utiliser les fonctions `bin2dec()`, `base2dec()`, `binaryVectorToDecimal()`, `binaryVectorToHex`. La fonction `bin2hex()` n'existe pas.

Exemples d'exécution :

Exemple 1 - appel: [a, b, c]=ConvBin({'10010200'})

```
Error using ConvBin (line 16)
nombre binaire incorrect
Error in test (line 2)
[a, b, c]=ConvBin({'10010200'})
```

Exemple 2 - appel: [a, b, c]=ConvBin({'10101'})

```
Error using ConvBin (line 13)
arguments d'entree incorrects
Error in test (line 3)
[a, b, c]=ConvBin({'10101'})
```

Exemple 3 - appel: [a, b] = ConvBin({'10101010', 'd'})

```
Error using ConvBin (line 3)
erreur nombre valeurs de retour
Error in test (line 5)
[a, b] = ConvBin({'10101010', 'd'})
```

Exemple 4 - appel: [a, b, c, d, e, f, g]=ConvBin({'10000001', 'd'},
{'10000000'}, {'00101010', 'x'}, {'10111101', 'b'}, {'11101111', 'm'})

```
Warning: sans parametre de conversion pour l'argument 2
> In ConvBin at 19
   In test at 5
Warning: choix conversion incorrect argument 5
> In ConvBin at 63
   In test at 5
```

```
a =
    129

b =
10000000

c =
    2A

d =
10111101

e =
11101111

f =
     2

g =
    0.2034
```

Solution :

```
function [varargout] = ConvBin( varargin ) %prototype 0.25
if length(varargin) ~= nargout-2 %0.25 condition
    error('erreur nombre valeurs de retour') %0.25 error
else
    for i=1:length(varargin) % initialisations varargin
        varargout{i}={};
    end
    varargout{i+1} = 0; varargout{i+2} = 0; nb=0; temp=0; %0.5 initialisation
    tic
    for i=1:length(varargin) %parcours varargin %0.25
        if length(varargin{i})>2 || length(varargin{i}{1}) ~= 8
            error('arguments d''entree incorrects')%0.5 condition length
varargin
        else
            if (sum(varargin{i}{1}=='1') + sum(varargin{i}{1}=='0')) ~=
length(varargin{i}{1}) %0.25 condition seulement des 1 et 0
                error('nombre binaire incorrect')
            end
            if length(varargin{i}) == 1 % 0.25 avertissement
                warning('sans parametre de conversion pour l''argument %i',i)
%0.5 syntaxe complete avec indice
                varargout{i} = varargin{i}{1}; %0.25 valeur de retour
            else
                a=varargin{i}{2};

                switch a % conversion b, d, x %0.25
                    case 'b' %0.25 case 'b'
                        varargout{i}=varargin{i}{1};
                    case 'd' %0.5 case 'd'
                        v1=0;
                        for jj=1:length(varargin{i}{1})
v2=str2num(varargin{i}{1}(jj))*2^(length(varargin{i}{1})-jj);
                            v1=v1+v2; %0.25 calcul
                        end
                        varargout{i}=v1; nb=nb+1;%0.25
                    case 'x'
                        v1=[];
%0.5 conversion hexa:
                        for jj=length(varargin{i}{1}):-4:1
v2 = str2num(varargin{i}{1}(jj))*2^0+str2num(varargin{i}{1}(jj-1))*2^1 + ...
str2num(varargin{i}{1}(jj-2))*2^2 + str2num(varargin{i}{1}(jj-3))*2^3;
                            switch v2 %0.5 structure valeurs hexa
                                case 10
                                    v2 = 'A';
                                case 11
                                    v2 = 'B';
                                case 12
                                    v2 = 'C';
                                case 13
                                    v2 = 'D';
                                case 14
```

```

        v2 = 'E';
    case 15
        v2 = 'F';
    otherwise
        v2 = num2str(v2);
    end
    v1=[v2 v1];%.25 concatenation regroupements de 4
end
varargout{i}=v1;
nb=nb+1;
otherwise %.25 si argument 2 incorrect
    warning('choix conversion incorrect argument %i', i)
    varargout{i}=varargin{i}{1};
end
temp=toc; %.25 calcul temps (on donne les points n'importe pas
ou tic a été place a l'exterieur de for i=1:length(varargin))
end
end
end
varargout{i+1} = nb; %.5 valeurs de retour nombre conversions et temps
varargout{i+2} = varargout{i+2} + temp;
end

```