

Question 1**(3.75 points)****1.1)** Répondez par vrai ou faux, vous ne devez pas justifier vos réponses :

- a) La fonction `format()` n'affecte pas les calculs ou la sauvegarde des variables en mémoire.
- b) Soit la matrice carrée $C = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$.
`strcmp(C,C)` et `isequal(C,C)` donnent le même résultat.
- c) Soit la matrice carrée $C = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$.
 C^2 et $C.^2$ donnent le même résultat.
- d) Soit `A= 'Montreal'`, les fonctions `size(a)` et `length(a)` donnent le même résultat.
- e) (tout `if` peut être remplacé par un `switch`) & (tout `for` peut être remplacé par un `while`)
- f) (tout `if` peut être remplacé par un `switch`) | (tout `for` peut être remplacé par un `while`)

1.2) Soit la matrice

$$C = \begin{bmatrix} 3 & -11 & -5.25 \\ 3.5 & 4 & 4 \\ 3.25 & 4 & 2.35 \end{bmatrix}$$

Donnez le résultat de l'exécution du code suivant (pour chaque sous question, on considère la matrice `C` non modifiée):

a)

```
for i=1:size(C, 1)*size(C,2)
    if (rem(C(i),1)~=0)
        C(i)=0;
        break;
    end
end
disp(C)
```

b) `C(C==4)=C(mod(C,1)~=0)`**Solution :****Question simple : V F F F F V**

$$\text{c) 2) a) } C = \begin{bmatrix} 3 & -11 & -5.25 \\ 0 & 4 & 4 \\ 3.25 & 4 & 2.35 \end{bmatrix}; \text{ b) erreur}$$

Question 2

(2.5 points)

Soient les variables suivantes :

```
Matrice = [101  1.5  2.1  101;
           0.89 0.3  0.75 8.6;
           1.4  0.7  1.5  0.4;
           101  20.6 20.7 101] ;
Chaine = 'Je vais bien réussir mon intra' ;
Ensemble = {'Mario Dupont', 11, 'Octobre', 1993, 'Quebec'} ;
Cellule = {Matrice, Chaine;
           Ensemble, 1:2:56};
```

Chaque sous-question est indépendante et considère l'ensemble des variables non modifiées. Une matrice de taille $M \times N$ est une matrice de M lignes et N colonnes. L'utilisation de structures de répétition et de décision est interdite.

- 2.1) Donnez 3 instructions différentes qui permettent de remplacer tous les nombres supérieurs à 100 de la matrice `Matrice` avec son double. Vous pouvez utiliser, pour chaque instruction une seule affectation.
- 2.2) En utilisant seulement 2 affectations, sauvegardez dans l'enregistrement « étudiant » ayant les champs « nom » et « ville » le premier et respectivement le dernier élément de la cellule `Ensemble`.
- 2.3) En utilisant seulement une affectation, concaténez verticalement la matrice `Matrice` avec sa transposée et mettez la matrice résultante dans la cellule `Cellule` à la place de `Chaine`.
- 2.4) Donnez 3 instructions différentes qui permettent de remplacer dans la cellule `Chaine` de la variable `Cellule`, les lettres `i` qui sont en minuscules, en majuscules. Vous pouvez utiliser, pour chaque instruction une seule affectation.

Solution

2.1) ATTENTION, IL Y A PLUSIEURS SOLUTIONS POSSIBLES)

`Matrice([1 end],[1 end])=2*Matrice([1 end], [1 end])`

`Matrice(Matrice>100) = 2*Matrice(Matrice>100)`

Matrice([1 4 13 16])= 2*Matrice(Matrice>100)

Matrice(Matrice==101)=2*Matrice(Matrice==101)

2.2) etudiant.nom= Ensemble{1} ; etudiant.ville= Ensemble{end}

2.3) (Cellule{1,2}= [Matrice ; Matrice']

2.4) (ATTENTION, IL Y A PLUSIEURS SOLUTIONS POSSIBLES)

Cellule {1,2}(Cellule {1,2}=='i')=upper(Cellule {1,2}(Cellule{1,2}=='i'))

Cellule {1,2}(Cellule {1,2}=='i')='I'

Cellule {1,2}(Cellule {1,2}=='i')=upper('i')

Question 3

(4.25 points)

Le théâtre *PolyT* décide de présenter 3 spectacles différents en une fin de semaine (jeudi, vendredi et samedi soir). Un individu peut se procurer, soit un billet simple (pour une soirée) ou un billet double (pour 2 soirs) ou une passe pour l'ensemble des 3 présentations. Le prix du billet varie selon plusieurs paramètres tels que le type de billet (Simple/Double/Passe), le choix de l'emplacement du siège (Parterre/Balcon) et l'âge du client. De plus, pour encourager les étudiants (es), *PolyT* offre 20% de rabais sur les billets doubles, et 25% de rabais sur les passes.

Pour mieux faciliter la gestion de son système de vente en ligne, *PolyT* enregistre les informations relatives à un client donné dans un ensemble de cellules comme suit :

Informations = {'Type de billet', 'Siège', 'Catégorie', Étudiant(e)}

Les 3 premières cellules de Informations contiennent des chaînes de caractères représentant les choix du client : type de billet (Simple/Double/Passe), siège (Parterre/Balcon), la catégorie (Enfant, Adulte ou Aîné) à laquelle il appartient; et la dernière cellule contient un scalaire (0 ou 1), 1 si le client est un(e) étudiant(e), 0 sinon.

Écrivez uniquement la partie d'un programme MATLAB qui permet de calculer le prix du billet en fonction des choix et du statut (catégorie d'âge, étudiant(e) ou non) du client, en considérant les tarifs présentés dans le tableau suivant :

Type de billet	Emplacement du siège	Enfant (Age < 12)	Adulte (12 ≤ Age < 60)	Aîné (Age ≥ 60)
Simple	Parterre	50\$	80\$	75\$
	Balcon	45\$	70\$	65\$

Double	Parterre	90\$	150\$	140\$
	Balcon	80\$	130\$	120\$
Passe	Parterre	130\$	220\$	205\$
	Balcon	110\$	190\$	175\$

Notes:

- L'ensemble de cellules `Informations` existe, vous ne devez pas le demander à l'utilisateur.
- Vous ne devez pas valider les entrées.
- Vous devez utiliser seulement des structures `switch`, l'utilisation de `if` est interdite
- On considère que toute personne, peu importe la catégorie d'âge, peut être un étudiant.
- Afficher le prix du billet, à deux décimales près, à l'aide de la fonction `fprintf()`.

Exemple :

Pour un client qui a choisi : billet double, balcon, est âgé de 25 ans et est un étudiant :

```
Informations = {'Double', 'Balcon', 'Adulte', 1}
```

Le prix du billet est : 104.00\$

Solution

```
Prix = 0;
```

```
switch Informations{1} % tous les case considérées
case 'Simple' % %Billet simple
    switch Informations{2} %switch {2}
        case 'Parterre' %% Emplacement de siège: Parterre
            switch Informations{3} % switch {3}
                case 'Enfant' % Enfant
                    Prix = 50;
                case 'Adulte' % Adulte
                    Prix = 80;
                case 'Aine' % Aine
                    Prix = 75;
            end
        % Emplacement de siège: Balcon
    case 'Balcon'
        switch Informations{3}
            case 'Enfant'
                Prix = 45;
            case 'Adulte'
                Prix = 70;
            case 'Aine'
                Prix = 65;
```

```
        end
    end
    case 'Double'          % Billet double
        switch Informations{2}
            case 'Parterre'
                switch Informations{3}
                    case 'Enfant'
                        Prix = 90;
                    case 'Adulte'
                        Prix = 150;
                    case 'Aine'
                        Prix = 140;
                end
            case 'Balcon' % Emplacement de siège: Balcon
                switch Informations{3}
                    case 'Enfant'
                        Prix = 80;
                    case 'Adulte'
                        Prix = 130;
                    case 'Aine'
                        Prix = 120;
                end
            end
        end
        Rabais = Informations{4} == 1; % prix
        Prix = Prix*(1 - Rabais*0.2);
    case 'Passe'          % Passe
        switch Informations{2}
            case 'Parterre'
                switch Informations{3}
                    case 'Enfant'
                        Prix = 130;
                    case 'Adulte'
                        Prix = 220;
                    case 'Aine'
                        Prix = 205;
                end
            case 'Balcon'
                switch Informations{3}
                    case 'Enfant'
                        Prix = 110;
                    case 'Adulte'
                        Prix = 190;
                    case 'Aine'
                        Prix = 175;
                end
            end
        end
        Rabais = Informations{4} == 1;
        Prix = Prix*(1 - Rabais*0.25);
    end
    fprintf('Le prix du billet est: %0.2f$\n', Prix) %
```

Question 4**(3.75 points)**

Soit le vecteur v , 1×10 contenant des valeurs réelles entre 0 et 1, trié en ordre croissant et la variable, $t=0.5$; trouvez le nombre d'éléments de v inférieurs à t en utilisant trois stratégies différentes :

- 4.1) En utilisant une structure de répétition `while`.
- 4.2) En utilisant une structure de répétition `for`. Il faut éviter des comparaisons inutiles avec chaque élément de v .
- 4.3) En utilisant ni `while` ni `for`, mais une seule instruction MATLAB.

Après chaque implémentation affichez avec `fprintf()` le nombre d'éléments inférieurs à t .

Solution

```
A) i=1; % noter i=1 important %
while i <= length(v) && v(i) < t % noter && est important %
    i=i+1;
end
fprintf('Il y a %i éléments inférieur à %i.\n', i-1,t); %
% noter i-1 important
```

```
B) %
for i=1:length(v)
    if v(i) < t
        j=i;
    else
        break;
    end
end
fprintf('Il y a %i éléments inférieur à %i.\n', j, t);
% C) %%
i=nnz(v<t); %i=sum(v<t) %i=length(v(v<t))
fprintf('Il y a %i éléments inférieur à %i.\n', i, t);
```

Question 5**(5.75 points)**

La STM vous embauche pour écrire un programme qui :

- demande à l'utilisateur d'entrer
 - ✓ le code de la station de départ et
 - ✓ l'heure de départ et le code de la station de destination et
- affiche à l'écran
 - ✓ le nom de la station de départ et l'heure de départ
 - ✓ le nom de la station de destination et l'heure d'arrivée.

Pour écrire le programme, la STM vous a fourni, pour une ligne de bus de l'île de Montréal, un ensemble de cellules qui contient toutes les stations, en ordre du passage. Sur chaque ligne de l'ensemble de cellules, on trouve toujours :

- Le code de la station - un scalaire, entier
- Le nom de la station d'autobus – une chaîne de caractères
- Les heures de départ – un ensemble de cellules où, sur chaque ligne, on a un vecteur des heures de départ. Le premier de chaque ligne élément est toujours l'heure et les autres éléments sont les minutes, par exemple [6 06 21 35 49] signifie que les autobus partent à 6h06, 6h21, 6h35, et à 6h49.

$$\left\{ \begin{array}{l}
 \begin{array}{l}
 608383 \text{ 'Metro Cote Vertu'} \left\{ \begin{array}{l}
 [5 \ 45] \\
 [6 \ 06 \ 21 \ 35 \ 49] \\
 [7 \ 01 \ 13 \ 25 \ 42] \\
 [8 \ 03 \ 24 \ 44] \\
 [9 \ 04 \ 24 \ 46] \\
 [10 \ 09 \ 32 \ 54] \\
 \vdots
 \end{array} \right\} \\
 \\
 55594 \text{ 'Cote Vertu/Marcel Laurin'} \left\{ \begin{array}{l}
 [5 \ 47] \\
 [6 \ 08 \ 23 \ 37 \ 51] \\
 [7 \ 03 \ 15 \ 27 \ 44] \\
 [8 \ 05 \ 26 \ 46] \\
 [9 \ 06 \ 26 \ 48] \\
 [10 \ 11 \ 34 \ 56] \\
 \vdots
 \end{array} \right\} \\
 \\
 55961 \text{ 'Marcel Laurin/ Thimens'} \left\{ \begin{array}{l}
 [5 \ 48] \\
 [6 \ 09 \ 24 \ 38 \ 52] \\
 [7 \ 04 \ 16 \ 28 \ 45] \\
 [8 \ 08 \ 27 \ 47] \\
 [9 \ 07 \ 27 \ 49] \\
 [10 \ 12 \ 35 \ 57] \\
 \vdots
 \end{array} \right\} \\
 \vdots \\
 \\
 55909 \text{ 'Marcel Laurin/Gal. St Laurent'} \left\{ \begin{array}{l}
 [5 \ 52] \\
 [6 \ 13 \ 28 \ 42 \ 56] \\
 [7 \ 08 \ 20 \ 32 \ 49] \\
 [8 \ 12 \ 31 \ 51] \\
 [9 \ 11 \ 31 \ 53] \\
 [10 \ 16 \ 39] \\
 \vdots
 \end{array} \right\} \\
 \vdots \\
 \vdots
 \end{array} \right.$$

- 5.1)** Le programme demande à l'utilisateur d'entrer le code de la station de départ et le code de la station de destination et vérifie **seulement** si les deux codes sont des entiers sinon le programme affiche un message d'erreur et s'arrête. **Vous ne devez pas vérifier** que le code d'arrêt est unique ou qu'il existe dans l'ensemble de cellules.
- 5.2)** Le programme demande ensuite de choisir l'heure de départ sous forme d'une matrice de 2 éléments : le premier élément est l'heure et le deuxième élément est la minute. On considère que la journée a 24 heures, une heure a 60 minutes et que les autobus ne circulent pas entre 2h00 et 5h00. Le programme vérifie seulement si ces contraintes sont respectées, sinon affiche un message indiquant un problème et l'exécution s'arrête. **Vous ne devez pas vérifier** que les deux éléments de la matrice sont des entiers.
- 5.3)** Le programme calcule l'heure d'arrivée à la destination et affiche avec `fprintf()` le nom de la station et l'heure de départ sous forme « heureminute » (i.e. si la matrice est [6 56] on affiche 6h56) ainsi que le nom de la station de destination et l'heure d'arrivée sous le même format « heureminute ». Si l'heure de départ choisie tombe entre deux valeurs dans le vecteur de valeurs de la minute, on considère la valeur immédiatement plus grande (c.-à-d. si l'heure choisie est [6 43] et dans le vecteur de valeurs de temps on a [6 06 21 35 49], votre programme considère l'heure de départ [6 49]).

Indice : Vous pouvez calculer le temps entre les stations d'autobus en considérant comme référence un passage d'un bus par toutes les stations (i.e. le premier passage d'un autobus).

Notes :

- Le résultat du calcul de la minute peut dépasser 60 (c.-à-d. si l'autobus passe par une station à 6h59 et la prochaine station est 2 minutes plus tard, l'heure d'arrivée est 7h01 et non 6h61) donc vous devez considérer ce cas. Même chose pour l'heure de départ.
- Utilisation de `error()`, `return()`, `break`, `sum()`, `min()`, `max()`, `sort()`, `unique()` est interdite.

Exemple #1 d'exécution du code :

```
code depart 4.5
code destination 55594
code station depart ou destination incorrect
>>
```

Exemple #2 d'exécution du code :

```
code depart 55594
code destination 55909
heure sous forme de matrice de 2 elements [heure minute] [2 25]
l'heure choisie n'est pas correcte
>>
```

Exemple #3 d'exécution du code :

```
code depart 55594
```

```

code destination 55909
heure sous forme de matrice de 2 elements [heure minute] [6 53]
station de depart: Cote Vertu/Marcel Laurin, heure de depart de la
station est: 7h03
destination: Marcel Laurin/Gal. St Laurent, heure d'arrivee est: 7h08
>>

```

Solution

```

code=input('code depart '); % saisie code
dest=input('code destination ');

if rem(code,1) ~= 0 || rem(dest,1)~=0 % condition correcte,
    stricture correcte if else
        disp('code station depart ou destination incorrect')
    else
        heure=input('heure matrice de 2 elements [heure minute] ');
        % saisie heure
        if (heure(1)>1 && heure(1)<5) || heure(2)>59 ||
length(heure)~=2 || heure(1)>23 % condition correcte, structure
correcte if else
            disp('l'heure choisie n'est pas correcte')
        else
            temp=[]; % sauvegarde valeurs de temps dans une matrice
            for t=2:size(bus, 1)
                a=bus{t,3}{1}(2)-bus{t-1,3}{1}(2);
                if a<0
                    b=60 + (bus{t,3}{1}(2)-bus{t-1,3}{1}(2));
                else
                    b=a;
                end
                temp=[temp b];
            end
            for i=1:size(bus, 1) % for trouver codes depart arret
                if isequal(code, bus{i,1}) % decision trouver code
depart arret
                    ind1=i;
                    tmp1=bus{ind1, 3};
                    ii=1;
                    while ii<=length(tmp1) % alg pour donner le temps
correct
                        if isequal(tmp1{ii}(1), heure(1))
                            mins=tmp1{ii}(2:end);
                            t=1;
                            while t<=length(mins) & (heure(2)-mins(t))>0
                                t=t+1;
                            end
                            if t<=length(mins)
                                heure(2)=mins(t)
                            else
                                ii=ii+1;

```

```
        heure(1)=tmp1{ii}(1);heure(2)=tmp1{ii}(2);
    end
        ii=ii+length(tmp1)
    else
        ii=ii+1;
    end
end
end
end
end

for i=1:size(bus, 1) % trouver la station d'arret
    if isequal(dest, bus{i,1})
        ind2=i;
    end
end

hr=heure(1); temps=heure(2);
for tt=ind1:ind2-1 % heure d'arrivee
    temps=temps+temp(tt);
    if temps>=60
        temps=temps-60; hr=heure(1)+1;
    end
end

fprintf('depart: %s heure: %dh%d\n', bus{ind1,2},heure(1),
heure(2)) % affichage correct 2 fprintf() et valeurs correctes heure
fprintf('destination: %s heure: %dh%d\n', bus{ind2,2}, hr, temps)
end end
```