

Nom et Prénom :**Numéro de TP :****Interrogation écrite****Durée de l'épreuve : 60 minutes.****Documents autorisés : supports de cours, notes de cours et de TD/TP.****Les réponses se font sur la feuille d'énoncé. Les barèmes sont donnés à titre indicatif.**

Vos réponses devront être précises et concises, répondant exactement à la question posée.

A. Questions de cours (6 pts) (2+2+1+1)**HTML/CSS****Question 1:** On donne le code suivant extrait d'une page HTML (ce code est directement dans la balise <body> de la page en question) :

```
<span><div><ul><li class="">list item</li></ol></div></span>
<img href="http://monsite.com/images/img.jpg">
<p>Bla bla bla bla bla <a href="autrepage.html">Cliquer ici</p></a>
```

Ce code est-il valide ? Si non, pourquoi ?

Réponse : (2 points)

Pas de <div> dans un , fermé par , propriété de img fausse (src au lieu de href) et ordre de fermeture des balises <p> et <a> incorrect. (0.5/erreur, mettre uniquement 0.25 si ont répondu non sans justifier).

Question 2: On donne le code suivant extrait d'une page HTML (ce code est directement dans la balise <body> de la page en question) ainsi que le code CSS l'accompagnant :**HTML**

```
<table class="larg"><tr>
<th></th>
<th class="ac">Classes</th><th class="ac">Effectifs</th>
</tr><tr>
<th class="ac" rowspan="2">Série L</th>
<td class="ac">1èreL1</td><td class="ad">27</td>
</tr><tr>
<td class="ac">1èreL2</td><td class="aligndroit">29</td>
</tr><tr>
<th class="ac" rowspan="3">Série ES</th><td class="ac">1èreES1</td><td
class="ad">31</td>
</tr><tr>
<td class="ac">1èreES2</td><td class="ad">32</td>
</tr><tr>
<td class="ac">1èreES3</td><td class="ad">33</td>
</tr></table>
```

Nom et Prénom :

Numéro de TP :

CSS

```
table.larg
{
    width: 450px;
}
table.snd
{
    width: 300px;
}
.ac
{
    text-align: center;
    border-width: thin thin thin thin;
    border-style: solid solid solid solid;
}
td.ad
{
    text-align: right;
    border-width: thin thin thin thin;
    border-style: solid solid solid solid;
    color: red;
}
```

Représenter l'affichage à l'écran, vous indiquerez les dimensions sur les bords du dessin.

Réponse : (2 points)

	Classes	Effectifs
Série L	1èreL1	27
	1èreL2	29
Série ES	1èreES1	31
	1èreES2	32
	1èreES3	33

PROGRAMMATION EVENEMENTIELLE/VB

On considère le code suivant:

```
Private Sub btn_Click(sender As System.Object, e As System.EventArgs)
Handles btnOne.Click, btnOne.MouseClick, btnTwo.Click, btnTwo.MouseClick
```

Question 3: Quelles propriétés de ce code permettent de savoir quel est l'objet et l'évènement qui ont déclenché l'exécution de ce code ?

Réponse : (1 point)

sender représente l'objet qui a déclenché l'événement qui est en train d'exécuter ce code.
e représente l'objet événement qui a été déclenché.

Question 4: Les paramètres sender et e sont-ils passés par référence ou par valeur ? Quelle est la différence ?

Réponse : (1 point)

Par défaut, c'est par valeur. Par valeur : on fait une copie, les modifications qu'on fait ne sont pas répercutées sur l'objet original.

Nom et Prénom :

Numéro de TP :

B. Conception d'applications et débogage (6 pts) (1+1.5+3.5)

Question 1: Soit le code

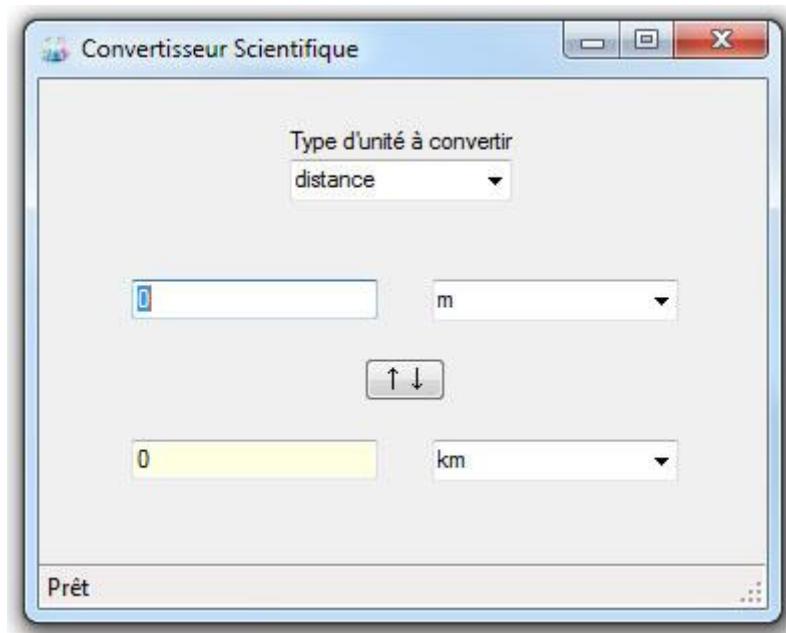
```
Public Sub divise (x As Integer, y As Integer)
    Dim div as Double
    If (x=0) Or (y=0) Then
        div = 1
    else
        div = x/y
    End If
    MsgBox("Le quotient est : " & div)
End Sub
```

Ce code doit calculer le quotient de deux entiers, toutefois, on voit que pour certaines valeurs de x et de y cela ne sera pas le cas. Quel type de test permet de détecter cette erreur : test boîte noire ou test boîte blanche ? Justifiez en une phrase votre réponse.

Réponse : (1 point)

Il faut faire un texte boîte blanche (structurel) car seule l'analyse du code permet de détecter le problème.

Contexte : on utilise un petit convertisseur (SciConv) d'unités scientifiques, dont vous avez une capture ci-dessous.



Nom et Prénom :

Numéro de TP :

On dispose sur cette interface de deux ComboBox, nommés cbSource et cbDest qui, sur la capture d'écran, affichent "m" et "km" respectivement. On considère le code suivant, qui permet de mettre à jour cbDest lorsque l'utilisateur choisit une unité dans cbSource :

```
Private Sub majDest()
    cbDest.Items.Clear() 'On vide la liste destination
    cbDest.Text = ""
    Dim i As Integer
    For i = 0 To cbSource.Items.Count() - 1
        If cbSource.Items.Item(i) <> cbSource.SelectedItem Then
            'On ne copie pas l'unité sélectionnée dans la liste source
            cbDest.Items.Add(cbSource.Items.Item(i))
        End If
    Next
    cbDest.Text = cbDest.Items.Item(0)
End Sub
```

Question 2: Déterminez, expliquez et corrigez les erreurs qui se sont glissées dans le code précédent.

Réponse : (1.5 points)

Les erreurs corrigées sont en rouge. (0.5 point/erreur)

Lorsque l'utilisateur modifie la valeur de texte à convertir (dans le textBox tbVal1), le programme déclenche un évènement, qui appelle la procédure suivante qui met à jour le textBox tbVal2 :

```
Private Sub doConv()
    Dim t As Double
    t = CDbl(tbVal1.Text)
    'convert(valeur à convertir, unité de départ, unité d'arrivée)
    'renvoie un Double
    tbVal2.Text = convert(t, cbSource.SelectedItem, cbDest.SelectedItem)
End Sub
```

Question 3: Un problème va apparaître si l'utilisateur saisit du texte à la place d'un nombre à convertir. Récrivez la procédure doConv() de manière à traiter cette erreur à l'aide d'une exception. Pour cela, déclarez et « lancez » une exception si le texte saisi n'est pas un nombre. Écrivez aussi le code nécessaire pour « attraper » cette exception et en notifier l'utilisateur (vous disposez pour cela d'un label nommé tssStatus).

Réponse : (3.5 points)

```
Private Sub doConv()
    Dim exPasNum As New Exception("Erreur : il faut saisir un nombre")
    Try 'Si numérique, on convertit, sinon on lance l'exception
        If IsNumeric(tbVal1.Text) Then
            tbVal2.Text = convert(tbVal1.Text, cbSource.SelectedItem,
cbDest.SelectedItem)
            tssText.Text = "Prêt" 'Facultatif
        Else
            Throw exPasNum
        End If
    Catch ex As Exception
        tssText.Text = ex.Message
    End Try
End Sub
```

Nom et Prénom :

Numéro de TP :

```
    End If
    Catch ex As Exception
        tssText.Text = "Saisie non valide !"
    End Try
End Sub
```

A. Programmation événementielle (8 pts) (5+3)

Contexte : on travaille toujours sur l'application SciConv de la partie B. De manière générale, dès que l'utilisateur tape un nombre, ou change les unités, la conversion se fait immédiatement (doConv est appelé)

L'interface est également composée d' :

- un label nommé `lblType`, contenant le texte « Type d'unité à convertir : »,
- un `comboBox` nommé `cbType` listant les types d'unités (distance, poids, volume...)
- un `comboBox` nommé `cbSource` listant les unités du type choisi dans `cbType` (m, km, ft... pour la distance par ex.)
- un `comboBox` nommé `cbDest` listant les unités du type choisi dans `cbType` (km, ft... pour la distance par ex.) moins l'unité choisie dans `cbSource`.
- un `toolStripStatusLabel` (fonctionne comme un `Label`) nommé `tssStatus`, contenant le texte « Prêt »,
- une `textBox` nommée `tbVal11`, initialisée à la chaîne "0",
- une `textBox` nommée `tbVal12`, initialisée à la chaîne "0",
- un `button` nommé `btnSwap`, contenant le texte « $\uparrow\downarrow$ ».

Question 1: Écrire la procédure événementielle qui, au clic sur le bouton `btnSwap`, échange le contenu des deux lignes (attention, on échange les unités sélectionnées et on échange le texte saisi avec le texte résultat) :

On rappelle que les `comboBox` ont un élément nommé `Items` contenant toutes les valeurs listées. Pour retrouver la position d'un item précis dont on connaît la valeur, on dispose de la méthode `IndexOf` qui prend une chaîne de caractères en entrée et renvoie un entier. Le texte de l'item sélectionné (affiché) dans une `comboBox` est dans la propriété `selectedItem` de la `comboBox`. Vous pouvez utiliser les procédures données dans la partie B en supposant qu'elles fonctionnent correctement.

Réponse : (5 points)

```
Private Sub btnSwap_Click(sender As System.Object, e As System.EventArgs) Handles btnSwap.Click
    Dim t, src, dest As String
    'Il faut inverser le texte avant, sinon on perd la valeur résultat !
    '(Inverser les unité déclenche un évènement qui recalcule la conversion)
    t = tbVal1.Text
    tbVal1.Text = tbVal2.Text
    tbVal2.Text = t
    'On inverse les unités :
    'On stocke la valeur courante des deux listes
```

Nom et Prénom :

Numéro de TP :

```
src = cbSource.SelectedItem
dest = cbDest.SelectedItem
'Puis on place le curseur de la liste source sur l'item correspondant à la
destination
cbSource.SelectedIndex = cbSource.Items.IndexOf(dest)
'On recrée la liste destination, ce qui enlève l'item qui était sélectionné en
destination et rajoute celui qui était en source
majDest()
'Puis on sélectionne le bon item, qui était la source avant
cbDest.SelectedIndex = cbDest.Items.IndexOf(src)
la conversion est déclenchée lors de l'instruction précédente, inutile de
refaire doConv()
End Sub
```

Question 2: Écrire la procédure événementielle easterEgg qui se déclenche si l'utilisateur double clique n'importe où ailleurs dans la fenêtre (donc hors des contrôles utilisés normalement par le programme) et affiche dans une boîte de dialogue un petit message humoristique tiré au hasard parmi une liste prédefinie contenue dans la variable globale lapin de type String().

Réponse : (3 points)

```
Private Sub easterEgg(sender As System.Object, e As System.EventArgs) Handles
 MyBase.DoubleClick
    Randomize() 'On initialise le générateur de nombres aléatoires
    Dim w = Int(Rnd() * 10) 'On tire un nombre entre 0 et 9
    MsgBox(lapin(w), , "Bravo !") 'Et on affiche le message
End Sub
```