

## TRAVAUX DIRIGES Visual Basic DEUST - 1

H. TSOUNGUI

### Exo-1 Programme de calcul de TVA

Pour ce premier exercice, il vous est proposé un formulaire. Par la suite, vous concevrez vous-mêmes vos propres interfaces.

Ce formulaire comporte des champs de saisie de valeurs (montant hors taxes, taux de TVA) ainsi qu'un premier bouton de commande permettant de déclencher une procédure de calcul et affichage des résultats. Deux derniers boutons permettent de remettre à « blanc » toutes les zones de saisie ou de fermer le formulaire.

1-Ecrire le code de chacun des boutons de commande de l'interface.

### Exo-2 Types de données complexes : vecteurs (tableaux à 1 dimension) - Structures de contrôle

-Ecrire un programme VB permettant de manipuler un vecteur (tableau à une dimension) d'entiers.

- 1-Déclarer le vecteur V de 5 nombres entiers
- 2-Saisir les 5 valeurs numériques du tableau avec **InputBox**
- 3-Afficher les valeurs saisies dans une liste (**listbox**).
- 4-Rechercher la plus petite valeur ainsi que la plus grande.
- 5-Calculer la somme des éléments du tableau
- 6-Calculer la moyenne des éléments du tableau

### Exo-3

- Refaire l'*exo-2* en utilisant la fonction de génération des nombres aléatoires **RND()** pour automatiser l'obtention des valeurs du tableau à une dimension.
- Remplir un vecteur de valeurs tirées au hasard entre 1 et 9.
- Lire une valeur à rechercher les occurrences dans le vecteur par exemple 2.
- Ecrire le code donnant le nombre de fois qu'apparaît cette valeur dans le vecteur.

### Exo-4 Tableaux à plusieurs dimensions (matrices) – structures de contrôle

-Manipulation d'un tableau à plusieurs dimensions

Tableau des clients **CLIENTS (4 lignes x 4 colonnes)**

Numéro-client	Nom-client	Prénom-client	Chiffre-affaires
C1	DEMON	Jean	945.34
C2	MOULOUD	Aziz	2304.85
C3	MARTIN	David	4963.57
C4	LEROUGE	Alain	3945.34

- 1-Déclarer dans un **module** un tableau CLIENTS(4,4) de chaînes ;
- 2-Saisir les données ci-dessus par client ;
- 3-Afficher les données de ce tableau dans un formulaire ou une liste déroulante ;
- 4-Créer une interface et écrire le code permettant de rechercher un client et d'afficher ses informations.
- 5-Rechercher le meilleur client et afficher ses infos (Numéro, Nom et Chiffre d'affaires).

Variations sur le thème

#### ---suite guide de résolution

-Manipulation d'un tableau à plusieurs dimensions

Tableau des clients CLIENTS (4 lignes x 4 colonnes)

- 1-Déclarer dans un **module** un tableau CLIENTS(4,4) de chaînes ;

Module **Modmatrice**

```
Public clients(4, 4) As String
End Module
```

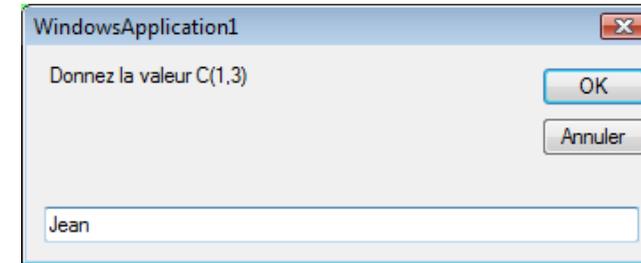
2-Saisir les données ci-dessus par client ;



- Quand on clique sur le bouton « LIRE », il faut saisir les données dans deux boucles imbriquées

```
Private Sub Button1_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Button1.Click
    Dim i As Integer : Dim j As Integer
    'Lecture des valeurs de la matrice
    For i = 1 To 4
        For j = 1 To 4
            clients(i, j) = InputBox("Donnez la
valeur C(" & i & "," & j & ")")
        Next j
    Next i
    'Affichage des valeurs lues
    l11.Text = clients(1, 1) : l12.Text = clients(1,
2) : l13.Text = clients(1, 3) : l14.Text = clients(1, 4)
    ... ..
End Sub
```

NB : On peut aussi initialiser les éléments du tableau une fois pour toutes avec des affectations du genre suivant  
clients(1, 1) = « C1 » : clients(1, 2) = « DEMON » :  
clients(1, 2) = ...



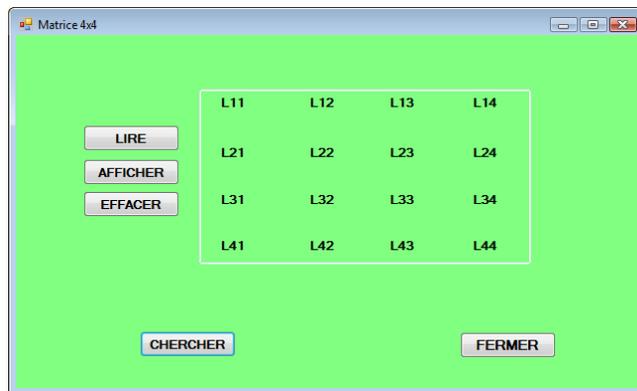
- 3-Ensuite, il faut afficher les valeurs lues dans les labels qui ont pour nom (propriété name L11, L12 tout en minuscules)

```
'Affichage des valeurs lues
    l11.Text = clients(1, 1) : l12.Text = clients(1,
2)
    l13.Text = clients(1, 3) : l14.Text = clients(1,
4)
    ... ..
```

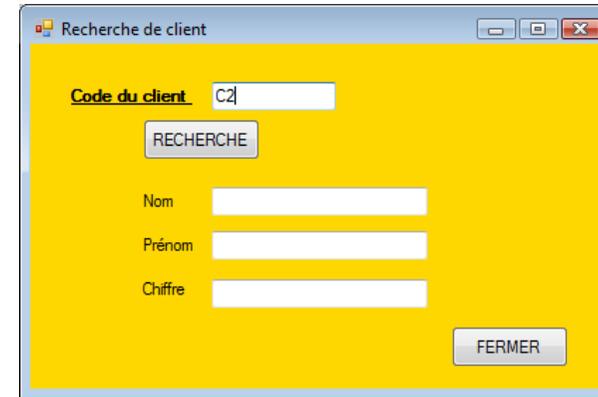


4-Créer une interface et écrire le code permettant de rechercher un client et d'afficher ses informations.

On ajoute un bouton « CHERCHER » dans le premier formulaire



En cliquant dessus, le formulaire suivant s'ouvre :



On écrit le code de la recherche du bouton « RECHERCHE » :

```

..... Algorithmme .....
Il faut parcourir les client(i,1) 'parcours des lignes de la colonne 1
comparer les codes, etc.
Si égalité entre le code client saisi et la valeur de client(i,1),
on a trouvé,
alors on sait que le client cherché est sur la ligne k=i ;
Il suffit alors d'afficher client(k,1) pour son code, client(k,2) pour son
nom, client(k,3) son prénom et
Client (k,4) pour son chiffre d'affaires.
Fin du parcours
    
```

5-Rechercher le meilleur client et afficher ses infos (Numéro, Nom et Chiffre d'affaires).

```

..... Algorithmme .....
On fixe comme meilleur client le premier de la liste ; puis on parcourt les autres
lignes à partir de i=2 dans une boucle
Pour i=2 à 4 ,
on compare le chiffre d'affaires avec le meilleur du moment
si on trouve plus grand,
on refixe le « nouveau meilleur » en prenant soin de récupérer la
position kpos = i.
Fin Pour ;
    
```

-La procédure est la même si on veut trouver le « plus mauvais » client.  
Les mauvais clients sont ceux qui auraient un chiffre négatif ...

**Exo-5 Gestion simplifiée de stock de produits**

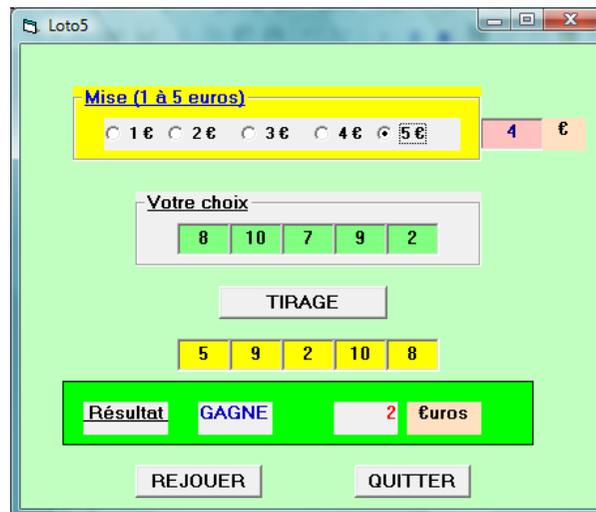
De manière similaire à l'exo précédent, cet exercice propose de gérer un stock de marchandises. Il s'agit d'utiliser un tableau PRODUITS(Référence, Libellé, Prix-HT, Quantité-en-stock), limité à 7 produits.

Créer les interfaces et écrire le code permettant de gérer le stock :

- 1-Entrée des produits
- 2-Sortie des produits
- 3-Inventaire du stock
- 4-Recherche d'un produit
- 5-Afficher la valeur du stock actuel
- 6-Saisie d'une commande et édition d'une facture

**Exo-6 Jeu Loto-5 Simulation du jeu de loterie**

Développer un programme en VB simulant une loterie à cinq nombres. Après avoir effectué une mise de **5 euros** au plus, on propose 5 nombres compris entre 1 et 10. On clique sur un bouton pour tirer au hasard cinq nombres **différents** dans le même intervalle. Chaque tirage diminue la mise de 1 euro.



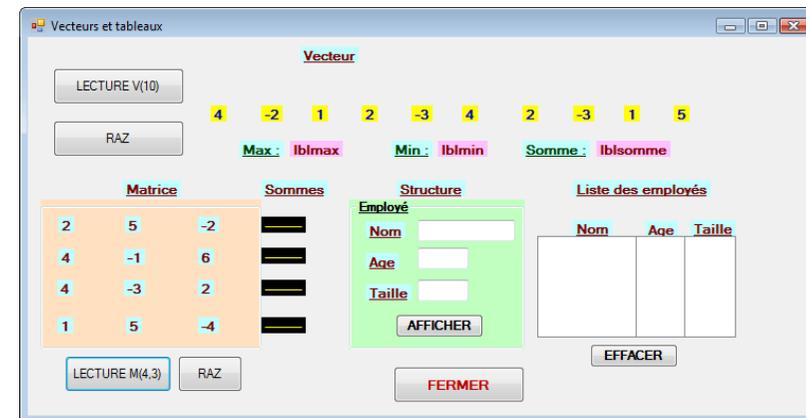
-Afficher le résultat du tirage et gérer les gains. On peut jouer tant que la mise est positive.

-Gestion des gains :

- \*Si le joueur a trouvé 1 ou 2 nombres, il perd 1 euro de mise (gain : 0 euro).
- \*Trois nombres trouvés, gain de 2 euros
- \*Quatre nombres trouvés, gain de 7 euros
- \*Cinq nombres trouvés, gain de 10 euros

**Exo-7**

Utilisation des types de données complexes : vecteurs, tableaux à plusieurs dimensions et structures de données (records).



Reproduire le formulaire et écrire le code nécessaire pour les boutons de commande.

**Exo-8 Utilisation de types structurés - Gestion des clients**

Description de la structure à utiliser (à déclarer dans un module) :

```
Module modstruct 'Module public
  Public Structure client 'Déclaration de type
    <VBFixedString(3)> Public cliNum As String
    <VBFixedString(15)> Public cliNom As String
    <VBFixedString(15)> Public cliPrenom As String
    <VBFixedString(20)> Public cliAdresse As String
    <VBFixedString(5)> Public cliPostal As String
    <VBFixedString(15)> Public cliVille As String
    <VBFixedString(8)> Public cliChiffre As String
  End Structure
End Module
```

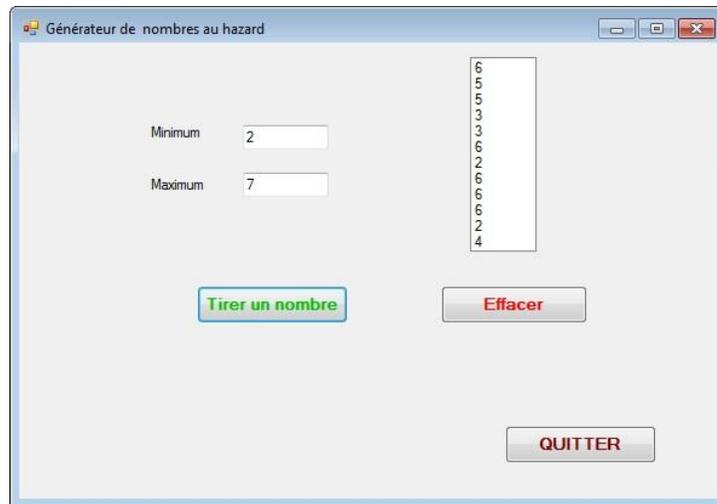
**Affichage des données dans un contrôle listBox**

```
Private Sub btnAfficher_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
btnAfficher.Click
  'Ajout dans la liste des clients
  Dim vcli As client
  Dim ligne As String

  With vcli
    .cliNum = txtNum.Text
    .cliNom = txtNom.Text
    .cliPrenom = txtPrenom.Text
    .cliAdresse = txtAdresse.Text
    .cliPostal = txtCpostal.Text
    .cliVille = txtVille.Text
    .cliChiffre = txtChiffre.Text
  End With
  ligne = vcli.cliNum + " - " + vcli.cliNom + " - " +
vcli.cliPrenom + " - " + vcli.cliPostal + _
" - " + vcli.cliVille + " - " + vcli.cliChiffre
  lstClients.Items.Add(ligne)
End Sub
```

**Autre exemple : gestion des étudiants**
**Exo-9**

Ecrire un programme permettant de générer des nombres aléatoires compris dans un intervalle choisi (entre les valeurs min et max). Utiliser l'interface graphique ci-dessous pour afficher les nombres générés.



**Exo-10** Simulation du jeu de Jackpot version légère 1.0 (un seul joueur)



Le jeu : on clique sur « JOUER ». Le système tire au hasard 3 nombres X, Y et Z compris entre 1 et 9. Le premier nombre concerne la première roue, le deuxième la

seconde et le troisième, la troisième roue. On dispose de 3 images  $img1$ ,  $img2$  et  $img3$ .  
 -Si le nombre tiré est dans [1-2-3], on affiche l'image  $img1$  pour la roue concernée.  
 -Si le nombre tiré est dans [4-5-6], on affiche l'image  $img2$  pour la roue concernée.  
 -Si le nombre tiré est dans [7-8-9], on affiche l'image  $img3$  pour la roue concernée.  
 Les gains :  
 -si on obtient trois images différentes => on perd (gain de 0 euro)  
 -si on obtient seulement deux images successives identiques ( $img1$ - $img1$ -W) ou ( $W$ - $img2$ - $img2$ ) ou ( $img3$ - $img3$ -W), W étant différente de l'image à côté => gain de 5 euros.  
 -si on obtient trois images identiques, deux cas peuvent se présenter  
 -les trois nombres générés sont différents => c'est un BANCO : gain de 10 euros.  
 -les trois nombres générés sont identiques => c'est un SUPER-BANCO : gain de 20 € !  
 Développez ce jeu en respectant les règles proposées.

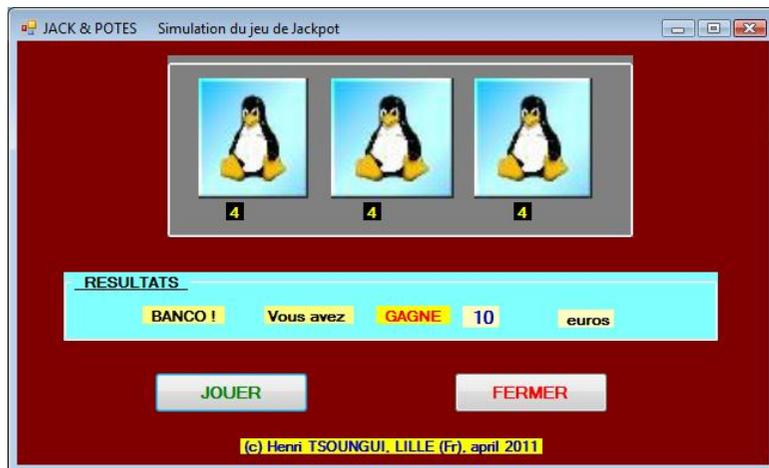
**Exemples d'illustrations de situations**



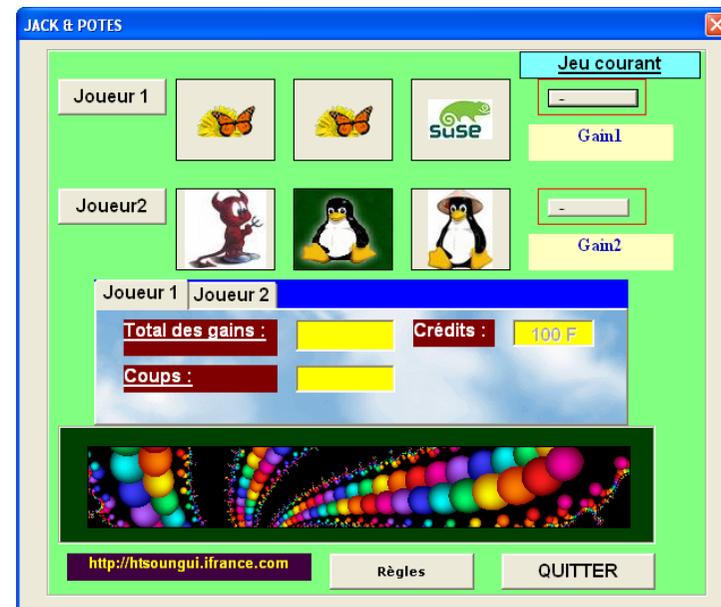
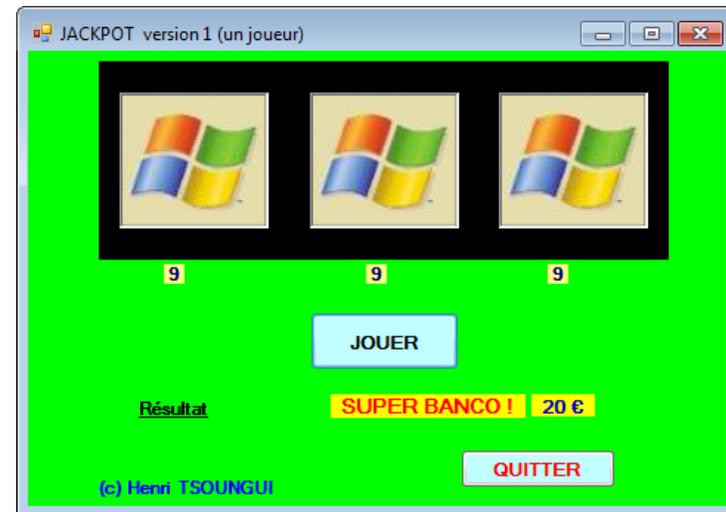
Gain simple (5 €)



BANCO ! (10€)



SUPER-BANCO ! (à mettre à 20 €)



**Exo-11 JACK&POTES Simulation du jeu de jackpot version 2.0**

Développer une application en Visual Basic utilisant les fichiers séquentiels à accès direct pour stocker les gains.

**Contraintes**

- L'application doit permettre à au moins un joueur de tenter sa chance.
- Pour jouer, tout joueur doit miser une somme minimale. Le jeu ne démarre que si la mise est suffisante.
- Pour gagner on doit avoir obtenu au moins deux *images successives identiques* :
  - 2 images successives identiques => gain de 10 euros
  - 3 images identiques => gain de 50 euros (Banco)
- La génération de *sons* est obligatoire pour distinguer les résultats.
- Les résultats des joueurs (Nom et gain total) seront sauvés dans un *fichier séquentiel* à accès direct et un bouton de commande permettra de les visualiser dans l'ordre croissant des gains (Tri des résultats).

**Exo-12 Gestion des véhicules reçus dans un garage (extrait DS DEUST - 1)**

Dans cet exercice, il vous est demandé de gérer des voitures qui entrent dans un garage pour qu'on y effectue certains travaux. Vous devrez également gérer la création et la suppression des clients. Vous développerez donc

- la création, la mise à jour et la suppression des véhicules
- la création et la suppression des clients

La première partie utilise le type utilisateur (structure) **voiture** déclaré dans le module **ModGarage** ainsi que le formulaire **frmVoiture** ci-dessous dont les champs de saisie sont nommés txtImmat, txtMarque, txtModèle, txtEnergie, txtTravail1, txtTravail2, txtClient. Faire de même pour la gestion des clients.



Ci-dessus, le formulaire **frmGarage** à améliorer avec des **menus déroulants**

Ci-dessus, le formulaire **frmVoiture**.

Les structures de données à utiliser sont décrites ci-après.

Module Modgarage

```
Public Structure client
  <VBFixedString(4)> Public cliNum As String
  <VBFixedString(15)> Public cliNom As String
  <VBFixedString(15)> Public cliPrenom As String
  <VBFixedString(25)> Public cliAdresse As String
  <VBFixedString(5)> Public cliCpostal As String
  <VBFixedString(20)> Public cliVille As String
  <VBFixedString(10)> Public cliTelfixe As String
  <VBFixedString(10)> Public cliTelmob As String
End Structure
```

```

Public Structure voiture
    <VBFixedString(8)> Public voitImmat As String
    <VBFixedString(15)> Public voitMarque As String
    <VBFixedString(10)> Public voitModele As String
    <VBFixedString(12)> Public voitEnergie As
String
    <VBFixedString(60)> Public voitTravail1 As
String
    <VBFixedString(60)> Public voitTravail2 As
String
    <VBFixedString(4)> Public voitClient As String
End Structure
End Module

```

NB : cet exercice peut être fait en utilisant une petite base de données ACCESS ou SQLServer et **ADO.NET**.

The screenshot shows a window titled "Fiche CLIENT" with a light blue background. It contains several text input fields arranged in two columns. The fields are: Numéro (C65), Nom (DARGAUD), Prénom (Frédéric), Adresse (43, rue de la digue), Code postal (59300), Ville (Valenciennes), Tél. fixe (0327895412), and Tél. mobile (0754670912). At the bottom, there are three buttons: SAUVER, EFFACER, and QUITTER.

The screenshot shows a window titled "Fiche CLIENT" with a light blue background. It contains several text input fields arranged in two columns. The fields are: Numéro (C58), Nom (ZAKID), Prénom (Hamid), Adresse (5, place de la république), Code postal (59000), Ville (Lille), Tél. fixe (0320864523), and Tél. mobile (0689432022). At the bottom, there are three buttons: SAUVER, EFFACER, and QUITTER.

### Exemples d'exécution du Jackpot simple

The screenshot shows a window titled "Jackpot version 1 (un joueur)" with a green background. It features a 3x3 grid of images: a blue hat, the Windows logo, and a penguin. Below the grid are the numbers 2, 9, and 4. A "JOUER" button is centered below the grid. Below the "JOUER" button, the text "Résultat" is followed by "PERDU!" in a yellow box and "0" in a yellow box. At the bottom left, it says "(c) Henri TSOUNGUI", and at the bottom right, there is a "QUITTER" button.

