

La syntaxe de VISUAL BASIC (partie 2)

10°) : La structure répétitive simple ITER

SYNTAXE :

```
DO  
    instruction  
    .....  
    IF condition THEN  
        EXIT DO  
    END IF  
    instruction  
    .....  
LOOP
```

La structure répétitive simple suppose de gérer la condition de sortie de la boucle à l'aide de l'instruction EXIT DO.

11°) : La structure répétitive ITER "test en tête : TantQue..FinTant Que"

SYNTAXE :

```
DO WHILE condition  
    instruction  
    .....  
    instruction  
    .....  
LOOP
```

Ou bien

SYNTAXE :

```
WHILE condition  
    instruction  
    .....  
    instruction  
    .....  
WEND
```

L'instruction While...Wend comprend les éléments suivants :

condition : Expression booléenne qui prend la valeur True ou False.

instruction : une ou plusieurs instructions exécutées tant que la condition est vraie.

Les boucles While...Wend peuvent être imbriquées à n'importe quel niveau.

Chaque instruction Wend correspond à l'instruction While la plus récente.

12°) : La structure répétitive ITER "test en queue : Répéter...Jusqu'à"

SYNTAXE:

```
DO  
    instruction  
    .....  
    instruction  
    .....  
LOOP UNTIL/WHILE condition
```

Do Doit être la première instruction dans la structure de contrôle Do...Loop.

While Indique que la boucle s'exécutera tant que condition est vraie.

Until Indique que la boucle s'exécutera jusqu'à ce que condition devienne vraie.

13°) : La structure répétitive POUR

SYNTAXE:

```
FOR compteur = debut TO fin [STEP pas]  
    instruction  
    .....  
    instruction  
    .....  
NEXT compteur
```

L'instruction For...Next comprend les éléments suivants :

compteur variable numérique utilisée comme compteur de boucle. Cette variable ne peut pas être un élément d'un type défini par l'utilisateur.

debut valeur initiale de l'argument compteur.

fin valeur finale de l'argument compteur.

pas valeur ajoutée au compteur à chaque passage dans la boucle. Si rien n'est précisé, la valeur de l'argument par défaut est un. Elle peut être positive ou négative.

instruction une ou plusieurs lignes de code entre les instructions For et Next qui sont exécutées un nombre donné de fois.

14°) : Génération des nombres aléatoires

a°) Initialisation du générateur de nombres aléatoires.

SYNTAXE :

RANDOMIZE [nombre]

L'argument nommé nombre peut être n'importe quelle expression numérique valide.

L'instruction Randomize utilise l'argument nombre pour initialiser le générateur de nombres aléatoires de la fonction Rnd, en lui donnant une nouvelle valeur initiale. Si l'argument nombre est omis, c'est la valeur renvoyée par la minuterie du système qui est prise comme nouvelle valeur initiale.

Si on n'utilise pas Randomize, la fonction Rnd (sans argument) utilise, la première fois qu'elle est appelée, le même nombre comme valeur initiale.

Ensuite, c'est le dernier nombre généré qui est utilisé comme valeur initiale.

b°) Renvoie un nombre aléatoire.

SYNTAXE :

RND [(nombre)]

L'argument nommé nombre peut être n'importe quelle expression numérique valide.

La fonction Rnd renvoie une valeur inférieure à 1 mais supérieure ou égale à 0. La valeur de l'argument nombre détermine la manière dont Rnd génère un nombre aléatoire.

Pour obtenir des entiers aléatoires dans un intervalle donné, utilisez la formule suivante :

$$\text{Int}((\text{bornesupérieure} - \text{borneinférieure} + 1) * \text{Rnd} + \text{borneinférieure})$$

Dans cette formule, bornesupérieure désigne la limite supérieure de l'intervalle, et borneinférieure la limite inférieure.

Exercice N°7

On veut écrire le programme console FACTORIEL qui calcule le factoriel d'un nombre en utilisant les cinq boucles suivantes :

- * La boucle POUR
- * La boucle DO WHILE... LOOP
- * La boucle DO ... LOOP WHILE
- * La boucle DO ... LOOP UNTIL
- * La boucle DO ... EXIT DO ... LOOP

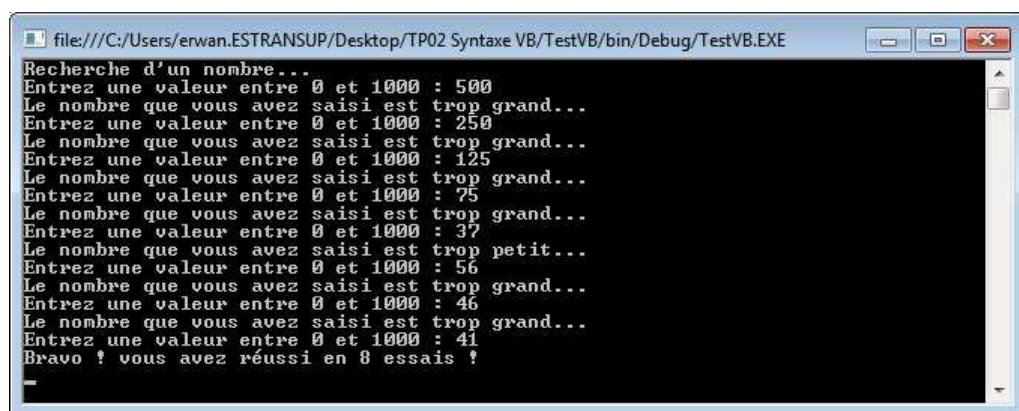
Exercice N°8

Ecrire un programme console qui permet de trouver un nombre compris entre 0 et 1000 (déterminé aléatoirement par l'ordinateur) en ayant comme indication :

- trop grand ou
- trop petit

A la fin on doit donner le nombre d'essais qui ont été nécessaires pour trouver la bonne réponse.

exemple :



```
file:///C:/Users/erwan.STRANSUP/Desktop/TP02 Syntaxe VB/TestVB/bin/Debug/TestVB.EXE
Recherche d'un nombre...
Entrez une valeur entre 0 et 1000 : 500
Le nombre que vous avez saisi est trop grand...
Entrez une valeur entre 0 et 1000 : 250
Le nombre que vous avez saisi est trop grand...
Entrez une valeur entre 0 et 1000 : 125
Le nombre que vous avez saisi est trop grand...
Entrez une valeur entre 0 et 1000 : 75
Le nombre que vous avez saisi est trop grand...
Entrez une valeur entre 0 et 1000 : 37
Le nombre que vous avez saisi est trop petit...
Entrez une valeur entre 0 et 1000 : 56
Le nombre que vous avez saisi est trop grand...
Entrez une valeur entre 0 et 1000 : 46
Le nombre que vous avez saisi est trop grand...
Entrez une valeur entre 0 et 1000 : 41
Bravo ! vous avez réussi en 8 essais !
```

Amélioration de l'IHM :

Afin de rendre votre programme plus convivial ajouter des couleurs et éviter d'avoir trop de lignes d'affichage.

Veille technologique :

On veut mesurer le temps que met le joueur pour trouver le nombre mystère, rechercher sur internet un code source permettant de mesurer des temps puis inclure ce code dans votre programme.

Exercice N°9

On veut écrire le programme console CaisseEnregistreuse qui simule l'utilisation d'une caisse enregistreuse.

Les ressources suivantes sont données :

- Executable de la version 4
- Le fichier Version.txt
- L'aide sur les tableaux de structures
- Le fichier de test

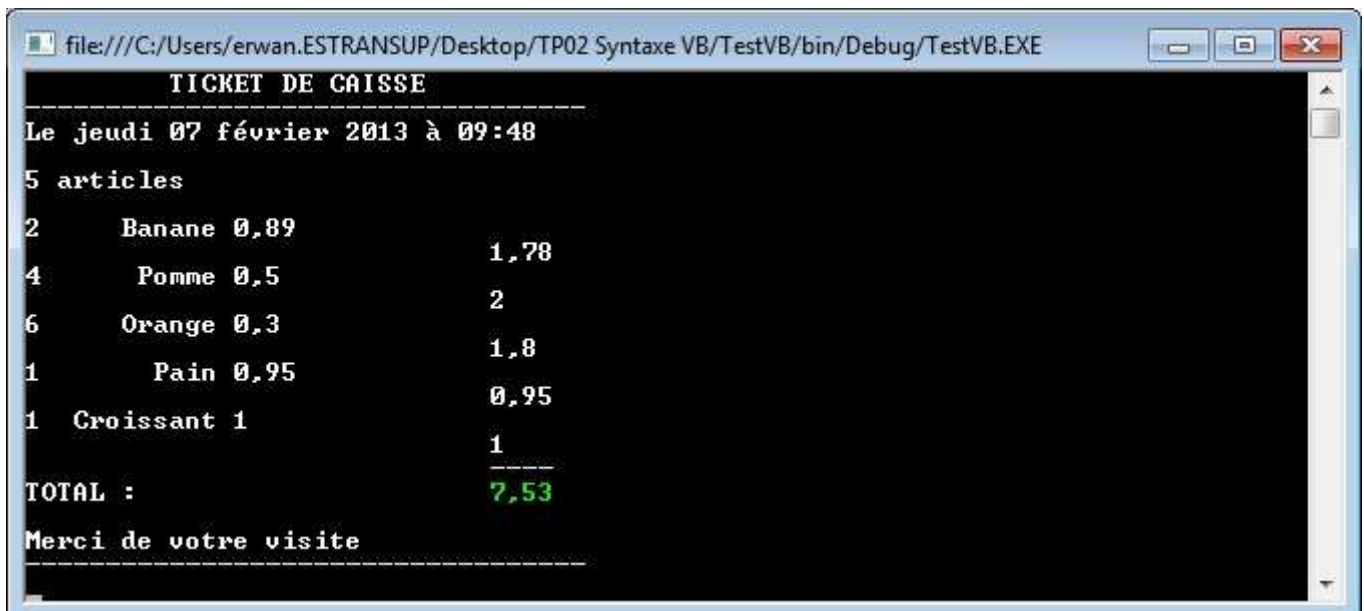
SAISIE DES INFORMATIONS :



```
file:///C:/Users/erwan.STRANSUP/Desktop/TP02 Syntaxe VB/TestVB/bin/Debug/TestVB.EXE
1 - Banane
2 - Pomme
3 - Orange
4 - Pain
5 - Croissant

Numero du produit <1..5> 6=Exit : _
```

AFFICHAGE DU TICKET DE CAISSE :



```
file:///C:/Users/erwan.STRANSUP/Desktop/TP02 Syntaxe VB/TestVB/bin/Debug/TestVB.EXE
TICKET DE CAISSE
-----
Le jeudi 07 février 2013 à 09:48
5 articles
2    Banane 0,89      1,78
4    Pomme 0,5        2
6    Orange 0,3       1,8
1    Pain 0,95        0,95
1    Croissant 1      1
TOTAL :              7,53
Merci de votre visite
-----
```

Veille technologique :

Comment fonctionne les tableaux de structures en VB.NET