

EXERCICES

RESOLUS

en

VISUAL BASIC

Aimé DIUMI DIKOLO

www.wiscorp.com

INTRODUCTION

De nos jours, l'informatique touche presque tous les domaines de la vie. Qu'on soit étudiant, professeur, juriste, médecin, politicien etc., il est difficile de vivre sans utiliser directement ou indirectement un produit de l'informatique. L'élément central de l'informatique est l'ordinateur.

Mais l'ordinateur n'est pas intelligent comme vous et moi, il ne sait rien faire de lui-même, à tout moment il ne fait qu'obéir aux ordres. Pour effectuer une tâche quelconque, l'ordinateur suit une suite d'instructions qu'on appelle programmation informatique.

Pour donner des ordres à l'ordinateur, soit pour écrire des programmes informatiques, on utilise ce qu'on appelle un langage de programmation. Actuellement, il existe plusieurs langages de programmation, nous pouvons citer : Python, Java, Visual basic, C#, Pascal, PHP, JavaScript, Ada etc.

Nous vous proposons 40 exercices résolus en Visual basic, certaines questions sont accompagnées d'une analyse avant la résolution. On fait la programmation pour trouver des solutions à des problèmes donnés, et un problème peut avoir plusieurs solutions, de même, les solutions proposées ici ne sont pas uniques ni les meilleures. Toute suggestion, remarque ou critique sera bien reçue et prise en considération.

Bonne lecture.

EXERCICE 1

Ecrire un programme en Visual basic mode console qui demande un nombre à l'utilisateur, puis l'informe si ce nombre est entier ou décimal.

Analyse et résolution

On va utiliser une structure conditionnelle car tout nombre n'est pas entier, il y a une ou des conditions pour un être entier. Et comment trouver cette condition ? On sait qu'un nombre est entier s'il n'a pas de partie décimale, c'est-à-dire si la partie décimale vaut zéro. Donc la condition est que la partie décimale donne zéro. Mais comment récupérer la partie décimale d'un nombre ?

En parcourant les fonctions mathématiques de Visual basic, il n'y a pas de fonctions permettant de récupérer la partie décimale d'un nombre et pourtant il y a la fonction Fix qui renvoie plutôt la partie entière d'un nombre. Que faire ???

Supposons que le nombre saisi par l'utilisateur est a, on peut récupérer la partie entière de a dans la variable b et si on fait la différence entre a et b, on trouve la partie décimale de a : voilà une solution.

Code VB.Net

```
Module Module1
```

```
Sub Main()  
    Dim n, deci As Single  
    Dim entier As Integer  
    Console.WriteLine("Entrez un nombre")  
    n = Val(Console.ReadLine())  
    'on récupère la partie entière  
    entier = Fix(n)  
    'on trouve la partie décimale  
    deci = n - entier  
    'test  
    If deci = 0 Then  
        Console.WriteLine("Le nombre " & n & "est un entier")  
    Else  
        Console.WriteLine("Le nombre " & n & "est un nombre décimal")  
    End If  
    Console.ReadLine()
```

```
End Sub
```

```
End Module
```

EXERCICE 2

Le calcul de salaire net (sn) des agents d'une entreprise est constitué des rubriques suivantes : salaire de base (sb) et prime enfant (pe).

Le salaire de base est déterminé à partir du grade suivant le tableau :

Le salaire de base est déterminé à partir du grade suivant le tableau :

Grade	Salaire de base (en \$US)
D	2700
SD	2400
CS	2100
CB	1750
ATB2	1500
ATB1	1350

La prime enfant qui est calculée en fonction du nombre d'enfants en vie (nef), de la manière suivante : du 1er au 5e enfant chacun génère une prime fixe de 50\$US, à partir du 6e enfant, la prime devient de 35\$US par enfant et enfin pas de prime à partir du 11e enfant.

Ecrire un programme en Visual basic en mode console qui calcule le salaire net à payer à un agent donné et disposant d'un nombre donné d'enfants.

Analyse et solution

Pour résoudre ce problème, l'utilisateur doit nous fournir le grade (grade) ainsi que le nombre d'enfants en vie (nef).

On voit clairement que le salaire de base dépend du grade de l'agent, ainsi donc on va utiliser des structures conditionnelles pour savoir s'il s'agit de quel grade enfin de lui attribuer le salaire de base. Et pour la prime enfant, on va utiliser des conditions sur le nombre d'enfants.

Code VB.Net

Module Module1

```
Sub Main()  
    Dim sn, sb, nef, pe As Integer  
    Dim grade As String  
    Console.WriteLine("Entrez le grade de l'agent")  
5:    grade = Console.ReadLine()  
    Console.WriteLine("Entrez le nombre d'enfant")  
10:   nef = Val(Console.Read())  
    'détermination du salaire de base  
    If UCase(grade) = "D" Then  
        sb = 2700
```

```

ElseIf UCase(grade) = "SD" Then
    sb = 2400
ElseIf UCase(grade) = "CS" Then
    sb = 2100
ElseIf UCase(grade) = "CB" Then
    sb = 1750
ElseIf UCase(grade) = "ATB2" Then
    sb = 1500
ElseIf UCase(grade) = "ATB1" Then
    sb = 1350
Else
    Console.WriteLine("Grade non reconnu, veuillez recommencer")
    GoTo 5
End If
'détermination de la prime enfant
If nef >= 1 And nef <= 5 Then
    pe = nef * 50
ElseIf nef >= 6 And nef <= 10 Then
    pe = 5 * 50 + (nef - 5) * 35
ElseIf nef >= 11 Then
    pe = 5 * 50 + 5 * 35
Else
    Console.WriteLine("Nombre erroné d'enfants, veuillez recommencer")
    GoTo 10
End If
sn = sb + pe
Console.WriteLine("Le salaire net vaut: " & sn & "$US")
Console.ReadLine()
End Sub

End Module

```

EXERCICE 3

Un groupe scolaire de la place organise le paiement de ses frais par enfant de la manière suivante :

Catégorie	Frais scolaires en FC	Frais de l'Etat en FC
Autres	80.000	5.000
Agent GS	*	5.000

* Pour la catégorie « Agent GS », le parent payera la moitié des frais scolaires pour chacun de ses trois enfants, le $\frac{3}{4}$ des frais scolaires pour le 4e et le 5e enfant et la totalité des frais scolaires pour chaque enfant à partir du 6e.

Ecrire un programme en Visual basic en mode console qui calcule les frais à payer pour un parent d'une catégorie quelconque qui a un nombre donné d'enfants.

Analyse et solution

Pour résoudre ce problème, l'utilisateur doit fournir deux informations : la catégorie du parent ainsi que le nombre d'enfants. Les frais scolaires dépendent de la catégorie du parent et du nombre d'enfants, dans ce cas c'est important d'utiliser une structure conditionnelle.

Code VB.Net

```
Module Module1
```

```
Sub Main()  
    Dim categorie As String  
    Dim nef, fraisEtat, fraisScolaire, total As Integer  
    Console.WriteLine("Entrez la catégorie du parent")  
1:   categorie = Console.ReadLine()  
    Console.WriteLine("Entrez le nombre d'enfants")  
2:   nef = Val(Console.ReadLine())  
    If UCase(categorie) = "AUTRES" Then  
        If nef <= 0 Then  
            Console.WriteLine("Le nombre d'enfants est incorrect, recommencez")  
            GoTo 2  
        End If  
        fraisScolaire = nef * 80000  
    ElseIf UCase(categorie) = "AGENT GS" Then  
        Select Case nef  
            Case 1 To 3  
                fraisScolaire = nef * 40000  
            Case 4 To 5  
                fraisScolaire = 3 * 40000 + (nef - 3) * 60000  
            Case Is > 5  
                fraisScolaire = 3 * 40000 + 2 * 60000 + (nef - 5) * 80000  
            Case Else  
                Console.WriteLine("Nombre d'enfants incorrect, recommencez")  
                GoTo 2  
        End Select  
    Else  
        Console.WriteLine("Catégorie incorrecte, recommencez")  
        GoTo 1  
    End If  
    fraisEtat = nef * 5000  
    total = fraisEtat + fraisScolaire  
    Console.WriteLine("Les frais scolaires à payer: " & fraisScolaire)  
    Console.WriteLine("Les frais de l'Etat: " & fraisEtat)  
    Console.WriteLine("Le total : " & total)  
    Console.ReadLine()  
End Sub
```

```
End Module
```

EXERCICE 4

Ecrire un programme en Visual basic mode console qui demande un nombre à l'utilisateur puis détermine si ce nombre est pair ou impair.

Analyse et solution

Pour ce problème, l'utilisateur doit entrer un nombre, et notre programme doit déterminer si c'est un nombre pair ou impair ; on sait qu'un nombre pair c'est un nombre qui est divisible par deux, et un nombre est divisible par deux lorsque le reste de la division de ce nombre par deux donne 0.

Code VB.Net

Module Module1

```
Sub Main()  
    Dim n As Integer  
    Console.WriteLine("Entrez un nombre entier")  
    n = Val(Console.ReadLine())  
    If n Mod 2 = 0 Then  
        Console.WriteLine("Le nombre " & n & "est pair")  
    Else  
        Console.WriteLine("Le nombre " & n & "est impaire")  
    End If  
    Console.ReadLine()  
End Sub
```

End Module

EXERCICE 5

Deux nombres entiers n et m sont qualifiés d'amis, si la somme des diviseurs de n est égale à m et la somme de diviseurs de m est égale à n (on ne compte pas le nombre lui-même et 1).

Exemple : les nombres 48 et 75 sont deux nombres amis puisque

Les diviseurs de 48 sont 2, 3, 4, 6, 8, 12, 16,24 et

$$2+3+4+6+8+12+16+24=75$$

Les diviseurs de 75 sont 3, 5, 15,25 et $3+5+15+25=48$

Ecrire un programme en Visual basic en mode console qui permet de déterminer si deux entiers n et m sont amis ou non.

Analyse et solution

Premièrement on doit chercher les diviseurs de n et m , et nous savons qu'un entier i est diviseur de n ssi le reste de la division de n par i égale 0, c'est-à-dire ssi $n \text{ Mod } i=0$.

Autre chose, les diviseurs d'un nombre ne peuvent pas dépasser ce nombre, c'est-à-dire les diviseurs d'un nombre sont toujours inférieurs ou égaux à ce nombre. Ainsi on peut parcourir les entiers allant de 1 à ce nombre qui vérifient la condition c'est-à-dire qui sont diviseurs....

Code VB.Net

Module Module1

```
Sub Main()
    Dim n, m, i, somme_n, somme_m As Integer
    Console.WriteLine("Entrez le premier nombre")
    n = CInt(Console.ReadLine())
    Console.WriteLine("Entrez le deuxième nombre")
    m = CInt(Console.ReadLine())
    'recherche et sommation des diviseurs de m
    somme_m = 0
    For i = 2 To m - 1
        If m Mod i = 0 Then
            somme_m += i
        End If
    Next
    'recherche et sommation des diviseurs de n
    somme_n = 0
    For i = 2 To n - 1
        If n Mod i = 0 Then
            somme_n += i
        End If
    Next
    'test
    If somme_m = n And somme_n = m Then
        Console.WriteLine("Les deux nombres sont amis")
    Else
        Console.WriteLine("Les deux nombres ne sont pas amis")
    End If
    Console.ReadLine()
End Sub
```

End Module

EXERCICE 6

Ecrire un programme en Visual basic mode console qui lit un nombre quelconque de valeurs entières en déterminant la somme des valeurs positives et la somme des valeurs négatives. Avec comme hypothèse qu'aucune de ces valeurs ne peut être nulle et que l'utilisateur introduira la valeur 0 pour signaler qu'il n'a plus de valeurs à fournir. Adapter le programme de manière qu'il fournisse la moyenne des valeurs négatives et la moyenne des valeurs positives (Nota : tenir compte de la protection contre le risque de division par zéro).

Analyse et solution

Selon l'énoncé, la saisie des valeurs va s'arrêter si l'utilisateur saisit zéro, et comme nous ne connaissons pas le nombre de valeurs non nulles que l'utilisateur saisira, on peut utiliser la boucle While ou Until.

A chaque fois que l'utilisateur saisit une valeur, on vérifie si elle est positive ou négative, puis on l'ajoute à la somme des valeurs positives ou négatives selon le cas et incrémenter les variables chargées de compter le nombre des valeurs positives et négatives....

Code VB.Net

Module Module1

```
Sub Main()  
    Dim n, somme_pos, somme_neg, taille_pos, taille_neg As Integer  
    Dim moyenne_pos, moyenne_neg As Single  
    somme_neg = 0  
    somme_pos = 0  
    taille_neg = 0  
    taille_pos = 0  
    Do  
        Console.WriteLine("Entrez une valeur")  
        n = CInt(Console.ReadLine())  
        If n > 0 Then  
            somme_pos += n  
            taille_pos += 1  
        ElseIf n < 0 Then  
            somme_neg += n  
            taille_neg += 1  
        End If  
    Loop Until n = 0  
    'calcul de la moyenne des valeurs positives  
  
    If taille_pos = 0 Then  
        moyenne_pos = 0  
    Else
```

```

        moyenne_pos = somme_pos / taille_pos
    End If
    'calcul de la moyenne des valeurs negatives

    If taille_neg = 0 Then
        moyenne_neg = 0
    Else
        moyenne_neg = somme_neg / taille_neg
    End If
    'Affichage du resultat
    Console.WriteLine("La somme des valeurs positives vaut : " &
somme_pos)
    Console.WriteLine("La moyenne des valeurs positives vaut: " &
moyenne_pos)
    Console.WriteLine("La somme des valeurs negatives vaut : " &
somme_neg)
    Console.WriteLine("La moyenne des valeurs negatives vaut: " &
moyenne_neg)
    Console.ReadLine()
End Sub

End Module

```

EXERCICE 7

Ecrire un programme en Visual basic mode console qui détermine les n nombres premiers. Prévoir que n varie dans l'intervalle [10, 60].

Nota : un nombre premier est un nombre qui n'a que deux diviseurs : l'unité et lui-même.

Analyse et solution

n représente l'effectif des nombres premiers, par exemple si l'utilisateur saisit n=10, notre programme doit afficher les 10 premiers nombres premiers. Dans l'énoncé, il est dit que n doit appartenir dans [10, 60], il suffit d'utiliser une boucle demandant la valeur de n jusqu'à ce que n soit compris dans [10, 60].

Un nombre premier est un nombre qui n'a pas d'autre diviseurs hormis 1 et lui-même, dans ce cas pour chaque nombre on va parcourir de 2 à ce nombre-1 ; si on trouve un diviseur alors ce nombre n'est pas premier, sinon il est premier et on l'affiche en prenant soin de compter les nombres affichés pour que l'effectif ne dépasse pas n.

Code VB.Net

```
Module Module1

    Sub Main()
        Dim n, i, nombre, compteur As Integer
        Dim test As Boolean
        nombre = 2
        compteur = 0
        Console.WriteLine("Entrez le nombre de valeurs")
1:      n = CInt(Console.ReadLine())
        If n < 10 Or n > 60 Then
            Console.WriteLine("Le nombre de valeurs doit se trouver entre 10 et 60, recommencez")
            GoTo 1
        End If
        'recherche et affichage des nombres premiers
        Console.WriteLine("Voici les " & n & " premiers nombres premiers")
        Do
            test = True
            For i = 2 To n - 1
                If nombre Mod i = 0 Then
                    test = False
                End If
            Next
            If test = True Then
                Console.WriteLine(nombre.ToString)
                compteur += 1
            End If
            nombre += 1
        Loop Until compteur = n
        Console.ReadLine()
    End Sub

End Module
```

EXERCICE 8

Ecrire un programme Visual basic mode console de sondage d'opinion sur les compétences en programmation. Le programme affichera la phrase

« Maitrisez-vous : 1. Le QBASIC ; 2. Le VBASIC ; 3. Les deux langages ;

4. Aucun de deux », après le choix de l'utilisateur (c.-à-d. uniquement l'un de numéro défini comme étiquette), le programme va incrémenter le compteur correspondant au choix effectué. Cette opération sera répétée de manière indéterminée aussi longtemps qu'il y aura une autre personne à interroger. Et le programme affichera au final le pourcentage de chacune d'options. (Recommandation : Utilisez la boucle DO UNTIL)

Analyse et solution

On va effectuer le sondage tant qu'il y a quelqu'un à interroger, sur ce il suffit de répéter la question « il y a-t-il quelqu'un à interroger ? » jusqu'à ce que la réponse soit NON. On va utiliser 4 compteurs (parce qu'il y a 4 réponses possibles) qui vont stocker le nombre de fois que le choix correspondant a été effectué...

Code VB.Net

Module Module1

```
Sub Main()  
    Dim reponse As String  
    Dim score_vb, score_qb, score_deux, score_aucun, total, choix As Integer  
    Dim pourc_vb, pourc_qb, pourc_deux, pourc_aucun As Single  
    score_vb = 0  
    score_qb = 0  
    score_aucun = 0  
    score_deux = 0  
    Do  
        Console.WriteLine("Il y a t il quelqu'un à interroger?")  
        reponse = Console.ReadLine()  
        If UCase(reponse) = "OUI" Then  
            Console.WriteLine("Maitrisez-vous:")  
            Console.WriteLine("1. Le visual basic")  
            Console.WriteLine("2. Le Q Basic")  
            Console.WriteLine("3.Les deux")  
            Console.WriteLine("Aucun")  
            Console.WriteLine("Faites un choix")  
5:        choix = CInt(Console.ReadLine())  
        Select Case (choix)  
            Case 1  
                score_vb += 1  
            Case 2  
                score_qb += 1  
            Case 3  
                score_deux += 1  
            Case 4  
                score_aucun += 1  
            Case Else  
                Console.WriteLine("choix incorrect, recommencez")  
                GoTo 5  
        End Select  
  
        End If  
        Loop Until UCase(reponse) = "NON"  
        total = score_aucun + score_deux + score_qb + score_vb  
        pourc_aucun = score_aucun * 100 / total  
        pourc_deux = score_deux * 100 / total  
        pourc_qb = score_qb * 100 / total  
        pourc_vb = score_vb * 100 / total  
        Console.WriteLine("Sur " & total & " personnes interrogées " & pourc_vb & " %  
maitrisent le Visual Basic ")  
        Console.WriteLine("Sur " & total & " personnes interrogées " & pourc_qb & " %  
maitrisent le QBasic")  
        Console.WriteLine("sur " & total & " personnes interrogées " & pourc_deux & " %  
maitrisent les deux langages")
```

```

        Console.WriteLine("sur " & total & " personnes interrogées " & pourc_aucun & " %
ne maitrisent aucun langage")
        Console.ReadLine()

    End Sub

End Module

```

EXERCICE 9

Concevoir un programme Visual basic en mode console de gestion de mot de passe qui est constitué de 3 parties (Nom, Prénom et date de naissance) qui sont successivement lues.

- a. L'utilisateur saisit premièrement le nom. Si celui-ci correspond à votre nom, le programme passe à l'étape (b). En cas de répétition de 3 lectures incorrectes, le programme affiche le message « Le système est verrouillé » puis s'arrête ;
- b. Il saisit le prénom. Si celui-ci correspond à votre prénom, le programme passe à l'étape (c). En cas de répétition de 2 lectures incorrectes, le programme affiche également le message « Le système est verrouillé » puis s'arrête ;
- c. Il saisit la date de naissance. Si celle-ci correspond à votre date de naissance, le programme affiche le message « Accès autorisé », sinon le message « Le système est verrouillé ».

Analyse et solution

Dans ce cas, l'utilisateur doit introduire trois mots de passe, le premier doit correspondre à mon nom(DIUMI), le deuxième à mon prénom (AIME) et le troisième à ma date de naissance(24/09) pour que l'accès lui soit autorisé, donc des structures conditions suffisent pour vérifier...mais l'utilisateur ne peut pas saisir indéfiniment le mot de passe quand ce dernier est incorrect, il a seulement droit à 3 essais pour la première étape, 2 pour la deuxième et un seul pour la dernière, sur ce on doit ajouter un compteur qui comptera le nombre d'essai

Code VB. Net

```

Module Module1

    Sub Main()
        Dim mot1, mot2, mot3 As String
        Dim i, j As Integer
        'etape a
        i = 1
5:    Console.WriteLine("Entrez le premier mot de passe")
        mot1 = Console.ReadLine()
        If mot1 <> "DIUMI" Then
            i += 1

```

```

        If i < 4 Then
            Console.WriteLine("Mot de passe incorrect, recommencez")
            GoTo 5
        Else
            GoTo 20
        End If
    Else
        GoTo 10
    End If
    'étape b
    j = 1
10: Console.WriteLine("Entrez le deuxième mot de passe")
    mot2 = Console.ReadLine()
    If mot2 <> "AIME" Then
        j += 1
        If j < 2 Then
            Console.WriteLine("mot de passe incorrect, recommencez")
            GoTo 10
        Else
            GoTo 20
        End If
    Else
        GoTo 15
    End If
    'étape c
15: Console.WriteLine("Entrez le troisième mot de passe")
    mot3 = Console.ReadLine()
    If mot3 = "24/09" Then
        Console.WriteLine("Accès autorisé")
    Else
20: Console.WriteLine("Système verrouillé")
        End If
        Console.ReadLine()
    End Sub
End Module

```

EXERCICE 10

Faites un programme VB en mode console qui lit l'âge de chacun de 500 habitants d'une rue, et détermine le nombre de personnes pour chaque catégorie, et la catégorie qui a le nombre de personnes le plus élevé. Les catégories sont définies suivant l'âge de manière ci-après :

- Moins d'un an (âge<1) : bébé
- De 1 à 11 (âge>=1 et âge<12) : Enfant
- De 12 à 17 (âge>=12 et âge<18) : Adolescent
- Plus de 17 (âge>=18) : Adulte

Indication : l'âge inférieur à 1 est considéré comme 0 an pour l'utilisateur.

Analyse et solution

Notre programme doit lire l'âge de 500 habitants, dans ce cas on peut utiliser et on utilisera for car le nombre de répétition est connu d'avance...Chaque fois que l'âge est lu, on va le comparer pour voir s'il appartient à quelle catégorie enfin d'incrémenter le compteur de la catégorie correspondant.

Code VB.Net

Module Module1

```
Sub Main()
    Dim bebe, ado, adulte, enfant As Integer
    Dim i, age As Integer
    bebe = 0
    ado = 0
    adulte = 0
    enfant = 0
    For i = 1 To 500
5:      Console.WriteLine("Entrez l'age de l'habitant n° " & i)
        age = CInt(Console.ReadLine())
        If age < 1 Then
            bebe += 1
        ElseIf age >= 1 And age < 12 Then
            enfant += 1
        ElseIf age >= 12 And age < 18 Then
            ado += 1
        ElseIf age >= 18 Then
            adulte += 1
        Else
            Console.WriteLine("age incorrect, recommencez")
            GoTo 5
        End If
    Next
    Console.WriteLine("il y a au total " & bebe & " personnes pour la catégorie bébé")
    Console.WriteLine("il y a au total " & enfant & " personnes pour la catégorie
enfant")
    Console.WriteLine("il y a au total " & ado & " personnes pour la catégorie
adolescent")
    Console.WriteLine("il y a au total " & adulte & " personnes pour la catégorie
adulet")
    'détermination de la plus grande catégorie
    If bebe > enfant And bebe > ado And bebe > adulte Then
        Console.WriteLine("C'est la catégorie bébé qui possède plus de personnes")
    ElseIf enfant > bebe And enfant > ado And enfant > adulte Then
        Console.WriteLine("C'est la catégorie enfant qui possède plus de personnes")
    ElseIf ado > bebe And enfant > enfant And ado > adulte Then
        Console.WriteLine("C'est la catégorie ado qui possède plus de personnes")
    ElseIf adulte > bebe And adulte > ado And adulte > enfant Then
        Console.WriteLine("C'est la catégorie adulte qui possède plus de personnes")
    Else
        Console.WriteLine("Il n'y a pas une catégorie qui a plus de personnes que
toutes les autres")
    End If
    Console.ReadLine()
End Sub
```

EXERCICE 11

Faites un programme VB en mode console qui lit les cotes des N étudiants d'une promotion dans le cours de programmation, ensuite détermine le pourcentage de réussites et la mention associée à ce pourcentage. La mention est déterminée selon les règles suivantes :

Pourcentage	Mention
< 20	Cours non assimilé
>= 20	Cours assez bien assimilé
>= 40	Cours bien assimilé
>= 70	Cours très bien assimilé

Analyse et solution

Pour ce problème, l'utilisateur doit saisir le nombre d'étudiants, puis les cotes de ces étudiants, pour la saisie des côtes, on va utiliser la boucle for...

On peut résoudre ce problème avec ou sans tableau, avec le tableau on va le remplir avec les cotes des étudiants, puis après on va le parcourir pour compter le nombre de réussites enfin de calculer le pourcentage et déterminer la mention associée.

Et sans tableau, après la saisie de chaque côte on vérifie s'il y a réussite pour incrémenter le compteur des réussites.

Sans utiliser un tableau

Code VB.Net

Module Module1

```

Sub Main()
    Dim cote, n, i, reussite As Integer
    Dim pourcentage As Single
    reussite = 0
    Console.WriteLine("Entrez le nombre d'étudiants")
    n = CInt(Console.ReadLine())
    For i = 1 To n
        Console.WriteLine("Entre la cote de l'étudiant n° " & i)
        cote = CInt(Console.ReadLine())
        If cote >= 10 Then
            reussite += 1
        End If
    Next i
    pourcentage = reussite / n * 100
    Console.WriteLine("Le pourcentage de réussites est de " & pourcentage & "%")
    Console.WriteLine("La mention associée est : ")
    Select Case pourcentage
        Case < 20
            Console.WriteLine("Cours non assimilé")
        Case >= 20 And < 40
            Console.WriteLine("Cours assez bien assimilé")
        Case >= 40 And < 70
            Console.WriteLine("Cours bien assimilé")
        Case >= 70
            Console.WriteLine("Cours très bien assimilé")
    End Select
End Sub

```



```

        End If
    Next
    pourcentage = reussite * 100 / n
    'Détermination de la mention
    If pourcentage < 20 Then
        Console.WriteLine("Il y a eu " & pourcentage & " % de reussites, donc le
cours est non assimilé")
    ElseIf pourcentage >= 20 And pourcentage < 40 Then
        Console.WriteLine("Il y a eu " & pourcentage & " % de reussites, donc le
cours est assez bien assimilé")
    ElseIf pourcentage >= 40 And pourcentage < 70 Then
        Console.WriteLine("Il y a eu " & pourcentage & " % de reussites, donc le
cours est bien assimilé")
    Else
        Console.WriteLine("Il y a eu " & pourcentage & " % de reussites, donc le
cours est très bien assimilé")
    End If
    Console.ReadLine()
End Sub

End Module

```

EXERCICE 12

Ecrire un programme VB en mode console qui lit les éléments d'un tableau linéaire T ayant n éléments, ensuite transfère les éléments supérieurs ou égaux à 50 dans un tableau TREUSSITE, et les éléments inférieurs à 50 dans un tableau TECHEC.

Analyse et solution

Pour ce problème, on aura besoin de trois tableaux, la taille du premier sera déterminée par l'utilisateur tandis que pour les deux autres leurs tailles dépendront des valeurs saisies.

Code VB.Net

```

Module Module1

    Sub Main()
        Dim n, i As Integer
        Console.WriteLine("Entrez la taille du tableau")
        n = CInt(Console.ReadLine())
        'déclaration des tableaux
        Dim T(n - 1) As Integer
        Dim TREUSSITE(n - 1) As Integer
        Dim TECHEC(n - 1) As Integer
        'remplissage du tableau
        Console.WriteLine("Remplissage du tableau")
        For i = 0 To n - 1
            Console.WriteLine("Entrez l'élément n° " & i)
            T(i) = CInt(Console.ReadLine())
        Next
    End Sub
End Module

```

```

'Transfert des éléments
Dim j, k As Integer
j = 0
k = 0
For i = 1 To n - 1
    If T(i) >= 50 Then
        TREUSSITE(j) = T(i)
        j += 1
    Else
        TECHEC(k) = T(i)
        k += 1
    End If
Next
'affichage de deux tableaux
If j > 0 Then
    Console.WriteLine("Voici le contenu du tableau TREUSSITE")
    For i = 0 To j - 1
        Console.WriteLine(TREUSSITE(i).ToString)
    Next
Else
    Console.WriteLine("Le tableau TREUSSITE n'a aucun élément")
End If
If k > 0 Then
    Console.WriteLine("Voici le contenu du tableau TECHEC")
    For i = 0 To k - 1
        Console.WriteLine(TECHEC(i).ToString)
    Next
Else
    Console.WriteLine("Le tableau TECHEC n'a aucun élément")
End If
Console.ReadLine()
End Sub

End Module

```

EXERCICE 13

Ecrire un programme en VB en mode console qui permet de compléter les informations sur les étudiants d'une université dans un tableau « liste » de type structure, de taille 50, dont la structure est : nom, prénom, sexe et âge ; ensuite affiche ces informations suivant l'ordre croissant des noms. En cas d'égalité des noms (noms identiques), un tri croissant s'effectue sur les prénoms.

Analyse et solution

Code VB.Net

```

Module Module1
    Structure etudiant
        Dim nom As String
        Dim prenom As String
        Dim sexe As String
        Dim age As Integer
    End Structure

```

```

Sub Main()
    Dim liste(49) As etudiant
    Dim i, j As Integer
    Dim intermediaire As etudiant
    'lecture des informations des étudiants
    For i = 0 To 49
        Console.WriteLine("Les informations de l'étudiant n° " & i)
        Console.WriteLine("Son nom :")
        liste(i).nom = Console.ReadLine()
        Console.WriteLine("Son prénom")
        liste(i).prenom = Console.ReadLine()
        Console.WriteLine("Son sexe")
        liste(i).sexe = Console.ReadLine()
        Console.WriteLine("Son age")
        liste(i).age = CInt(Console.ReadLine())
    Next
    'tri des informations d'après les noms
    For i = 49 To 2 Step -1
        For j = 1 To i - 1
            If liste(j).nom > liste(j + 1).nom Then
                intermediaire = liste(j)
                liste(j) = liste(j + 1)
                liste(j + 1) = intermediaire
            ElseIf liste(j).nom = liste(j + 1).nom Then
                If liste(j).prenom > liste(j + 1).prenom Then
                    intermediaire = liste(j)
                    liste(j) = liste(j + 1)
                    liste(j + 1) = intermediaire
                End If
            End If
        Next
    Next
    'Affichage des informations
    Console.WriteLine("Les informations des étudiants")
    Console.WriteLine("NOMS      PRENOMS      SEXE      AGE")
    For i = 0 To 49
        Console.WriteLine(liste(i).nom & " " & liste(i).prenom & " " & liste(i).sexe
& " " & liste(i).age)
    Next
    Console.ReadLine()
End Sub

End Module

```

EXERCICE 14

Le guichet unique de Boma organise la tarification des containers des produits manufacturés à l'aide des rubriques : frais de dédouanement (fd), frais de contrôle de qualité (focc) et frais de l'Onatra (fo).

Les frais du dédouanement exprimés en unité monétaire sont déterminés à partir des catégories, suivant le tableau ci-après :

Catégorie	Prix/kilo
Engrais chimiques et semences améliorées	0,25
Produits vivriers et pharmaceutiques	0,5
Fournitures de bureau et scolaires	0,75
Matériaux de construction	1
Pièces de rechange	1,5
Divers	2

Les frais de contrôle de qualité valent 4% des frais de dédouanement.

Les frais de l'Onatra sont composés de deux parties : frais de manutention (fm) et frais d'emmagasiner (fe). La catégorie « Engrais chimiques et semences améliorées » en est exonérée.

Les frais de manutention sont calculés en fonction du poids des containers.

Ils sont calculés de la manière suivante : Jusqu'à 100 tonnes, les frais sont fixés à 0.005/kilo et à partir de la 101ème tonne, les frais sont fixés à 0.003/kilo.

Les frais d'emmagasiner sont calculés en fonction du nombre de containers. Ils valent 20 unités monétaires par container et par jour.

On vous demande d'écrire un programme Visual basic en mode console constitué d'un module principal et des fonctions chargées de calculer : frais de dédouanement, frais de contrôle de qualité, frais de manutention(fm) et frais d'emmagasiner(fe).

Solution

Module Module1

```
Sub Main()
    Dim cat As String
    Dim fd, focc, fm, fe, fo, np As Long
    Dim nbre_jrs, nbre_c, poids As Integer
5: Console.WriteLine("Entrez la catégorie")
    cat = Console.ReadLine
    Console.WriteLine("Entrez le poids")
    poids = CInt(Console.ReadLine())
    Console.WriteLine("Entrez le nombre de containers")
    nbre_c = CInt(Console.ReadLine())
    Console.WriteLine("Entrez le nombre de jours d'emmagasinage")
    nbre_jrs = Console.ReadLine
    fd = dedouanement(cat, poids)
    focc = controle(fd)
    If UCase(cat) = "ENGRAIS CHIMIQUES ET SEMENCES AMELIOREES" Then
        fo = 0
    Else
        fm = manutention(poids)
        fe = emmagasinage(nbre_c, nbre_jrs)
        fo = fm + fe
    End If
    If fm = 0 Or fd = 0 Then
        Console.WriteLine("Il y a eu erreur des données, recommencez")
        GoTo 5
    Else
        np = fd + fo + focc
        Console.WriteLine("Le net à payer est: " & np)
    End If
    Console.ReadLine()
End Sub
'fonction qui calcule les frais de dédouanement
Function dedouanement(ByVal cat As String, ByVal poids As Integer) As Long
    Dim fd As Long
    If UCase(cat) = "ENGRAIS CHIMIQUES ET SEMENCES AMELIOREES" Then
        fd = poids * 1000 * 0.25
    ElseIf UCase(cat) = "PRODUITS VIVRIERS ET PHARMACEUTIQUES" Then
        fd = poids * 1000 * 0.5
    ElseIf UCase(cat) = "FOURNITURES DE BUREAU ET SCOLAIRES" Then
        fd = poids * 1000 * 0.75
    ElseIf UCase(cat) = "MATERIAUX DE CONSTRUCTION" Then
        fd = poids * 1000 * 1
    ElseIf UCase(cat) = "PIECES DE RECHANGE" Then
        fd = poids * 1000 * 1.5
    ElseIf UCase(cat) = "DIVERS" Then
        fd = poids * 1000 * 2
    Else
        Console.WriteLine("catégorie incorrecte")
        fd = 0
    End If
    Return fd
End Function
```

```

End Function

'fonction qui calcule les frais de contrôle de qualité
Function controle(ByVal fd As Long) As Long
    Dim focc As Long
    focc = fd * 0.04
    Return focc
End Function

'fonction qui calcule les frais de manutention
Function manutention(ByVal poids As Integer) As Long
    Dim fm As Long
    If poids >= 0 And poids <= 100 Then
        fm = poids * 1000 * 0.005
    ElseIf poids > 100
        fm = 100 * 1000 * 0.005 + (poids - 100) * 1000 * 0.003
    Else
        Console.WriteLine("poids incorrect")
        fm = 0
    End If
    Return fm
End Function

'fonction qui calcule les frais d'emmagasiner
Function emmagasinage(ByVal nbreC As Integer, ByVal nbJ As Integer) As Long
    Dim fe As Long
    fe = 20 * nbreC * nbJ
    Return fe
End Function

End Module

```

EXERCICE 15

Un magasin de la place organise le paiement de salaire de ses employés à l'aide des rubriques : salaire de base (sb) et prime familiale (pf) :

Les salaires de base exprimés en unité monétaire sont déterminés à partir des grades, suivant le tableau ci-après :

Grade	Salaire de base
Gérant	200
Comptable	180
Caissier	160
Facturier	150
Livreur	130
Vendeur	110

La prime familiale est composée de deux parties : de la prime mariage (pm) et de la prime enfant (pe). La prime mariage est octroyée à tous les agents qui ne sont

pas célibataires (c.à.d. mariés, divorcés ou veufs). Elle vaut 2% du salaire de base de l'agent.

La prime enfant est calculée en fonction du nombre d'enfants vivants de moins de 19 ans. Elle est calculée de la manière suivante : Jusqu'à 5 enfants, chacun génère une prime fixe de 50. A partir du 6^{ème} enfant, la prime fixe devient de 20. Enfin, pas de prime à partir du 11^e enfant.

On vous demande d'écrire un programme Visual basic en mode console constitué d'un module principal et des fonctions chargées de calculer : le salaire de base, la prime mariage et la prime enfant.

Solution

Code VB.Net

Module Module1

```
Sub Main()
    Dim grade, etat_civil As String
    Dim nef, sb, pe, pm, pf, sn As Integer
5:   Console.WriteLine("Entrez le grade de l'agent")
    grade = Console.ReadLine()
    Console.WriteLine("Entrez son état civil")
    etat_civil = Console.ReadLine()
    Console.WriteLine("Entrez le nombre d'enfants")
    nef = CInt(Console.ReadLine())
    'Appel des fonctions
    sb = salaireBase(grade)
    pm = primeMariage(etat_civil, sb)
    pe = PrimeEnfant(nef)
    pf = pm + pe
    If sb = 0 Or pe = -1 Or pm = -1 Then
        Console.WriteLine("Données incorrectes, recommencez")
        GoTo 5
    Else
        sn = sb + pf
        Console.WriteLine("Le salaire net vaut : " & sn)
    End If
    Console.ReadLine()
End Sub
'fonction qui calcule le salaire de base
Function salaireBase(ByVal grade As String) As Integer
    Dim sb As Integer
    If UCase(grade) = "GERANT" Then
        sb = 200
    ElseIf UCase(grade) = "COMPTABLE" Then
        sb = 180
    ElseIf UCase(grade) = "CAISSIER" Then
        sb = 160
    ElseIf UCase(grade) = "FACTURIER" Then
        sb = 150
    ElseIf UCase(grade) = "LIVREUR" Then
        sb = 130
    ElseIf UCase(grade) = "VENDEUR" Then
```

```

        sb = 110
    Else
        sb = 0
    End If
    Return sb
End Function

'fonction qui calcule la prime mariage
Function primeMariage(ByRef etatCivil As String, ByVal sb As Integer) As Integer
    Dim pm As Integer
    If UCase(etatCivil) <> "CELIBATAIRE" Or UCase(etatCivil) <> "MARIE(E)" Or
UCase(etatCivil) <> "DIVORCE(E)" Or UCase(etatCivil) <> "VEUF(VE)" Then
        pm = -1
    End If

    If UCase(etatCivil) = "CELIBATAIRE" Then
        pm = 0
    Else
        pm = sb * 0.02
    End If
    Return pm
End Function

'fonction qui calcule la prime enfant
Function PrimeEnfant(ByRef nef As Integer) As Integer
    Dim pe As Integer
    Select Case (nef)
        Case 0 To 5
            pe = nef * 50
        Case 6 To 10
            pe = 5 * 50 + (nef - 5) * 20
        Case Is > 10
            pe = 5 * 50 + 5 * 20
        Case Else
            pe = -1
    End Select
    Return pe
End Function
End Module

```


EXERCICE 16

Ecrire un programme Visual basic en mode console qui lit une phrase, une lettre se trouvant dans la phrase et une autre lettre quelconque, et ensuite le programme remplace à chaque occurrence la première lettre par la deuxième

Solution

Module Module1

```
Sub Main()  
    Dim chaine, lettre1, lettre2 As String  
    Dim test As Boolean = False  
    Dim i As Integer  
    Console.WriteLine("Entrez la chaine")  
    chaine = Console.ReadLine()  
1:    Console.WriteLine("Entrez la lettre à remplacer")  
    lettre1 = Console.ReadLine()  
    'on vérifie si la lettre se trouve dans la chaine  
    For Each caractere As String In chaine  
        If caractere = lettre1 Then  
            test = True  
        End If  
    Next  
    If test = True Then  
        GoTo 5  
    Else  
        Console.WriteLine("La lettre de se trouve pas dans la chaine, recommencez")  
        GoTo 1  
    End If  
    'remplacement  
5:    Console.WriteLine("Entrez la lettre de remplacement")  
    lettre2 = Console.ReadLine()  
    For i = 1 To Len(chaine)  
        If Mid(chaine, i, 1) = lettre1 Then  
            Mid(chaine, i, 1) = lettre2  
        End If  
    Next  
    Console.WriteLine("La chaine après remplacement est : " & chaine)  
    Console.ReadLine()  
  
End Sub  
  
End Module
```

EXERCICE 17

Ecrire un programme Visual basic en mode console qui demande un mot puis donne le mot miroir. Par exemple, si l'utilisateur saisit DIUMI, le programme affiche « IMUID ».

Solution

Module Module1

```
Sub Main()  
    Dim mot, s As String  
    Dim i As Integer  
    Console.WriteLine("Entrez le mot")  
    mot = Console.ReadLine()  
    For i = 1 To Len(mot) \ 2  
        s = Mid(mot, i, 1)  
        Mid(mot, i, 1) = Mid(mot, Len(mot) - i + 1, 1)  
        Mid(mot, Len(mot) - i + 1, 1) = s  
    Next  
    Console.WriteLine("Le mot miroir est : " & mot)  
    Console.ReadLine()  
End Sub
```

End Module

EXERCICE 18

Ecrire un programme qui permet de dire si un mot est un mot palindrome, c.à.d. identique à son mot miroir. Exemple : KAYAK, ESSE....

Solution

Module Module1

```
Sub Main()  
    Dim mot As String  
    Dim i As Integer = 1  
    Dim test As Boolean = True  
    Console.WriteLine("Entrez le mot")  
    mot = Console.ReadLine()  
    For i = 1 To Len(mot) \ 2  
        If Mid(mot, i, 1) <> Mid(mot, Len(mot) - i + 1, 1) Then  
            test = False  
        End If  
    Next  
    If test = True Then  
        Console.WriteLine("Le mot est palindrome")  
    Else  
        Console.WriteLine("Le mot n'est pas palindrome")  
    End If  
    Console.ReadLine()  
End Sub
```

End Module

EXERCICE 19

Ecrire en Visual basic mode console une procédure `decale1` qui reçoit en argument un tableau d'entiers `t` (à une dimension) et un entier positif inférieur la taille du tableau, puis décale les éléments du tableau d'un pas `n` vers la gauche de façon circulaire. Si le tableau initialement :

5	2	3	0	8	7	3	1
---	---	---	---	---	---	---	---

Si `n=2`, la procédure le transforme en :

3	0	8	7	3	1	5	2
---	---	---	---	---	---	---	---

Solution

Module `Module1`

```
Sub Main()  
    Dim m, n As Integer  
    Console.WriteLine("Entrez la taille du tableau")  
    m = CInt(Console.ReadLine())  
    Dim T(m - 1) As Integer  
    Do  
        Console.WriteLine("Entrez le décalage")  
        n = CInt(Console.ReadLine())  
    Loop Until n < m  
    For I As Integer = 0 To T.Length - 1  
        Console.WriteLine("entrez l'élément n° " & I + 1)  
        T(I) = CInt(Console.ReadLine())  
    Next  
    decale1(T, n)  
End Sub  
Sub decale1(ByRef A() As Integer, ByVal n As Integer)  
    Dim k As Integer = A.Length  
    Dim b(k - 1) As Integer  
    Dim i As Integer  
    For i = 0 To k - n - 1  
        b(i) = A(i + n)  
    Next  
    For i = 0 To n - 1  
        b(k - n + i) = A(i)  
    Next  
    Console.WriteLine(" voici le nouveau tableau après décalage")  
    For i = 0 To k - 1  
        Console.WriteLine(b(i).ToString)  
    Next  
    Console.ReadLine()  
End Sub  
End Module
```

EXERCICE 20

Ecrire une procédure en Visual basic mode console qui détermine, puis affiche le plus grand entier pair contenu dans un tableau d'entiers strictement positifs. Attention si le tableau ne contient aucun entier pair, la procédure affichera le message « Le tableau ne contient aucun entier pair »

Solution

Code VB.Net

Module Module1

```
Sub Main()
    Dim n, i As Integer
    Console.WriteLine("Entrez la taille du tableau")
    n = CInt(Console.ReadLine())
    Dim A(n - 1) As Integer
    Console.WriteLine("Lecture des éléments du tableau")
    For i = 0 To n - 1
5:      Console.WriteLine("Entrez l'élément n° " & i + 1)
        A(i) = CInt(Console.ReadLine())
        If A(i) < 0 Then
            Console.WriteLine("entrez un nombre positif seulement, recommencez")
            GoTo 5
        End If
    Next
    affiche_pair(A)
    Console.ReadLine()
End Sub
Sub affiche_pair(ByVal A() As Integer)
    Dim max, i As Integer
    max = 0
    For i = 0 To A.Length - 1
        If A(i) Mod 2 = 0 Then
            If A(i) > max Then
                max = A(i)
            End If
        End If
    Next
    If max = 0 Then
        Console.WriteLine("Le tableau ne contient aucun entier pair")
    Else
        Console.WriteLine("Le plus grand entier pair du tableau est: " & max)
    End If
End Sub
End Module
```

EXERCICE 21

Proposez en Visual basic en mode console un programme demandant à l'utilisateur la saisie de 10 nombres premiers compris entre 1 et 100 qui seront sauvegardés dans un tableau linéaire, puis partant de ces 10 nombres, le programme calcule et affiche la somme ainsi que la moyenne de tous les nombres impairs.

Solution

Code VB.Net

Module Module1

```
Sub Main()
    Dim A(9) As Integer
    Dim i, n As Integer
    Dim test As Boolean
    'lecture de nombres premiers
    i = 0
    Do
        Console.WriteLine("entrez un autre nombre premier entre 1 et 100")
5:      n = CInt(Console.ReadLine())
        test = True
        For j As Integer = 2 To n - 1
            If n Mod j = 0 Then
                test = False
            End If
        Next
        If n < 1 Or n > 100 Or test = False Then
            Console.WriteLine("Le nombre saisi n'est pas premier ou n'est pas inclus
de 1 à 100, recommencez")
            GoTo 5
        Else
            A(i) = n
            i += 1
        End If
    Loop Until i = 10
    'calcul de la somme et moyenne de nombres impaires
    Dim moyenne, somme, t As Integer
    moyenne = 0
    somme = 0
    t = 0
    For J As Integer = 0 To 9
        If A(J) Mod 2 <> 0 Then
            somme += A(J)
            t += 1
        End If
    Next
    If t = 0 Then
        moyenne = 0
    Else
        moyenne = somme / t
    End If
    Console.WriteLine("La moyenne des nombres impairs vaut : " & moyenne)
```

```
Console.WriteLine("La somme des nombres impairs vaut : " & somme)
Console.ReadLine()
```

```
End Sub
```

```
End Module
```

EXERCICE 22

Ecrire un programme en Visual basic en mode console utilisant un tableau linéaire de taille 50. Ce programme lit un nombre réel et l'insère dans le tableau si celui-ci est compris entre 65 et 100 et qu'il n'a jamais été inséré auparavant.

Solution

Code VB.Net

```
Module Module1
```

```
Sub Main()
    Dim A(49) As Single
    Dim n As Single
    Dim test As Boolean
    test = False
    Console.WriteLine("Saisie des éléments du tableau")
    For i As Integer = 0 To 49
        Console.WriteLine("entrez l'élément n° " & i + 1)
        A(i) = Val(Console.ReadLine())
    Next
    Console.WriteLine("entrez un nombre réel")
    n = Val(Console.ReadLine())
    For j As Integer = 0 To 49
        If A(j) = n Then
            test = True
        End If
    Next
    If test = True Or (n < 65 Or n > 100) Then
        Console.WriteLine("Le nombre a déjà été inséré ou il n'est pas compris entre
65 et 100")
    ElseIf test = False Then
        'on redimensionne le tableau pour insérer l'élément
        ReDim Preserve A(50)
        A(50) = n
        'on affiche le nouveau tableau
        Console.WriteLine("Voici le nouveau tableau")
        For i As Integer = 0 To 50
            Console.WriteLine(A(i).ToString)
        Next
    End If
    Console.ReadLine()
End Sub
```

```
End Module
```

EXERCICE 23

Proposez en Visual Basic.Net en mode console, un programme demandant à l'utilisateur la saisie de 50 nombres quelconques compris entre 1 et 100 qui seront sauvegardés dans un tableau linéaire, puis partant de ces 50 nombres, le programme calcule et affiche la somme ainsi que la moyenne géométrique de tous les nombres divisibles par 5.

Solution

Code VB.Net

Module Module1

```
Sub Main()  
    Dim A(49) As Integer  
    Dim somme, produit As Integer  
    Dim g As Single  
    Dim i, n, t As Integer  
    i = 0  
    Do  
        Console.WriteLine("entrez un nombre compris entre 1 et 100")  
5:      t = Val(Console.ReadLine())  
        If t < 1 Or t > 100 Then  
            Console.WriteLine("Entrez seulement un nombre entre 1 et 100,  
recommencez")  
            GoTo 5  
        Else  
            Console.WriteLine("nombre correct")  
            A(i) = t  
            i += 1  
        End If  
    Loop Until i = 50  
    'calcul de la somme et de la moyenne des nombres divisibles par 5  
    somme = 0  
    produit = 1  
    n = 0  
    For J As Integer = 0 To 49  
        If A(J) Mod 5 = 0 Then  
            somme += A(J)  
            produit *= A(J)  
            n += 1  
        End If  
    Next  
    If n = 0 Then  
        g = 0  
    Else  
        g = (produit) ^ (1 / n)  
    End If  
    Console.WriteLine("La somme des nombres divisibles par 5 vaut :" & somme)  
    Console.WriteLine("La moyenne géométrique des nombres divisibles par 5 vaut: " &  
g)      Console.ReadLine()  
  
End Sub
```

End Module

EXERCICE 24

L'interclassement consiste à réunir en une seule liste triée, deux listes qui sont préalablement triées. Ecrire une procédure interclas qui reçoit en argument deux tableaux triés A de dimension N et B de dimension P, puis effectue l'interclassement et affiche le résultat final.

(Attention : sans utiliser l'algorithme de tri)

Solution

Module Module1

```
Sub Main()  
    Dim n, p, i, j As Integer  
    Console.WriteLine("entrez la taille du premier tableau")  
    n = CInt(Console.ReadLine())  
    Dim A(n - 1) As Integer  
    Console.WriteLine("entrez la taille du deuxieme tableau")  
    p = CInt(Console.ReadLine())  
    Dim B(p - 1) As Integer  
    Console.WriteLine("lecture des elements du premier tableau")  
    For i = 0 To n - 1  
        Console.WriteLine("entrez l'élément n° " & i + 1)  
        A(i) = CInt(Console.ReadLine())  
    Next  
    'on trie le tableau A  
    Array.Sort(A)  
    Console.WriteLine("lecture des elements du deuxieme tableau")  
    For i = 0 To p - 1  
        Console.WriteLine("entrez l'élément n° " & i + 1)  
        B(i) = CInt(Console.ReadLine())  
    Next  
    'on trie le tableau B  
    Array.Sort(B)  
    interclas(A, B)  
End Sub  
  
Sub interclas(ByVal A() As Integer, ByVal B() As Integer)  
    Dim i, k, j, m, n, p As Integer  
    n = A.Length  
    p = B.Length  
    m = n + p  
    Dim C(m - 1) As Integer  
    i = 0  
    k = 0  
5:   C(k) = A(i)  
    i += 1  
    k = i  
    If i <= n - 1 Then  
        GoTo 5  
    End If  
End Sub
```



```

Else
    j = 0
End If
10: k = j + n - 1
C(k) = B(j)
j += 1
If j <= p - 1 Then
    GoTo 10
End If
Array.Sort(C)
Console.WriteLine("voici la fusion de deux tableaux")
For i = 0 To m - 1
    Console.WriteLine(C(i).ToString)
Next
Console.ReadLine()
End Sub
End Module

```

EXERCICE 25

Faites un programme en Visual Basic.Net qui entre 50 nombres, compris entre 10 et 100 inclusivement dans un tableau linéaire, puis affiche chaque nombre entré, si ce dernier n'a jamais été affiché auparavant. Assurez-vous de pouvoir traiter le « pire de scenario » qui consiste en une série de 50 nombres différents.

Solution

Code VB.Net

```

Module Module1

    Sub Main()
        Dim A(49) As Integer
        Dim n, i As Integer
        Dim test As Boolean
        i = 0
        Console.WriteLine("entrez un nombre entre 10 et 100")
        Do
5:           n = CInt(Console.ReadLine())
            If n < 10 Or n > 100 Then
                Console.WriteLine("entrez uniquement un nombre compris entre 10 et 100,
recommencez")
                GoTo 5
            Else
                A(i) = n
                i += 1
                Console.WriteLine("nombre correcte,entrez un autre nombre")
            End If
        Loop Until i = 50
        'on affiche les nombres distincts du tableau
        Console.WriteLine("voici les nombres distinct du tableau")
        Console.WriteLine(A(0).ToString)
        For i = 1 To 49
            test = True

```

```

        For j As Integer = 0 To i - 1
            If A(j) = A(i) Then
                test = False
            End If
        Next
        If test = True Then
            Console.WriteLine(A(i).ToString)
        End If
    Next
    Console.ReadLine()
End Sub

```

```
End Module
```

EXERCICE 26

Ecrire une procédure en Visual Basic en mode console, qui détermine, puis affiche le plus petit entier impair contenu dans un tableau d'entiers strictement positifs. Attention, si le tableau ne contient aucun entier impair, la procédure affichera le message « Le tableau ne contient aucun entier impair ».

Solution

Code VB.Net

```
Module Module1
```

```

Sub Main()
    Dim n As Integer
    Console.WriteLine("entrez la taille du tableau")
    n = CInt(Console.ReadLine())
    Dim A(n - 1) As Integer
    'lecture des éléments du tableau
    Console.WriteLine("Saisie des éléments du tableau")
    For i As Integer = 0 To n - 1
        Do
            Console.WriteLine("entrez l'element n °" & i + 1)
            A(i) = CInt(Console.ReadLine())
        Loop Until A(i) > 0
    Next
    determine(A)
    Console.ReadLine()
End Sub
Sub determine(ByVal A() As Integer)
    Dim min As Integer
    Dim n As Integer
    n = A.Length
    'recherche premier entier impair
    Dim j, pos As Integer
    Dim test As Boolean = False
    j = 0
    Do
        If A(j) Mod 2 <> 0 Then
            pos = j
        End If
        j += 1
    Loop
End Sub

```

```

        test = True
    Else
        j += 1
    End If
Loop Until test = True Or j = n
'recherche du plus petit nombre impair
min = A(j)
For I As Integer = j + 1 To n - 1
    If A(I) Mod 2 <> 0 Then
        If A(I) < min Then
            min = A(I)
        End If
    End If
Next
If min Mod 2 = 0 Then
    Console.WriteLine("Le tableau ne contient pas de nombre impair")
Else
    Console.WriteLine("Le plus petit nombre impair est " & min)
End If
End Sub

End Module

```

EXERCICE 27

Faites un programme VB qui lit les noms d'un nombre déterminé d'étudiants, puis affiche chaque nom entré, si ce dernier n'a jamais été affiché auparavant. Assurez-vous de pouvoir tenir compte de toutes les casses possibles (par exemple les lettres minuscules correspondent aux lettres majuscules)

Solution

Code VB.Net

Module Module1

```

Sub Main()
    Dim n As Integer
    Console.WriteLine("Entrez le nombre d'étudiants")
    n = CInt(Console.ReadLine())
    Dim Noms(n - 1) As String
    For i As Integer = 0 To n - 1
        Console.WriteLine("Saisir le nom du " & i + 1 & "e étudaint")
        Noms(i) = Console.ReadLine()
    Next
    Console.WriteLine("voici les noms des étudiants")
    Console.WriteLine("Sans tenir compte des majuscules et minuscules")
    Console.WriteLine(Noms(0))
    For i As Integer = 1 To n - 1
        Dim test As Boolean = False
        For J As Integer = 0 To i - 1
            If UCase(Noms(J)) = UCase(Noms(i)) Then
                test = True
            End If
        Next
        Console.WriteLine(Noms(i))
    Next
End Sub

```

```

        Next
        If test = False Then
            Console.WriteLine(Noms(i))
        End If
    Next
    Console.WriteLine("En tenant compte des majuscules et minuscules")
    Console.WriteLine(Noms(0))
    For i As Integer = 1 To n - 1
        Dim test As Boolean = False
        For J As Integer = 0 To i - 1
            If Noms(J) = Noms(i) Then
                test = True
            End If
        Next
        If test = False Then
            Console.WriteLine(Noms(i))
        End If
    Next
    Console.ReadLine()
End Sub

End Module

```

EXERCICE 28

Ecrire un programme Visual Basic qui permet de lire les données sur l'âge (celui-ci ne pouvant pas être en dessous de 16, ni au-delà de 65) et le sexe (les valeurs possibles étant M ou F) de 100 personnes de la communauté estudiantine constituant un échantillon statistique, ensuite il détermine l'âge le plus élevé, l'âge le moins élevé, l'âge moyen, le pourcentage des hommes et celui des femmes.

Solution

Code VB.Net

```

Module Module1

    Sub Main()
        Dim age(99) As Integer
        Dim sexe(99) As String
        Dim i As Integer
        Console.WriteLine("Lecture de données")
        For i = 0 To 99
            Console.WriteLine("entrez l'age de l'etudiant n° " & i + 1)
            age(i) = CInt(Console.ReadLine())
            If age(i) < 16 Or age(i) > 65 Then
                Console.WriteLine("l'age doit etre compris entre 16 et 65, recommencez")
                GoTo 5
            Else
                Console.WriteLine("entrez le sexe de l'etudiant n°" & i + 1)
                sexe(i) = Console.ReadLine()
                If UCase(sexe(i)) <> "M" And UCase(sexe(i)) <> "F" Then
                    Console.WriteLine("les deux valeurs permises pour le sexe sont M ou F,
recommencez")
                End If
            End If
        Next
    End Sub
End Module

```

```

        GoTo 15
    End If
End If
Next
'recherche de l'age le plus élevé
Dim max As Integer
max = age(0)
For j As Integer = 1 To 99
    If age(j) > max Then
        max = age(j)
    End If
Next
'recherche de l'age le plus bas
Dim min As Integer
min = age(0)
For j As Integer = 1 To 99
    If age(j) < min Then
        min = age(j)
    End If
Next
'calcul de l'age moyen
Dim somme As Integer
Dim moy As Single
somme = 0
For j As Integer = 0 To 99
    somme += age(j)
Next
moy = somme / 9
'calcul du pourcentage des hommes et femmes
Dim Som_H, som_F As Integer
Dim pH, pF As Single
som_F = 0
Som_H = 0
For J As Integer = 0 To 99
    If UCase(sexe(J)) = "M" Then
        Som_H += 1
    Else
        som_F += 1
    End If
Next
pH = (Som_H * 100) / 100
pF = (som_F * 100) / 100
Console.WriteLine("L'age le plus élevé est : " & max)
Console.WriteLine("L'age le moins élevé est: " & min)
Console.WriteLine("L'age moyen est: " & moy)
Console.WriteLine("il y a " & pF & "% des femmes")
Console.WriteLine("il y a " & pH & "% des hommes")
Console.ReadLine()

End Sub

End Module

```

EXERCICE 29

A la naissance de Junior, sa maman lui ouvre avec un montant de départ de 100 dollars, un compte bancaire dont le taux d'intérêt annuel est de 4%. Ensuite, à chaque anniversaire, la maman de Junior verse sur ce compte 100 dollars, auxquels elle ajoute un montant correspondant au double de l'âge de Junior. Par exemple lorsqu'il aura 4 ans, elle lui versera 108 dollars. Ecrire un programme Visual Basic en mode console qui permette de déterminer la somme qu'aura Junior lors de son nième anniversaire (avec $n \geq 2$).

Analyse et Résolution

Le montant initial est de 100 dollars.

Après une année, il aura $100 + 100 * 0.04 + 102 = 206$

Pour la deuxième année : $206 + 206 * 0.04 + 104$. On constate que le montant courant de la première année devient le montant initial de la deuxième année. On peut généraliser : pour une année donnée, on aura :

Montant initial (montant courant de l'année passée)

+ Montant initial * 0.04 (Intérêt annuel)

+ $100 + 2 * \text{le nombre d'année}$.

Code VB.Net

```
Module Module1
```

```
    Sub Main()  
        Dim n, i As Integer  
        Dim Montant_initial, Montant_courant As Single  
        Console.WriteLine("Entrez le nombre d'années")  
5:        n = CInt(Console.ReadLine())  
        If n < 2 Then  
            Console.WriteLine("Le nombre d'années doit être supérieur ou égal à 2,  
recommencez")  
            GoTo 5  
        Else  
            Montant_initial = 100  
            For i = 1 To n  
                Montant_courant = Montant_initial * 1.04 + 100 + 2 * i  
                Montant_initial = Montant_courant  
            Next  
        End If  
        Console.WriteLine("Au " & n & " e anniversaire, il aura: " &  
Montant_courant)  
        Console.ReadLine()  
    End Sub
```

```
End Module
```

EXERCICE 30

Ecrire un programme Basic qui lit une phrase, ensuite le parcourt pour en déterminer les mots qui sont repris plusieurs fois dans celle-ci et nombre de répétitions pour chacun d'eux. (Exemple : dans la phrase « David est vaillant. Un vaillant, est un homme prudent », les mots est, vaillant et un sont repris chacun deux fois).

Analyse et résolution

Après la saisie de la phrase, pour connaître les mots présents dans la phrase, on peut utiliser la fonction Split qui convertit une chaîne de caractères en un tableau ayant pour éléments les mots formant la chaîne. Etant donné que le séparateur de mots utilisé par la fonction est l'espace, il faut bien prendre soin de vérifier s'il y a les principaux signes de ponctuation à la fin des mots pour ne pas prendre ça en compte.

Code VB.Net

```
Module Module1
```

```
Sub Main()
    Dim phrase As String
    Dim n, i, j, k As Integer
    Dim A(5) As Integer
    Dim m(500) As String
    Dim test As Boolean
    Console.WriteLine("Entrez la phrase")
    phrase = Console.ReadLine
    'Détermination du nombre de mots de la phrase
    Dim Mots() As String = Split(phrase)
    n = Mots.Length - 1
    ' on vérifie si les principaux signes de ponctuation se trouvent à la fin de
    certains mots
    For i = 0 To n
        If Mid(Mots(i), Mots(i).Length, 1) = "," Or Mid(Mots(i),
            Mots(i).Length, 1) = ";" Or Mid(Mots(i), Mots(i).Length, 1) = ":" Or
            Mid(Mots(i), Mots(i).Length, 1) = "!" Or Mid(Mots(i), Mots(i).Length,
            1) = "?" Or Mid(Mots(i), Mots(i).Length, 1) = "." Then
            Mots(i) = Mid(Mots(i), 1, Mots(i).Length - 1)
        End If
    Next
    'on retient les mots distincts qui composent la phrase
    m(0) = Mots(0)
    k = 1
    For i = 1 To n
        test = False
        For j = 0 To i - 1
            If UCase(Mots(i)) = UCase(Mots(j)) Then
                test = True
            End If
        Next
        If test = False Then
```

```

        m(k) = Mots(i)
        k = k + 1
    End If
Next
ReDim A(k - 1)
'initialisation des nombres d'occurrences à zéro
For i = 0 To k - 1
    A(i) = 0
Next
'on compte le nombre d'occurrence de chaque mot
ReDim Preserve m(k - 1)

For i = 0 To k - 1
    For j = 0 To n
        If UCase(m(i)) = UCase(Mots(j)) Then
            A(i) += 1
        End If
    Next
Next
'on affiche que les mots qui sont repris plusieurs fois
Console.WriteLine("Voici les mots qui sont repris plusieurs fois")
test = False
For i = 0 To k - 1
    If A(i) >= 2 Then
        Console.WriteLine(" Le mot " & UCase(m(i)) & " est repris " & A(i) & "
fois")
        test = True
    End If
Next
If test = False Then
    Console.WriteLine("Aucun mot n'est repris plusieurs fois")
End If
Console.ReadLine()

End Sub

End Module

```

EXERCICE 31

Ecrire un programme Basic qui lit le nom, prénom et salaire de chacun de 200 travailleurs d'une entreprise. Ensuite le programme trie et affiche cette liste par ordre décroissant des salaires, puis des noms en cas de montants de salaire identiques. (Prévoir une structure et le montant de salaire ne peut pas être inférieure à 0).

Résolution

Code VB.Net

```

Module Module1
    Structure Employe
        Dim nom As String
        Dim prenom As String

```



```

    Dim salaire As Integer
End Structure
Sub Main()
    Dim liste(5) As Employe
    Dim a As Employe
    Dim i, j As Integer
    'Lecture des informations
    Console.WriteLine("Lecture des informations sur les employés")
    For i = 0 To 199
        Console.WriteLine("Entrez les informations de l'employé n° " & i + 1)
        Console.Write("Nom: ")
        liste(i).nom = Console.ReadLine()
        Console.Write("Prénom: ")
        liste(i).prenom = Console.ReadLine()
5:    Console.Write("salaire: ")
        liste(i).salaire = CInt(Console.ReadLine())
        If liste(i).salaire < 0 Then
            Console.WriteLine("Le salaire est incorrect, recommencez")
            GoTo 5
        End If
    Next
    'tri
    For i = 199 To 1 Step -1
        For j = 0 To i - 1
            If liste(j).salaire < liste(j +
1).salaire Then
                a = liste(j)
                liste(j) = liste(j + 1)
                liste(j + 1) = a
            ElseIf liste(j).salaire = liste(j +
1).salaire Then
                If liste(j).nom < liste(j + 1).nom Then
                    a = liste(j)
                    liste(j) = liste(j + 1)
                    liste(j + 1) = a
                End If
            End If
        Next
    Next
    'on affiche la nouvelle liste
    Console.WriteLine("Voici la liste des employés selon le salaire")
    Console.WriteLine("Nom" & Space(10) & "Prénom" & Space(10) & " Salaire")
    For i = 0 To 199
        Console.WriteLine(liste(i).nom & Space(10) & liste(i).prenom & Space(10) &
liste(i).salaire)
    Next
    Console.ReadLine()
End Sub

End Module

```

EXERCICE 32

Soit un tableau B (7,7) constitué des valeurs numériques. Ecrire un programme Basic qui après la lecture des données, calcule :

- La somme de chaque ligne
- La somme de chaque colonne
- La somme totale de tous les éléments du tableau

Résolution

Code VB.Net

Module Module1

```
Sub Main()
    Dim B(7, 7) As Integer
    Dim i, j As Integer
    Dim somLigne(7) As Integer
    Dim somColonne(7) As Integer
    Dim somme As Integer
    'Lecture des éléments
    For i = 0 To 7
        For j = 0 To 7
            Console.WriteLine("Entrez l'élément B(" & i & "," & j & ")")
            B(i, j) = CInt(Console.ReadLine())
        Next
    Next

    'calcul de la somme de chaque ligne
    For i = 0 To 7
        somLigne(i) = 0
        For j = 0 To 7
            somLigne(i) += B(i, j)
        Next
    Next

    'calcul de la somme de chaque colonne
    For j = 0 To 7
        somColonne(j) = 0
        For i = 0 To 7
            somColonne(j) += B(i, j)
        Next
    Next

    'calcul de la somme total
    somme = 0
    For i = 0 To 7
        For j = 0 To 7
            somme += B(i, j)
        Next
    Next

    'affichage des résultats
    For i = 0 To 7
        Console.WriteLine("La somme de la " & i & "e ligne est: " & somLigne(i))
    Next
End Sub
```

```

    For j = 0 To 7
        Console.WriteLine("La somme de la " & i & "e colonne est: " & somColonne(j))
    Next
    Console.WriteLine("La somme totale est: " & somme)
    Console.ReadLine()
End Sub

```

```
End Module
```

EXERCICE 33

Ecrire un programme Basic qui donne la liste de tous les nombres de AMSTRONG compris entre 1 et 2000. (Les nombres de AMSTRONG sont des nombres qui sont égaux à la somme des cubes des chiffres qui les composent. Par exemple 153 est un nombre de AMSTRONG, car $153 = 1^3 + 5^3 + 3^3$)

Résolution

Code VB.Net

```
Module Module1
```

```

Sub Main()
    Dim n, i, j, k As Integer
    Dim chaine As String
    Dim somme As Long
    Console.WriteLine("Voici les nombres de AMSTRONG compris entre 1 et 2000")

    For i = 1 To 2000
        'on affecte chaque nombre à la variable chaine pour une manipulation aisée
        chaine = Str(i)
        n = Len(chaine)
        somme = 0
        For j = 1 To n
            k = Val(Mid(chaine, j, 1))
            somme += k ^ 3
        Next
        If i = somme Then
            Console.WriteLine(i.ToString)
        End If
    Next
    Console.ReadLine()
End Sub

```

```
End Module
```

EXERCICE 34

Une entreprise rémunère ses représentants commerciaux à la commission. Ceux-ci reçoivent 200\$ par semaine plus 9% de leurs ventes brutes hebdomadaires. Par exemple, un représentant qui totalise des ventes brutes hebdomadaires de 5000\$ reçoit 200\$ plus 9% de 5000\$, soit un total de 650\$. Ecrivez un programme Basic qui déterminera combien de représentants ont perçu des émoluments situés dans chacune des échelles salariales ci-dessus et l'échelle salariale qui a le nombre le moins élevé (précisez que le salaire de chaque représentant est arrondi à l'entier le plus près).

\$200-\$299	\$300-\$399	\$400-\$499
\$500-\$599	\$600-\$699	\$700-\$799
\$800-\$899	\$900-\$999	\$1000 et plus

Résolution

Code VB.Net

Module Module1

```
Sub Main()
    Dim n, i, min, pos As Integer
    'la variable qui va stocker le nombre de représentants dans chaque échelle
    Dim nombre() As Integer = {0, 0, 0, 0, 0, 0, 0, 0, 0}
    Console.WriteLine("Entrez le nombre de représentants")
    n = CInt(Console.ReadLine())
    Dim ventes(n - 1) As Integer
    Dim salaire(n - 1) As Integer
    'Lecture des ventes brutes de chaque représentant
    For i = 0 To n - 1
        Console.WriteLine("Entrez les ventes brutes du représentant n ° " & i + 1)
        ventes(i) = CInt(Console.ReadLine())
    Next
    'calcul de salaire
    For i = 0 To n - 1
        salaire(i) = Int(200 + 0.09 * ventes(i))
    Next
    'Determination de nombre de représentants dans chaque échelle
    For i = 0 To n - 1
        If salaire(i) >= 200 And salaire(i) <= 299 Then
            nombre(0) += 1
        ElseIf salaire(i) >= 300 And salaire(i) <= 399 Then
            nombre(1) += 1
        ElseIf salaire(i) >= 400 And salaire(i) <= 499 Then
            nombre(2) += 1
        ElseIf salaire(i) >= 500 And salaire(i) <= 599 Then
            nombre(3) += 1
        ElseIf salaire(i) >= 600 And salaire(i) <= 699 Then
            nombre(4) += 1
        ElseIf salaire(i) >= 700 And salaire(i) <= 799 Then
            nombre(5) += 1
        End If
    Next
End Sub
```

```

    ElseIf salaire(i) >= 800 And salaire(i) <= 899 Then
        nombre(6) += 1
    ElseIf salaire(i) >= 900 And salaire(i) <= 999 Then
        nombre(7) += 1
    ElseIf salaire(i) >= 1000 Then
        nombre(8) += 1
    End If
Next
'determination de l'effectif le plus faible
min = nombre(0)
pos = 0
For i = 1 To 8
    If nombre(i) < min Then
        min = nombre(i)
        pos = i
    End If
Next
'Affichage résultats
Console.WriteLine(nombre(0) & " représentant(s) a(ont) un
salaire situé entre 200-299")
Console.WriteLine(nombre(1) & " représentant(s) a(ont) un
salaire situé entre 300-399")
Console.WriteLine(nombre(2) & " représentant(s) a(ont) un
salaire situé entre 400-499")
Console.WriteLine(nombre(3) & " représentant(s) a(ont) un
salaire situé entre 500-599")
Console.WriteLine(nombre(4) & " représentant(s) a(ont) un
salaire situé entre 600-699")
Console.WriteLine(nombre(5) & " représentant(s) a(ont) un
salaire situé entre 700-799")
Console.WriteLine(nombre(6) & " représentant(s) a(ont) un
salaire situé entre 800-899")
Console.WriteLine(nombre(7) & " représentant(s) a(ont) un
salaire situé entre 900-999")
Console.WriteLine(nombre(8) & " représentant(s) a(ont) un
salaire supérieur ou égal à 1000")
Select Case pos
    Case 0
        Console.WriteLine("C'est l'échelle 200-299 qui a moins de représentants ")
    Case 1
        Console.WriteLine("C'est l'échelle 300-399 qui a moins de représentants ")
    Case 2
        Console.WriteLine("C'est l'échelle 400-499 qui a moins de représentants ")
    Case 3
        Console.WriteLine("C'est l'échelle 500-599 qui a moins de représentants ")
    Case 4
        Console.WriteLine("C'est l'échelle 600-699 qui a moins de représentants ")
    Case 5
        Console.WriteLine("C'est l'échelle 700-799 qui a moins de représentants ")
    Case 6
        Console.WriteLine("C'est l'échelle 800-899 qui a moins de représentants ")
    Case 7
        Console.WriteLine("C'est l'échelle 900-999 qui a moins de représentants ")
    Case 8
        Console.WriteLine("C'est l'échelle 1000 et plus qui a moins de
représentants ")
End Select
Console.ReadLine()

```

End Sub

End Module

EXERCICE 35

Le calcul du montant à payer par un malade pour une intervention chirurgicale dans une clinique spécialisée tient compte du cout d'intervention, du cout du matériel, des frais d'anesthésie, de la catégorie du malade et des frais d'hospitalisation. Le cout d'intervention est fonction du type de chirurgie et s'évalue suivant le tableau ci-après :

Types de chirurgies	Coût d'intervention (en \$)
Chirurgie classique	200
Chirurgie plastique	600
Greffe de tissu	650
Microchirurgie	800
Transplantation d'organe	1000/organe

Le cout d'intervention pour transplantation tient par ailleurs compte du nombre d'organes.

Une réduction sur le cout d'intervention sur le cout d'intervention est accordée uniquement aux malades des catégories déterminées dans le tableau ci-après.

Catégorie de malades	Réduction (en %)
Ayant droit	50
Chômeur	40

Les frais d'hospitalisation dépendent de nombre de jours et se calculent comme suit : de 0 à 14 jours, les frais journaliers d'hospitalisation sont fixés à 2\$ et à partir du 15^e jour, ces frais journaliers passent à 1.5\$.

Proposez un programme en Visual basic.Net (Interface et codes) qui permette de calculer et d'afficher le montant à payer pour un malade quelconque.

Attention : L'utilisateur sélectionnera dans des ListBox la catégorie et le type de chirurgie. Après sélection du type de chirurgie, le cout d'intervention correspondant apparaîtra automatiquement dans la zone de texte associée qui est en lecture seule. Si le type de chirurgie est la transplantation d'organes, l'étiquette nombre d'organes sera activée ainsi que sa zone de texte pour autoriser la lecture d'une valeur. Le cout matériel et le nombre de jours d'hospitalisation peuvent être saisis ou sélectionnés à travers des ComboBox correspondants.

Après un clic sur le bouton « Calcul », le programme calculera et affichera le montant à payer ainsi que les autres montants (réduction, frais d'hospitalisation).

Solution

Code VB.Net

```
Public Class Form1
    Private Sub Form1_Load(sender As Object, e As EventArgs) Handles MyBase.Load
        ListBox1.Items.Add("Chirurgie classique")
        ListBox1.Items.Add("Chirurgie plastique")
        ListBox1.Items.Add("Grefe de tissu")
        ListBox1.Items.Add("Microchirurgie")
        ListBox1.Items.Add("Transplantation d'organe")
        ListBox2.Items.Add("Ayant droit")
        ListBox2.Items.Add("Chomeur")
        ListBox2.Items.Add("Autres")
        TxtCoutInter.ReadOnly = True
        Label_nbreOrgane.Enabled = False
        TxtnbreOrgane.Enabled = False
        TxtNP.ReadOnly = True
        TxtReduction.ReadOnly = True
        TxtFraisHosp.ReadOnly = True
    End Sub

    Private Sub ListBox1_SelectedIndexChanged(sender As Object, e As EventArgs) Handles
        ListBox1.SelectedIndexChanged
        If ListBox1.SelectedItem = "Chirurgie classique" Then
            TxtCoutInter.Text = 200
        End If
    End Sub
End Class
```

```

ElseIf ListBox1.SelectedItem = "Chirurgie plastique" Then
    TxtCoutInter.Text = 600
ElseIf ListBox1.SelectedItem = "Grefe de tissu" Then
    TxtCoutInter.Text = 650
ElseIf ListBox1.SelectedItem = "Microchirurgie" Then
    TxtCoutInter.Text = 800
ElseIf ListBox1.SelectedItem = "Transplantation d'organe" Then
    Label_nbreOrgane.Enabled = True
    TxtnbreOrgane.Enabled = True
End If
End Sub

Private Sub TxtnbreOrgane_TextChanged(sender As Object, e As EventArgs) Handles
TxtnbreOrgane.TextChanged
    Dim Nbre As Integer
    Nbre = CInt(TxtnbreOrgane.Text)
    TxtCoutInter.Text = (1000 * Nbre).ToString
End Sub

Private Sub BtnCalcul_Click(sender As Object, e As EventArgs) Handles BtnCalcul.Click
    Dim coutInter, coutMat, fraisAnes, fraisHosp, nbre, red, np As Integer
    Dim cat As String
    'recuperation des informations
    coutInter = Val(TxtCoutInter.Text)
    coutMat = Val(ComboCoutMat.Text)
    fraisAnes = Val(TxtFraisAnest.Text)
    nbre = Val(ComboNbreJrs.Text)
    cat = ListBox2.SelectedItem
    If ListBox1.SelectedItem = "" Or ListBox2.SelectedItem = "" Or
ComboCoutMat.Text = "" Or ComboNbreJrs.Text = "" Or TxtFraisAnest.Text = "" Then
        Console.WriteLine("Veuillez remplir tous les champs")
    Else
        'calcul de frais d'hospitalisation
        Select Case (nbre)
            Case 0 To 14
                fraisHosp = nbre * 2
            Case Is > 14
                fraisHosp = 14 * 2 + (nbre - 14) * 1.5
            Case Else
                MsgBox("Nombre de jours incorrects")
        End Select
        TxtFraisHosp.Text = fraisHosp.ToString
        'calcul de la reduction
        If cat = "Ayant droit" Then
            red = coutInter * 0.5
        ElseIf cat = "Chomeur" Then
            red = coutInter * 0.4
        Else
            red = 0
        End If
        TxtReduction.Text = red.ToString
        np = coutInter + coutMat + fraisAnes + fraisHosp - red
        TxtNP.Text = np.ToString
    End If
End Sub

Private Sub BtnAnnuler_Click(sender As Object, e As EventArgs) Handles
BtnAnnuler.Click

```



```

        ListBox1.SelectedIndex = -1
        ListBox2.SelectedIndex = -1
        ComboCoutMat.Text = ""
        ComboNbreJrs.Text = ""
        TxtFraisAnest.Text = ""
        TxtCoutInter.Text = ""
        TxtNP.Text = ""
        TxtReduction.Text = ""
        TxtFraisHosp.Text = ""
        TxtnbreOrgane.Text = ""
    End Sub

    Private Sub BtnQuitter_Click(sender As Object, e As EventArgs) Handles
        BtnQuitter.Click
        End
    End Sub
End Class

```

EXERCICE 36

Voici l'interface :

Un magasin de la place organise le paiement de ses employés à l'aide des rubriques : Salaire de base (Sb) et prime familiale (Pf). Les salaires de base exprimés en unité monétaire sont déterminés à partir des grades suivant le tableau ci-après :

Grade	Salaire de base
Gérant	20
Comptable	18
Caissier	16
Facturier	15
Livreur	13
Vendeur	11

La prime familiale est composée de deux parties : de la prime mariage (pm) et de la prime enfant (pe). La prime mariage est octroyée à tous les agents qui ne sont pas célibataires (c.à.d. mariés, divorcés ou veufs). Elle vaut 2% du salaire de base de l'agent. Ainsi, un agent de grade Facturier marié obtient une prime mariage de $15 \times 2\% = 0.30$ tandis qu'un agent de grade gérant célibataire n'obtient aucune prime mariage.

La prime enfant est calculée en fonction du nombre d'enfants vivants de moins de 19 ans. Elle est calculée de la manière suivante : jusqu'à 5 enfants, chacun génère une prime de 0.05, à partir du 6e enfant la prime devient 0.02 par enfant. Enfin, pas de prime à partir du 11e enfant.

On vous demande de gérer le bouton « Calculer » pour calculer la prime mariage, la prime enfant et le salaire net, le bouton « Initialiser » pour réinitialiser les objets et « Quitter » pour quitter le programme.

Solution

Code VB.Net

```
Public Class Form1
    Private Sub Form1_Load(sender As Object, e As EventArgs) Handles MyBase.Load
        ComboGrade.Items.Add("Gerant")
        ComboGrade.Items.Add("Comptable")
        ComboGrade.Items.Add("Caissier")
        ComboGrade.Items.Add("Facturier")
        ComboGrade.Items.Add("Livreur")
        ComboGrade.Items.Add("Vendeur")
        ComboEtat.Items.Add("Marié(e)")
        ComboEtat.Items.Add("Celibataire")
        ComboEtat.Items.Add("veuf(ve)")
        ComboEtat.Items.Add("Divorcé(e)")
        TxtPe.ReadOnly = True
        TxtPm.ReadOnly = True
        TxtSn.ReadOnly = True
        TextSB.ReadOnly = True
    End Sub

    Private Sub ComboGrade_SelectedIndexChanged(sender As Object, e As EventArgs) Handles
        ComboGrade.SelectedIndexChanged
        Dim grade = ComboGrade.SelectedItem
        If grade = "Gerant" Then
            TextSB.Text = 20
        End If
    End Sub
End Class
```

```

ElseIf grade = "Comptable" Then
    TextSB.Text = 18
ElseIf grade = "Caissier" Then
    TextSB.Text = 16
ElseIf grade = "Facturier" Then
    TextSB.Text = 15
ElseIf grade = "Livreur" Then
    TextSB.Text = 13
ElseIf grade = "Vendeur" Then
    TextSB.Text = 11
End If
End Sub

Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
    Dim sn, sb, pe, pm, nef, pf As Single
    Dim etat As String
    'récuperation des informations
    nef = Val(Txtnbre.Text)
    sb = Val(TextSB.Text)
    etat = ComboEtat.Text
    If etat = "Celibataire" Then
        pm = 0
    Else
        pm = 0.02 * sb
    End If
    Select Case (nef)
        Case 0 To 5
            pe = nef * 0.05
        Case 6 To 10
            pe = 5 * 0.05 + (nef - 5) * 0.02
        Case Is > 10
            pe = 5 * 0.05 + 5 * 0.02
        Case Else
            MsgBox("Nombre d'enfants incorrect", MsgBoxStyle.OkOnly +
MsgBoxStyle.Information, "ERREUR ")
    End Select
    pf = pe + pm
    sn = sb + pf
    TxtPe.Text = pe.ToString
    TxtPm.Text = pm.ToString
    TxtSn.Text = sn.ToString
End Sub

Private Sub Button2_Click(sender As Object, e As EventArgs) Handles Button2.Click
    Txtnbre.Text = ""
    TxtPe.Text = ""
    TxtPm.Text = ""
    Txtnbre.Text = ""
    ComboEtat.Text = ""
    ComboGrade.Text = ""
    TextSB.Text = ""
    TxtSn.Text = ""
End Sub

Private Sub Button3_Click(sender As Object, e As EventArgs) Handles Button3.Click
    End
End Sub
End Class

```

EXERCICE 37

Soit l'interface utilisateur ci-après sur la facturation de la location véhicule :

The screenshot shows a Windows application window titled "Aimé DIUMI". The window contains a form for car rental. The form is divided into two main sections. The top section has three dropdown menus on the left labeled "Catégorie", "Prix horaire", and "Nombre d'heures". To the right of these is a large image of a silver sports car. The bottom section has two input fields on the right labeled "Prix but" and "Réduction". On the left of this section is a button labeled "Prix net" next to an empty input field. At the very bottom of the window are two buttons: "Annuler" and "Fermer".

Proposer des codes associés aux 3 boutons de commande, tels qu'en cliquant sur « Prix Net », le programme calcule et affiche le prix brut (sans réduction en fonction du prix horaire et du nombre d'heures), la réduction (15% du prix brut pour seulement la catégorie « Privilégié » et pas pour « autres » et le prix net (tenant compte de la réduction), en cliquant sur « Annuler », le programme est réinitialisé et en cliquant sur « Fermer », le programme s'arrête. (Indication : nommer les objets concernés par les initiales de leurs noms).

Solution

Code VB.Net

```
Public Class Form1
    Private Sub Form1_Load(sender As Object, e As EventArgs) Handles MyBase.Load
        ComboCat.Items.Add("Privilegié")
        ComboCat.Items.Add("Autres")
        ComboNbre.Items.Add("1")
        ComboNbre.Items.Add("10")
        ComboNbre.Items.Add("14")
        ComboNbre.Items.Add("12")
        ComboNbre.Items.Add("31")
    End Sub
End Class
```

```

        ComboNbre.Items.Add("11")
        ComboNbre.Items.Add("5")
        ComboNbre.Items.Add("6")
        ComboNbre.Items.Add("3")
        ComboNbre.Items.Add("7")
        TxtNp.ReadOnly = True
        TxtPb.ReadOnly = True
        TxtRed.ReadOnly = True
    End Sub

    Private Sub BtnPn_Click(sender As Object, e As EventArgs) Handles BtnPn.Click
        Dim cat As String
        Dim ph, nbre, prixBrut As Integer
        Dim reduction, np As Single
        'recuperation des informations
        cat = ComboCat.Text
        nbre = Val(ComboNbre.Text)
        ph = Val(TxtPrixH.Text)
        prixBrut = nbre * ph
        TxtPb.Text = prixBrut.ToString
        If cat = "Privilegié" Then
            reduction = prixBrut * 0.15
        Else
            reduction = 0
        End If
        TxtRed.Text = reduction.ToString
        np = prixBrut - reduction
        TxtNp.Text = np.ToString
    End Sub

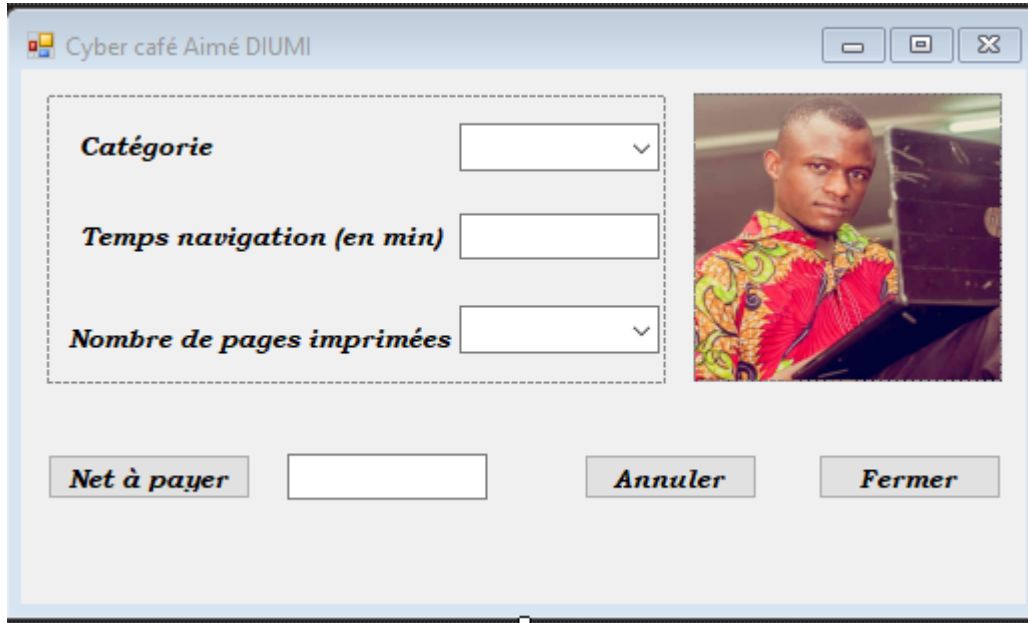
    Private Sub BtnAnnuler_Click(sender As Object, e As EventArgs) Handles
BtnAnnuler.Click
        TxtNp.Text = ""
        TxtPb.Text = ""
        TxtPrixH.Text = ""
        TxtRed.Text = ""
        ComboCat.Text = ""
        ComboNbre.Text = ""
    End Sub

    Private Sub BtnFermer_Click(sender As Object, e As EventArgs) Handles BtnFermer.Click
        Environment.Exit(0)
    End Sub
End Class

```

EXERCICE 38

Soit l'interface utilisateur ci-après sur la facturation d'un cyber café



Proposez des codes associés aux trois boutons de commande, tels qu'en cliquant sur « Net à payer », le programme calcule et affiche le net à payer qui est constitué du cout de navigation (le cout de navigation par minute est 10Fc pour la catégorie « Etudiant » et 15Fc pour la catégorie « Autres ») et du cout d'impression (L'impression d'une page vaut 50Fc). En cliquant sur « Annuler », le programme est réinitialisé et en cliquant sur « Fermer », le programme s'arrête. (Indication : nommer les objets concernés par les initiales de leurs noms).

Solution

Code VB.Net

```
Public Class Form1
    Private Sub Form1_Load(sender As Object, e As EventArgs) Handles MyBase.Load
        ComboCat.Items.Add("Etudiant")
        ComboCat.Items.Add("Autre")
        ComboNbre.Items.Add("1")
        ComboNbre.Items.Add("2")
        ComboNbre.Items.Add("3")
        ComboNbre.Items.Add("4")
        ComboNbre.Items.Add("5")
        ComboNbre.Items.Add("6")
        ComboNbre.Items.Add("7")
        ComboNbre.Items.Add("8")
        ComboNbre.Items.Add("9")
        ComboNbre.Items.Add("10")
    End Sub
End Class
```

```

        ComboNbre.Items.Add("11")
        ComboNbre.Items.Add("12")
        ComboNbre.Items.Add("13")
        ComboNbre.Items.Add("14")
        ComboNbre.Items.Add("15")
        TxtNp.ReadOnly = True
    End Sub

    Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
        Dim cat As String
        Dim temps, nbre, np As Integer
        'recuperation des informations
        cat = ComboCat.SelectedItem
        temps = TxtTemps.Text
        nbre = ComboNbre.Text
        'calcul du net à payer
        If cat = "Etudiant" Then
            np = 10 * temps + nbre * 50
        ElseIf cat = "Autre" Then
            np = 15 * temps + nbre * 50
        End If
        TxtNp.Text = np.ToString
    End Sub

    Private Sub BtnAnnuler_Click(sender As Object, e As EventArgs) Handles
BtnAnnuler.Click
        ComboCat.Text = ""
        ComboNbre.Text = ""
        TxtNp.Text = ""
        TxtTemps.Text = ""
    End Sub

    Private Sub BtnFermer_Click(sender As Object, e As EventArgs) Handles BtnFermer.Click
        Dim i As Integer
        i = MsgBox("voulez vous réellement quitter?", MsgBoxStyle.YesNo +
MsgBoxStyle.Question, "CYBER CAFE")
        If i = 6 Then
            End
        End If
    End Sub
End Class

```

EXERCICE 39

L'hôtel HB vous confie la conception du calcul du montant à payer par ses clients et vous informe que ce montant est constitué des rubriques suivantes : logement, restauration, transport et buanderie. Ses clients sont organisés en catégories suivantes : Clergé, agent de l'état et autres. Le cout du logement est fonction du type de chambre et la tarification est donnée par le tableau ci-après :

Type de chambre	Coût de logement par jour (en \$)
Chambre ordinaire	30
Chambre V.I.P	80
Appartement 4 pièces	90
Appartement 6 pièces	100

Le cout de logement journalier connaît une réduction qui tienne compte du nombre de jours telle :

- Pour les gens des catégories « Clergé » et « Agent de l'état », à partir du 6e jour ce cout de logement journalier connaît une réduction de 20%.
- Pour les gens de la catégorie « Autres », à partir du 11e jour ce cout connaît une réduction de 15%.

Proposez un programme en Visual Basic.Net (Interface et code) qui permette de calculer et afficher le montant à payer pour un client quelconque à la fin de son séjour.

Attention : L'utilisateur sélectionnera dans des ListBox la catégorie et le type de chambre. Après sélection du type de chambre, le cout de logement journalier apparait automatiquement dans la zone de texte associée qui est seulement en lecture.

Si le type de chambre est Chambre ordinaire ou Chambre V.I.P, la case à cocher « Accompanyer d'un enfant » sera activée pour autoriser si nécessaire sa coche. Les couts de restauration, transport et buanderie peuvent être saisis ou sélectionnés à travers des ComboBox correspondants.

En cas d'accompagnement par un enfant, le complément de 50% est effectué sur le cout de logement.

Après un clic sur le bouton « Calcul », le programme calculera et affichera le montant à payer ainsi que le montant total de logement.

Solution

The screenshot shows a Windows application window titled "Aimé DIUMI". The window contains a form with the following elements:

- Informations** section:
 - Catégorie clients**: A list box labeled *ListCat*.
 - Type chambre**: A list box labeled *ListTypeChambre*.
 - cout de logement/jr**: A text box.
 - Nombre de jours**: A text box.
 - Accompagné d'un enfant**: A checkbox.
- Services** section (indicated by a dashed border):
 - Restauration**: A dropdown menu.
 - Transport**: A dropdown menu.
 - Buanderie**: A dropdown menu.
- Calculs** section (indicated by a dashed border):
 - Cout de logement**: A text box.
 - Net à payer**: A text box.
- Buttons**: **Calcul**, **Annuler**, and **Fermer**.

Code VB.Net

```
Public Class Form1
    Private Sub Form1_Load(sender As Object, e As EventArgs) Handles MyBase.Load
        'chargement des listes
        ListTypeChambre.Items.Add("Chambre ordinaire")
        ListTypeChambre.Items.Add("Chambre V.I.P")
        ListTypeChambre.Items.Add("Appartement 4 pièces")
        ListTypeChambre.Items.Add("Appartement 6 pièces")
        ListCat.Items.Add("Clergé")
        ListCat.Items.Add("Agent de l'état")
        ListCat.Items.Add("Autres")
        TxtCoutLogJr.ReadOnly = True
        TxtNp.ReadOnly = True
        TxtCoutlog.ReadOnly = True
        CheckAcc.Enabled = False
    End Sub

    Private Sub ListTypeChambre_SelectedIndexChanged(sender As Object, e As EventArgs)
Handles ListTypeChambre.SelectedIndexChanged
        Dim type As String
        type = ListTypeChambre.SelectedItem
        If type = "Chambre ordinaire" Then
            TxtCoutLogJr.Text = 30
            CheckAcc.Enabled = True
        ElseIf type = "Chambre V.I.P" Then
            TxtCoutLogJr.Text = 80
        End If
    End Sub
End Class
```

```

        CheckAcc.Enabled = True
    ElseIf type = "Appartement 4 pièces" Then
        TxtCoutLogJr.Text = 90
    ElseIf type = "Appartement 6 pièces" Then
        TxtCoutLogJr.Text = 100
    End If
End Sub

Private Sub BtnCalcul_Click(sender As Object, e As EventArgs) Handles BtnCalcul.Click
    Dim logementTotal, restauration, trans, buand, nbreJr, logJ, np As Integer
    Dim cat As String
    If ComboBuanderie.Text = "" Or TxtCoutLogJr.Text = "" Or TxtNbreJrs.Text = "" Or
    ComboRestauration.Text = "" Or ComboTransport.Text = "" Or TxtCoutLogJr.Text = "" Or
    ListCat.SelectedItem = "" Or ListTypeChambre.SelectedItem = "" Then
        MsgBox("veuillez remplir tous les champs", MsgBoxStyle.OkOnly +
    MsgBoxStyle.Information, "HOTEL HB")
    Else
        'recuperation des informations
        cat = ListCat.SelectedItem
        restauration = Val(ComboRestauration.Text)
        buand = Val(ComboBuanderie.Text)
        trans = Val(ComboTransport.Text)
        nbreJr = Val(TxtNbreJrs.Text)
        logJ = Val(TxtCoutLogJr.Text)
        'calcul du cout de logement total
        If cat = "Clergé" Or cat = "Agent de l'état" Then
            Select Case (nbreJr)
                Case 0 To 5
                    logementTotal = logJ * nbreJr
                Case Is >= 6
                    logementTotal = 5 * logJ + (logJ * (1 - 0.2)) * (nbreJr - 5)
            End Select
        Else
            Select Case (nbreJr)
                Case 0 To 10
                    logementTotal = logJ * nbreJr
                Case Is >= 11
                    logementTotal = 10 * logJ + (logJ * (1 -
0.15)) * (nbreJr - 10)
            End Select
        End If
        If CheckAcc.Checked = True Then
            logementTotal = logementTotal * (1 + 0.5)
        End If
        'on affiche le cout de logement total
        txtCoutlog.Text = logementTotal.ToString
        'on calcule le Net à payer
        np = buand + logementTotal + restauration + trans
        TxtNp.Text = np.ToString
    End If
End Sub

Private Sub BtnAnnuler_Click(sender As Object, e As EventArgs) Handles
BtnAnnuler.Click
    ComboBuanderie.Text = ""
    TxtCoutLogJr.Text = ""
    TxtNbreJrs.Text = ""
    ComboRestauration.Text = ""

```

```

    ComboTransport.Text = ""
    TxtCoutLogJr.Text = ""
    ListCat.SelectedItem = ""
    ListTypeChambre.SelectedItem = ""
    TxtNp.Text = ""
    TxtCoutlog.Text = ""
    CheckAcc.Checked = False
End Sub

Private Sub BtnFermer_Click(sender As Object, e As EventArgs) Handles BtnFermer.Click
    Dim i As Integer
    i = MsgBox("Vous voulez réellement quitter?", MsgBoxStyle.YesNo +
MsgBoxStyle.Question, "HOTEL HB")
    If i = 6 Then
        End
    End If
End Sub
End Class

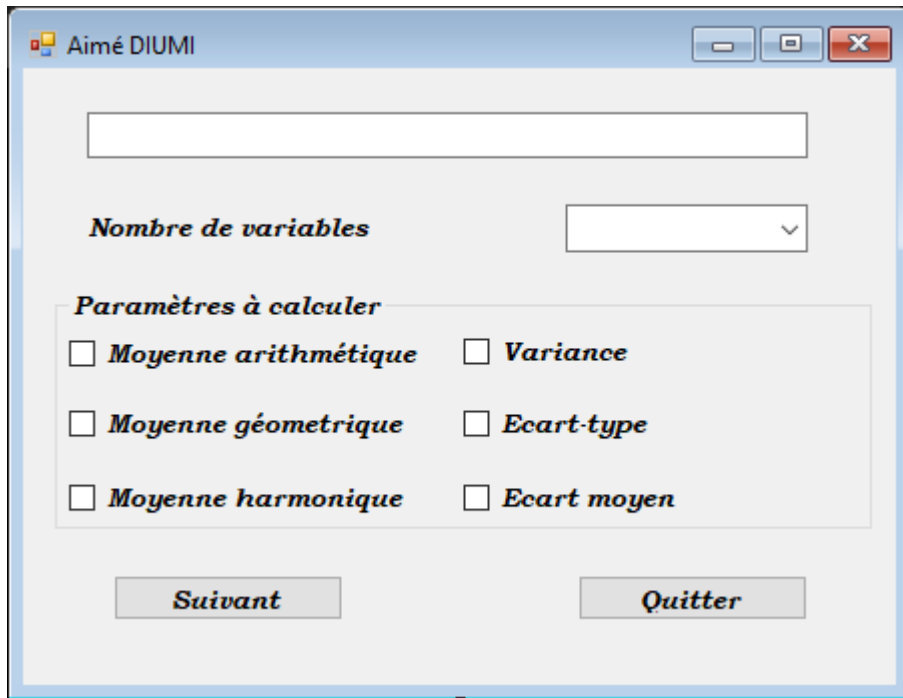
```

EXERCICE 40

Concevoir un programme en VB de statistique qui calcule la moyenne (arithmétique, géométrique et harmonique), la variance, l'écart-type et l'écart-moyen, en suivant les indications suivantes :

- Définir une interface qui permette de : lire le titre du tableau (zone de texte), lire ou sélectionner le nombre de variables (liste combinée), cocher les paramètres à calculer (case à cocher), lancer la suite du programme ou arrêter l'exécution (Boutons de commande).
- Les autres données sont ensuite lues à travers une boîte d'entrée (InputBox).
- Après calcul, le résultat s'affiche dans une boîte de message.

Solution



Code VB.Net

```
Public Class Form1
    Private Sub Form1_Load(sender As Object, e As EventArgs) Handles MyBase.Load
        ComboBox1.Items.Add(5)
        ComboBox1.Items.Add(10)
        ComboBox1.Items.Add(15)
        ComboBox1.Items.Add(20)
        ComboBox1.Items.Add(25)
        ComboBox1.Items.Add(30)
        ComboBox1.Items.Add(35)
    End Sub

    Private Sub BtnSuivant_Click(sender As Object, e As EventArgs) Handles
BtnSuivant.Click
        Dim n As Integer
        Dim somme, produit, inverse, diff1, diff2 As Long
        Dim m, g, h, var, s, ecartMoyen As Single
        n = Val(ComboBox1.Text)
        If Val(ComboBox1.Text) = 0 Or n <= 0 Then
            MsgBox("nombre d'observation incorrect")
        Else
            Dim A(n - 1) As Integer
            For i As Integer = 0 To n - 1
                A(i) = Val(InputBox("entrez la " & i + 1 & " e
observation", "LECTURE DES DONNEES"))
            Next
            'calcul de la somme et produit et la somme des inverses
            somme = 0
            produit = 1
            inverse = 0
        End If
    End Sub
End Class
```

```

For i As Integer = 0 To n - 1
    somme += A(i)
    produit *= A(i)
    inverse += 1 / A(i)
Next
m = somme / n
g = produit * (1 / n)
h = n / inverse
'calcul de la variance, ecart type et ecart moyen
diff1 = 0
diff2 = 0
For i As Integer = 0 To n - 1
    diff1 += (A(i) - m) ^ 2
    diff2 += Math.Abs((A(i) - m))
Next
var = diff1 / n
s = var * (1 / 2)
ecartMoyen = diff2 / n
If ChecEcartType.Checked = True Then
    MsgBox("L'ecart type vaut " & s, MsgBoxStyle.OkOnly +
MsgBoxStyle.Information, "RESULTAT")
End If
If CheckEcartMoyen.Checked = True Then
    MsgBox("L'ecart moyen vaut " & ecartMoyen, MsgBoxStyle.OkOnly +
MsgBoxStyle.Information, "RESULTAT")
End If
If CheckMoyenneA.Checked = True Then
    MsgBox("La moyenne arithmetique vaut " & m, MsgBoxStyle.OkOnly +
MsgBoxStyle.Information, "RESULTAT")
End If
If CheckMoyenneG.Checked = True Then
    MsgBox("La moyenne geometrique vaut " & g, MsgBoxStyle.OkOnly +
MsgBoxStyle.Information, "RESULTAT")
End If
If CheckMoyenneH.Checked = True Then
    MsgBox("La moyenne harmonique vaut " & h, MsgBoxStyle.OkOnly +
MsgBoxStyle.Information, "RESULTAT")
End If
If CheckVariance.Checked = True Then
    MsgBox("La variance vaut " & var, MsgBoxStyle.OkOnly +
MsgBoxStyle.Information, "RESULTAT")
End If
End If
End Sub

Private Sub BtnQuitter_Click(sender As Object, e As EventArgs) Handles
BtnQuitter.Click
End
End Sub
End Class

```