

## Visual Basique : le langage, projet mode console

Au total faire au moins 10 exercices choisis chacun dans une tranche différente parmi ce qui suit. Vous pouvez travailler avec votre ordinateur mais :

1. Fournir SUR PAPIER ECRIT A LA MAIN :
  - le code source en VB.NET
  - le code source en C
2. Donner en numérique le projet VB.NET et éventuellement le projet en C

Ce travail donne lieu à deux notes :

- une pour ce qui sera rendu à la fin de la séance
- une pour ce qui sera rendu au retour des vacances le lundi 09 mars 2009

### M1.3 Exercices : avoir des variables dans un programme

#### Exercice 1

Combien y a-t-il de types en VB et quels sont-ils ?

Qu'est-ce qui les différencie ?

Donner leurs fourchettes de valeurs.

Par défaut sont-ils signés ou non signés ?

#### Exercice 2

Dans un programme déclarer une variable de chaque type, lui affecter une valeur avec la fonction Rnd() et l'afficher avec Console.WriteLine()

#### Exercice 3

Tous les noms de variable suivants peuvent-ils se trouver dans un même bloc ?

Conserver uniquement ceux qui fonctionnent et sont compatibles dans un même bloc :

```
0t, ti, p0, freste, div-total, tata, t2345, c, cc, C,Cc,TaTa  
Err_, _E_, err_, ERR_, +DIV, +dIv
```

### M2.3 Exercices : branchement, aiguillage

#### Exercice 2

Écrire un programme qui lit deux nombres entiers a et b et donne le choix à l'utilisateur :

1. de savoir si la somme de a+b est paire.
2. de savoir si le produit a\*b est pair.
3. de connaître le signe de a-b;
4. de connaître le signe de a\*b

#### Exercice 4

Écrire un programme affichant un menu proposant de jouer avec un, deux, trois ou quatre dés. Selon le choix fait le programme lance les dés. Les dés identiques sont relancés et il y a un cumul des points. L'utilisateur gagne si : si le total est supérieur au deux tiers du maximum (avec deux dés ça fait  $8 : (12 / 3) * 2$ ). Le programme indique combien il manque pour gagner ou combien il y a de plus.

## M2.4 Exercices : opérateurs logiques ET, OU

### Exercice 2

Le jeu du bandit à bras multiple. Trois roues sont mises en parallèle elles ont sur leur tranche des numéros ou des lettres. Au départ elles sont lancées et tournent à des vitesses différentes lorsqu'elles s'arrêtent trois numéros ou lettres sont alignés : A5R ou 67T, on gagne lorsque certaines combinaisons sortent, par exemple : AAA, BCD, OIO... Faire un programme pour tenter sa chance un seul coup.

### Exercice 4

Donnez l'algorithme qui détermine le nombre de valeurs distinctes parmi trois variables à faire saisir par l'utilisateur. (ex : 8, 8 et 8 saisi par l'utilisateur donne 1 valeur distincte...) Comment faire avec 4 variables ?

## M2.5 Exercices : mettre en place des boucles

### Exercice 3

Tirage aux dés. Écrivez un programme qui simule un nombre n, entré par l'utilisateur, de tirages avec 6 dés. Le programme compte et affiche à la fin le nombre de coups où tous les dés ont eu la même valeur.

### Exercice 4

Dichotomie. Écrire un programme qui génère aléatoirement un nombre que l'utilisateur devra découvrir. A chaque saisie, le programme dira si le nombre est plus petit ou plus grand. Si le nombre est trouvé par l'utilisateur, le programme lui annoncera qu'il a gagné en indiquant le nombre d'essais qui ont été nécessaires.

### Exercice 8

Écrire un programme qui convertit un entier naturel en chiffres romains, en utilisant l'ancienne notation.

Exemple : 4 ( IIII), 9 ( VIIII), 900 ( DCCCC)

Rappelons les éléments de base :

I : 1, V :5, X : 10, L : 50, C : 100, D : 500, M : 1000.

### Exercice 10

Écrire un programme C qui propose à l'utilisateur de dessiner à l'écran certaines figures composées d'étoiles triangle, carré, sablier, des lettres...

Les figures seront proposées par un menu (triangle, carré, sablier...). La hauteur de la figure sera saisie par l'utilisateur.

Exemple : hauteur=4

```
*****
*****
*       ****
***     ****
*****  ****
*****  ****
*****  ****
*****
*****
*****
```

---

## M2.6 Exercices : menus, boucles d'événements

---

### Exercice 1

Faire un petit convertisseur franc-euro et euro-franc. Le programme propose un menu avec les deux possibilités, à l'issue d'une conversion il propose à l'utilisateur de recommencer ou de quitter (1 euro vaut 6,55759 francs)

### Exercice 6

Jeu de dés à deux joueurs. Chaque joueur à son tour jette le dé autant de fois qu'il veut. Il accumule les points qu'il obtient sauf s'il fait 1 et s'il fait 1, il ne marque aucun point, il est obligé de s'arrêter et c'est à l'autre joueur de jouer. Le gagnant est le premier qui dépasse 100.

Faire un programme pour jouer contre l'ordinateur. Le dé n'est pas pipé. L'ordinateur joue honnêtement quand c'est son tour et lance le dé quand c'est votre tour. Il affiche toujours ce qui sort et quand c'est à vous de jouer il vous demande à chaque lancé si vous voulez continuer. Lorsque la partie est finie il propose une autre partie ou de quitter.

La programmation de ce jeu suppose de concevoir une stratégie pour l'ordinateur. Quand et comment décide-t-il de s'arrêter de lancer le dé ?

---

## M3.2 Exercice : tests tableaux à plusieurs dimensions

---

### Exercice 1

Faire un programme qui déclare un tableau à 3 dimensions de  $2*2*2$  et :

- initialiser ce tableau en stockant à chaque position le numéro de la position (1, 2, 3 ...)
- afficher le résultat.

### Exercice 2

Faire un programme qui déclare deux matrices d'entiers de  $2*3$  et :

- initialisez ces matrices avec des valeurs aléatoires entre 0 et 20
- pour chaque position comparez les valeurs stockées et stockez la plus grande dans une troisième matrice.
- affichez les trois matrices (sur 6 lignes uniquement)

---

## M3.4 Exercices : manipulations de base sur des tableaux

---

### Exercice 5

Remplir un tableau avec des valeurs aléatoires et opérer une rotation des valeurs afin que ce qui est à la position 0 passe à la position 1, ce qui est à la position 1 passe à la position 2 et ainsi de suite jusqu'au dernier qui vient à la position 0.

Faire le même exercice avec une matrice : la valeur de chaque position passe à la position suivante jusqu'à la dernière qui vient en première position.

### Exercice 6

Remplir un tableau avec des valeurs aléatoires, afficher le tableau puis :

- affichez les deux plus grandes valeurs
- affichez la plus petite valeur et sa position dans le tableau
- affichez les valeurs du tableau en ordre croissant sans modifier le tableau
- modifiez le tableau afin que toutes ses valeurs soient triées en ordre décroissant et affichez le résultat.

---

### **M3.5 Exercices : manipulations de lettres et chaînes de caractères**

---

#### Exercice 1

A partir d'un menu proposé à l'utilisateur faire le programme suivant :

- Saisir une chaîne de caractères
- Compter la longueur d'une chaîne
- Convertir la chaîne saisie en majuscules. Seuls les caractères en minuscule sont modifiés les autres sont évités.
- Convertir la chaîne saisie en minuscules
- Comparer 2 chaînes à saisir. Il s'agit de dire si elles sont identiques et laquelle est la première dans l'ordre lexicographique.
- Concaténer 2 chaînes dans la première (attention aux débordements)
- Crypter une chaîne en appliquant un décalage circulaire dont la valeur est entrée par l'utilisateur
- Décrypter une chaîne de caractère cryptée en appliquant un décalage circulaire inverse à partir d'une valeur entrée par l'utilisateur.

---

### **M4.3 Exercices : manipuler des procédures sans paramètre**

---

#### Exercice 1

Afin de tester la fonction rand() du point de vue statistique, faire une fonction qui tire 100000 fois une valeur comprise entre 0 et 5, compte les occurrences de chaque résultat et affiche le pourcentage de chaque résultat. Appeler la fonction dans un programme.

#### Exercice 2

Faire une fonction damier qui affiche un damier de 20 sur 15 à l'écran. Le damier est encadré en vert. Tester la fonction dans un programme.

#### Exercice 3

Faire une fonction qui affiche l'alphabet en boucle dans un rectangle de 20 lignes sur 40 colonnes. Chaque alphabet aura ses propres couleurs. Tester dans un programme.

---

### **M4.4 Exercices : appels de fonctions sans paramètre**

---

#### Exercice 3

Faire une fonction qui initialise une matrice d'entiers avec des valeurs aléatoires comprises entre 50 et 100. La fonction retourne le nombre de valeurs paires qu'il a eu dans la matrice. Tester dans un programme.

#### Exercice 4

Faire une fonction qui initialise une matrice d'entiers avec des 0 ou des 1 de façon aléatoire. Ensuite pour chaque position compter combien il y a de 1 parmi les 8 positions voisines. Retourner la fréquentation la plus répandue dans la matrice. Tester dans un programme.

---

### **M4.5 Exercices : appels de fonctions, variables simples en paramètre**

---

#### Exercice 3

Faire une fonction qui affiche le caractère (table ASCII) correspondant à une valeur numérique décimale entre 0 et 255. A partir de cette fonction faire une deuxième fonction qui affiche toute la table ASCII.

#### Exercice 4

Faire une fonction qui donne la moyenne de trois nombres quels qu'ils soient (flottants ou non). Tester dans un programme. Le programme quitte uniquement lorsque l'utilisateur le demande.

#### Exercice 5

Soit un barème de l'impôt défini comme suit : pour un ménage X avec un revenu total R et un nombre n de membres du foyer, l'impôt est donné par :

10% de R si  $R/n < 500$  euros

20% de R si  $R/n \geq 500$  euros

- 1) Écrire une fonction qui calcule le montant de l'impôt en fonction de R et de n
- 2) Écrire une fonction qui donne le revenu net d'un ménage après paiement de l'impôt en fonction de R et de n. Tester dans un programme, R et n sont entrés par l'utilisateur ensuite le montant de l'impôt et le revenu net du ménage sont affichés ensuite.

Le programme quitte uniquement lorsque l'utilisateur le demande.

#### Exercice 7

Ecrire une première fonction qui indique si un nombre entier est multiple de 2 ou non.

Ecrire une seconde fonction qui indique si un nombre entier est multiple de 3 ou non.

Utiliser ces deux fonction dans un programme qui lit un nombre entier et précise s'il est pair, multiple de 3 et/ou multiple de 6. Le programme quitte uniquement lorsque l'utilisateur le demande.

Exemple d'exécution :

entrer un nombre : 9  
il est multiple de 3  
Recommencer ? (o/n)

Autre exemple :

entrer un nombre : 12  
il est pair, multiple de 3 et divisible par 6  
Recommencer ? (o/n)

#### Exercice 8

Soit une matrice d'entiers de 15\*20 déclarée dans le main(), remplir la matrice de 0 et de 1 aléatoirement et afficher la matrice. Ecrire une fonction "compte\_voisin()" qui renvoie pour chaque position de la matrice le nombre de position voisine à 1. Appeler ensuite la fonction pour chaque position de la matrice et afficher à chaque fois le nombre trouvé. Le programme quitte uniquement lorsque l'utilisateur le demande.

### **M4.6 Exercices : appels de fonctions, tableaux en paramètre**

#### Exercice 1

Ecrire une fonction qui affiche une lettre sur deux d'une chaîne de caractères donnée en paramètre. Tester la fonction dans un programme

#### Exercice 2

Faire une fonction qui reconnaît et affiche des lettres communes à deux mots ou phrases saisies au clavier. La saisie est elle-même une fonction. Tester dans un programme qui s'arrête à la demande de l'utilisateur.

#### Exercice 3

Faire une fonction qui compte le nombre de répétition des lettres dans un mot ou une phrase entrée par l'utilisateur. Le résultat est affiché avec une fonction différente. Tester dans un programme qui s'arrête à la demande de l'utilisateur.

#### Exercice 4

Faire une fonction qui permet d'examiner la distribution des valeurs aléatoires obtenues avec la fonction rand() pour un nombre n de tirages entrés par l'utilisateur et sur 10 plages (des résultats de 0 à 9). Tester dans un programme qui s'arrête à la demande de l'utilisateur.

#### Exercice 7

Afficher au hasard un certain nombre d'étoiles (\*) dans un rectangle. Le nombre d'étoile et la taille du rectangle sont entrées par l'utilisateur sachant que la taille maximum est 25 par 80. Attention, il ne peut pas y avoir d'étoiles superposées.

#### Exercice 8

Dans un jeu de démineur le terrain est spécifié par une matrice de nombres. Faire deux fonctions, une pour l'initialisation avec un nombre de mines répandues aléatoirement dans le terrain et une autre pour indiquer combien il y a de mines autour de chaque position dans la zone de jeu. Faire un programme de test qui affiche les mines en rouge et pour chaque position le nombre des mines à proximité s'il y en a (pas d'affichage sinon).

### **M6.2 Exercices : mise en œuvre de structures**

#### Exercice 1

Une menuiserie industrielle gère un stock de panneaux de bois. Chaque panneau possède une largeur, une longueur et une épaisseur en millimètres ainsi que le type de bois. Il y a trois types de bois : pin (code 0), chêne (code 1), hêtre (code 2).

- 1) Définir une structure panneau contenant toutes les informations relatives à un panneau de bois.
- 2) Écrire les fonctions de saisie et d'affichage d'un panneau de bois.
- 3) Ecrire une fonction qui calcule et affiche le volume en mètre cube d'un panneau.

#### Exercice 2

Un grossiste de composants électronique vend quatre types de produits :

- des cartes mères (code 1)
- des processeurs (code 2)
- des barrettes mémoire (code 3)
- des cartes graphiques (code 4)

Chaque produit possède une référence, qui est un nombre entier, un prix en euros et une quantité disponible.

- 1) Définir une structure produit qui code un produit
  - 2) Écrire une fonction de saisie et d'affichage des données d'un produit
  - 3) Écrire une fonction qui permet à un utilisateur de saisir une commande d'un produit.
- L'utilisateur saisie les quantités demandées et les données du produit. L'ordinateur affiche toutes les données de la commande y compris le prix.

#### Exercice 3

Soit les deux modèles de structure "date" et "personne" déclarés ainsi :

```
#define LG_NOM 80
struct date{
    int jour, mois, annee;
};
struct personne{
    char nom[LG_NOM];
    struct date date_embauche;
```

```
struct date date_poste;
};
```

- Ecrire une fonction pour initialiser une structure de type "personne".
- Ecrire une fonction d'affichage de façon à obtenir soit l'un soit l'autre des deux affichage ci-dessous :

```
nom : DUPONT
date embauche (jj mm aa) : 16 01 08
date poste = date embauche ? (O/N) : O
```

```
nom : DUPONT
date embauche (jj mm aa) : 16 01 08
date poste = date embauche ? (O/N) : N
date poste (jj mm aa) : 01 09 08
```

### Exercice 5

Soit dans un jeu vidéo une entité en mode console. Elle est définie par une position, un déplacement, un type (rampant, grouillant, serpentant, plombant, assommant), une couleur et une lettre. L'entité a également un nom et une série de taux : taux d'agressivité, de colère, de convoitise, de faim, de peur.

Définir la structure de donnée pour coder une entité. Faire une fonction d'initialisation, une fonction de mise à 0 et une fonction d'affichage. Tester dans un programme avec un menu : quitter, afficher, initialiser, reset

### Exercice 6

Faire carnet de rendez-vous. Un rendez-vous sera défini par :

- Un libellé
- Un lieu de rendez-vous
- Une date (choisissez un format qui permettra le tri)
- Un horaire de début
- Un horaire de fin
- Une catégorie (bureau, personnel, loisirs etc.)
- Tout autre champ pouvant vous sembler utile

1) Mettre en forme la bonne structure de donnée

2) Dans un menu donnez les possibilités de :

- saisir un rendez-vous
- afficher un rendez-vous
- supprimer un rendez-vous
- quitter

## **M6.3 Exercices : mise en œuvre tableaux de structures**

### Exercice 1

Soit la structure suivante :

```
struct point{
    int x, y, color;
};
```

Dans un programme déclarer un tableau de NB\_POINT (par exemple 10).

- Faire une fonction d'initialisation. Cette fonction initialise chaque point soit au hasard soit

- par des valeurs entrées par l'utilisateur.
- Faire une fonction d'affichage
- Faire une fonction qui permet de modifier les valeurs d'un des points du tableau.
- Faire une fonction de mise à zéro de toutes les valeurs du tableau.
- Donner un menu utilisateur avec les commandes :  
quitter, initialiser, afficher, modifier, mise à 0

### Exercice 2

Faire un programme qui permet d'afficher le signe du zodiaque à partir d'un jour et d'un mois de naissance. Le jour est entré sous forme de nombre et le mois sous forme de chaîne de caractères. Voici les différents signes avec leur période :

Capricorne	23 décembre - 19 janvier
Verseau	20 janvier - 19 février
Poisson	20 février - 20 mars
Bélier	21 mars - 19 avril
Taureau	20 avril - 20 mai
Gémeau	21 mai - 20 juin
Cancer	21 juin - 21 juillet
Lion	22 juillet - 22 août
Vierge	23 août - 22 septembre
Balance	23 septembre - 22 octobre
Scorpion	23 octobre - 21 novembre
Sagittaire	22 novembre - 22 décembre

Exemple d'exécution :

```
donner votre jour et votre mois de naissance :
11 july
*** erreur de nom de mois****
donner votre jour et votre mois de naissance :
16 janvier
vous êtes né sous le signe : capricorne
```

### Exercice 2

Soit dans un jeu vidéo une entité en mode console. Elle est définie par une position, un déplacement, un type (rampant, grouillant, serpentant, plombant, assommant), une couleur et une lettre. L'entité a également un nom et une série de taux : taux d'agressivité, de colère, de convoitise, de faim, de peur.

Définir un type pour coder une entité.

Faire un programme dans lequel il y a NB\_ENTITE (par exemple 20) chacune a ses propres valeurs (entrées au hasard). Afficher la composition de votre ensemble d'entités pour le jeu.

### Exercice 3

Réaliser un programme établissant une facture pour une commande de plusieurs articles. Pour chaque article de la commande, l'utilisateur fournit la quantité et un numéro de code à partir duquel le programme retrouve à la fois le libellé et le prix unitaire. Le programme refuse les codes inexistantes. A la fin il affiche un récapitulatif tenant lieu de facture.

Les informations relatives aux différents articles sont définies dans le source du programme (et non dans un fichier de données). Elles sont placées en globale dans une librairie personnelle. Aidez-vous d'un catalogue de supermarché pour faire cette liste ; les codes sont à inventer.

1) Le programme propose un menu avec :



- nouvelle commande
- afficher la liste des articles disponibles

Prévoir une fonction de recherche des informations relatives à un article à partir de son numéro de code et une fonction d'affichage de la facture récapitulative.

Exemple d'exécution :

ARTICLE	NBRE	FACTURE P-UNIT	MONTANT
centrifugeuse	33	47.29	1560.57
Grille-pain	12	35.84	430.08
Four Raclette 6p	6	51.33	307.98
TOTAL			2298.63

2) Ajouter dans le programme une information sur le nombre d'articles disponibles. Au moment de la commande il y a vérification que le nombre d'articles demandés est disponible sinon l'utilisateur est invité à modifier sa commande. Le stock disponible est mis à jour après chaque commande.