

# Evaluation sur la programmation procédurale

Programmation procédurale en C#

## CONTENU

But .....	1
Matériel à disposition .....	1
Temps d'évaluation .....	1
Créations et initialisations de variables .....	2
Mois de naissance d'un utilisateur .....	2
Ma journée de samedi.....	2
Tableaux statiques .....	3
Fonctions et Tableaux statiques de nombres entiers.....	4
Le triangle de Pascal et fonction récursive .....	6
Bonus.....	7

## BUT

Cette évaluation a pour objet de permettre de jauger votre compréhension sur la programmation procédurale à ce jour. Il vous sera demandé d'être capable de :

- Décomposer un algorithme (saisie, traitement, affichage du résultat)
- Savoir déclarer et initialiser des variables
- Coder des structures alternatives
- Coder des structures répétitives
- Initialiser, parcourir et modifier un tableau statique
- Coder une procédure et une fonction statique
- Coder une structure récursive

Des codes propres, commentés et/ou documentés sont attendus. Votre code n'a pas besoin d'être optimisé au mieux, mais évitez de faire du code mort. Rappel : du code mort est un morceau de code inaccessible ou inutile.

## MATERIEL A DISPOSITION

Vous aurez droit d'utiliser tout ce que vous voulez sauf :

- Aucun accès internet ne sera toléré sur tout type de support (ordinateur, téléphone, ordinateur portable, ...)
- Aucune utilisation de téléphone, vous poserez vos téléphones sur vos bureaux en mode silencieux et sans y toucher, y compris durant les pauses.
- Pas d'ordinateur personnel
- Les notes et/ou programmes de vos collègues

En résumé : travaillez par vous-même.

## TEMPS D'EVALUATION

L'évaluation aura une durée de 8 heures. S'il s'avère que vous terminez avant le temps imparti, il serait bienvenu de vous occuper sans déranger vos collègues.

Étant donné que l'évaluation prendra une journée entière, soyez fairplays avec moi et vous-mêmes : pendant les pauses, ne discutez pas des exercices ou de vos idées pour les solutions avec vos collègues. Le but est d'évaluer votre niveau personnel, pas votre travail d'équipe.

## CREATIONS ET INITIALISATIONS DE VARIABLES

Créer un programme qui contient des constantes pour stocker votre nom, votre prénom et un âge. Afficher ces valeurs dans une phrase.

Demander l'âge de l'utilisateur. Afficher s'il est né avant ou après l'an 2000.

## MOIS DE NAISSANCE D'UN UTILISATEUR

Demander à l'utilisateur d'entrer le numéro de son mois de naissance (entre 1 et 12 inclus). Si le nombre indiqué est hors limite : demander une nouvelle saisie à l'utilisateur en lui précisant qu'il a commis une erreur.

Selon le mois entré par l'utilisateur, afficher un message contenant le nom du mois de naissance.

Exemple :

Entrez le numéro de votre mois de naissance :

-2

-2 ne correspond à aucun mois. Entrez un numéro (entre 1 et 12) correspondant à votre mois de naissance :

4

Avril est le quatrième mois de l'année !

## MA JOURNEE DE SAMEDI

Développer un programme qui décrit le texte suivant en veillant à demander à l'utilisateur les informations nécessaires pour pouvoir satisfaire toutes les conditions possibles du texte. Il est autorisé de ne pas afficher tout le texte mot pour mot et de l'adapter si besoin, mais il faudra veiller à respecter toutes les conditions du texte. La méthode de gestion des entrées de l'utilisateur est à la discrétion du développeur.

« Samedi matin, je suis en weekend. Il va falloir organiser ma journée.

Je me souviens qu'il faudra d'abord que j'aie faire mes courses. Je me prépare à sortir avec mes sacs de courses et ma liste de courses. Je vérifie que mes sacs sont bien vides. S'ils ne sont pas vides, je dois rapidement les vider. Ensuite je m'en vais. Par chance, le magasin n'est pas trop loin et le temps extérieur ne me dérangera pas trop sur le chemin.

Arrivé au magasin, je prends quelques articles puis je vérifie si j'ai tous mes articles. Si je ne les ai pas tous, je continue de chercher les articles manquants, sinon je passe aux caisses et je paie. Je rentre, je range mes achats.

J'ai pu voir le temps extérieur en allant et revenant des courses. Selon le temps, je vais m'occuper différemment :

S'il fait beau, je vais aller me promener le long de la rivière. Je pourrai profiter des installations sportives si je me sens en forme et si j'ai eu le temps de m'échauffer sur le chemin, sinon je vais simplement observer la faune encore présente en ce début de journée. Je serais de retour pour préparer mon repas ce midi.

S'il y a des nuages et de la pluie, je vais rester chez moi au chaud. Au programme : feu de cheminée, café ou thé chaud selon mon envie. Il va falloir continuer ma série préférée sur ma télé si ma connexion internet tient le coup. Sinon, ce sera lecture du dernier livre de mon auteur favoris.

Si au contraire il a neigé la nuit passée et qu'il reste encore de la neige au sol et qu'il neige un peu, alors je vais essayer d'en profiter. Je prends mes gants, mon bonnet, mon écharpe. Si ma voiture n'était pas dans le garage cette nuit, je

*prends ma pelle à neige en plus et je dégage ma voiture. Après avoir rangé ma pelle si je l'ai utilisée, je pars en montagne faire du ski pour la journée. J'aviserais sur place pour louer l'équipement qui me manque.*

*Si je suis chez moi à midi, il va falloir que je me fasse à manger. Si je suis motivé : petits pois, purée de carottes et un steak. Sinon ce sera juste un sandwich fait rapidement avec salade, tomate, cornichon, œufs durs, jambon et beurre. Si je suis parti faire du ski, je vais devoir me trouver un restaurant sur les pistes et espérer trouver une place. Tant qu'il n'y a pas de place, je refais la même piste pour voir si une place se libère au restaurant trouvé. Après le repas je retournerai sur les pistes jusqu'à leur fermeture en vérifiant l'heure après chaque descente. Puis je rentrerai chez moi.*

*Pour l'après-midi dans le cas où je ne suis pas parti faire du ski, si des amis sont disponibles je vais sortir avec eux cet après-midi, s'ils ne peuvent pas sortir ce sera plutôt un jeu multijoueur en ligne. Enfin, si aucun ami n'est disponible, je vais me contenter de m'occuper de mes plantes, je me souviens que j'avais un souci de cochenilles. Il va falloir s'en débarrasser ! Pour le repas du soir, si je suis sorti avec mes amis, je vais sûrement au restaurant avec eux. Si je me suis occupé de mes plantes, j'ai eu tout le temps de préparer un festin, je vais même pouvoir proposer à ma famille de profiter du repas ensemble. Si j'ai joué en ligne avec mes amis, je n'ai pas vu le temps passer, il va falloir manger sur le pouce un repas rapide à préparer. En rentrant du ski, je suis condamné à manger rapidement, je suis trop fatigué pour me faire un bon repas. »*

## TABLEAUX STATIQUES

Faites un tableau dans lequel vous allez stocker 6 questions à poser à l'utilisateur. Ces questions devront être des questions auxquelles les seules réponses possibles sont " vrai " ou " faux ". Vous poserez ensuite ces questions une à une à l'utilisateur en le laissant répondre et en stockant ces réponses dans un autre tableau. Dans le cas où l'utilisateur ne répond pas par vrai ou faux, reposez-lui la même question jusqu'à ce que la réponse soit valide. Enfin vous devrez afficher les réponses à chaque question en réaffichant chaque question avec sa réponse.

Répondez aux questions suivantes par vrai ou faux :

Avez-vous votre permis voiture ?

Vrai

Aimez-vous faire du ping-pong ?

Vrai

Avez-vous des enfants ?

Non

Avez-vous des enfants ?

Faux

[. . .]

Avez-vous votre permis voiture ? Vous avez répondu "Vrai"

Aimez-vous faire du ping-pong ? Vous avez répondu "Vrai"

Avez-vous des enfants ? Vous avez répondu "Faux"

[...]

## FONCTIONS ET TABLEAUX STATIQUES DE NOMBRES ENTIERS

- 1) Faire une fonction qui génère un tableau statique d'entiers dont la taille est aléatoire entre 1 et 20 compris. Chaque entier du tableau doit également être aléatoire et compris entre -50 et 50.

Rappel :

```
//générer et stocker un nombre entier aléatoire entre 0 et 19 inclus
Random rnd = new Random() ;
int monEntierAleatoire = rnd.Next(0 ; 20) ;
```

- 2) Faire une fonction qui génère un tableau statique d'entiers à partir de la saisie de l'utilisateur. L'utilisateur devra préciser la taille du tableau (entre 1 et 20 compris). L'utilisateur devra ensuite saisir dans un deuxième temps les nombres les uns à la suite des autres et séparés par un underscore '\_' .  
Si l'utilisateur entre plus de 20 nombres entiers : ne tenez pas compte des nombres en trop. Si l'utilisateur a entré des valeurs non valides (des lettres, des mots, de la ponctuation autre que le '\_', des nombres à virgule) : demander une nouvelle saisie.

- 3) Faire une procédure qui affiche les éléments d'un tableau et la taille du sur une ligne.

- 4) Faire une procédure qui calcule et affiche les éléments suivants :

- la moyenne du tableau. Rappel :  $\text{moyenne de } n \text{ nombres} = \text{somme des nombres} / n$
- le minimum du tableau : le nombre le plus petit du tableau
- le maximum du tableau : le nombre le plus grand
- l'écart entre le minimum et le maximum du tableau

- 5) Demandez à l'utilisateur s'il veut entrer lui-même jusqu'à 20 nombres entier ou s'il veut que ces nombres soient remplis aléatoirement.

Selon la réponse de l'utilisateur appeler la fonction de génération de tableau d'entiers à partir de la saisie utilisateur ou bien appeler la fonction de génération de tableau d'entiers aléatoire.

Dans tous les cas vous afficherez ensuite les éléments du tableau et sa taille ainsi que la moyenne du tableau, son minimum, son maximum et l'écart entre le minimum et le maximum à l'aide des procédures créées.

Exemple :

Souhaitez-vous entrer jusqu'à 20 entiers séparés par le caractère '\_' ? ('OUI' pour valider, sinon le tableau sera aléatoire) :

OUI

Combien de nombres souhaitez-vous entrer ?

10

Entrez 10 entiers positif ou négatifs séparés par '\_'

12\_99\_68\_42\_1000\_27\_1010\_45\_13\_27

Vous avez saisi :

12 99 68 42 1000 27 1010 45 13 27, ce qui fait un total de 10 valeurs

La moyenne de votre tableau est : 234,3

Le minimum est : 12

Le maximum est : 1010

L'écart entre le minimum et le maximum est : 998

## LE TRIANGLE DE PASCAL ET FONCTION RECURSIVE

"Le triangle de Pascal est une présentation des coefficients binomiaux dans un triangle." – Wikipédia

Maintenant que vous avez une description de ce qu'est le triangle de Pascal, voici à quoi il ressemble :

n = 0	1									
n = 1	1	1								
n = 2	1	2	1							
n = 3	1	3	3	1						
n = 4	1	4	6	4	1					
n = 5	1	5	10	10	5	1				
n = 6	1	6	15	20	15	6	1			
n = 7	1	7	21	35	35	21	21	1		
n = 8	1	8	28	56	70	56	42	22	1	
n = 9	1	9	36	84	126	126	98	64	23	1

La première ligne est composée d'un seul 1. A chaque nouvelle ligne, on ajoute une valeur en plus que sur la ligne précédente.

Chaque ligne commence par un 1 et se termine également par un 1. Les autres valeurs sont calculées à partir de la ligne précédente. On additionne la valeur de la ligne précédente dans la même colonne et la valeur de la ligne précédente dans la colonne précédente. Ainsi ici, la case rose est 35 car on additionne les valeurs des deux cases bleues de la ligne précédente  $20 + 15 = 35$ .

- 1) A partir de ces informations, faire une fonction qui permet d'afficher les 10 premières lignes du tableau de Pascal. (ce qui devrait afficher les lignes du tableau ci-dessus)
- 2) Faire une fonction récursive qui renvoie la Nième ligne du tableau de Pascal.  
Faire une fonction pour demander à l'utilisateur quelle ligne afficher et l'afficher

Exemple :

Entrer le numéro de ligne du triangle de pascal à afficher :

6

1 6 15 20 15 6 1

## BONUS

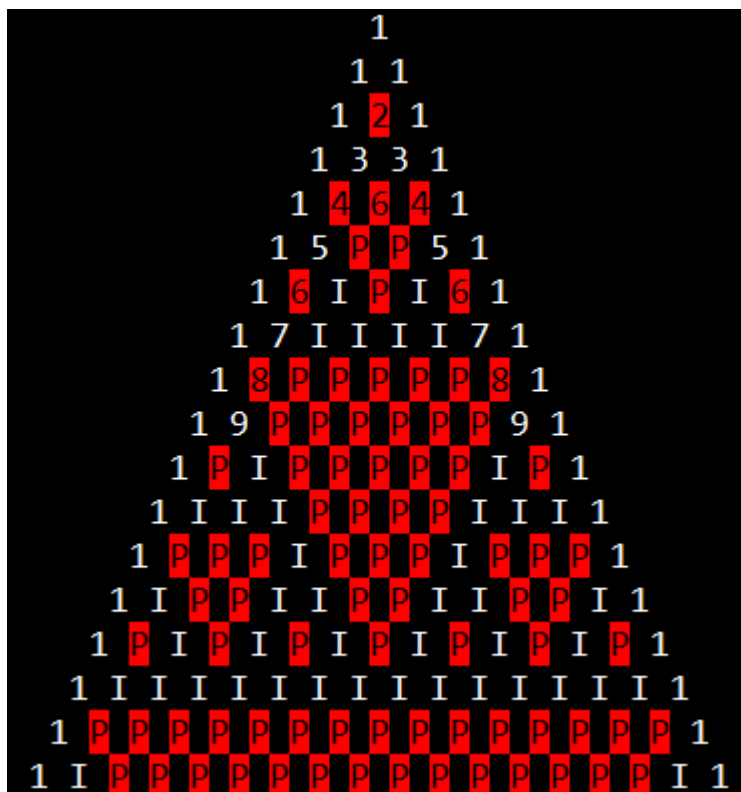
Pour occuper un peu ceux qui ont terminé :

1. Reprenez chaque exercice et assurez-vous qu'ils fonctionnent bien même si l'utilisateur entre de mauvaises données : s'il entre un mot à la place d'un nombre, des caractères qui ne devraient pas être là, etc ...
2. Reprenez le triangle de Pascal. En regardant l'image suivante, vous verrez qu'il est possible de calculer la suite de Fibonacci depuis le triangle de Pascal en additionnant les sommes des diagonales.

n = 0	1									
n = 1	1	1								
n = 2	1	2	1							
n = 3	1	3	3	1						
n = 4	1	4	6	4	1					
n = 5	1	5	10	10	5	1				
n = 6	1	6	15	20	15	6	1			
n = 7	1	7	21	35	35	21	7	1		
n = 8	1	8	28	56	70	56	28	8	1	
n = 9	1	9	36	84	126	126	84	36	9	1

Vous vous en doutez, je vais vous demander de faire cela : Calculer les nombres de la suite de Fibonacci à partir du triangle de Pascal

3. Reprenez l'affichage des 10 premières lignes du triangle de Pascal. Votre objectif sera d'afficher le triangle de Pascal de la manière suivante :

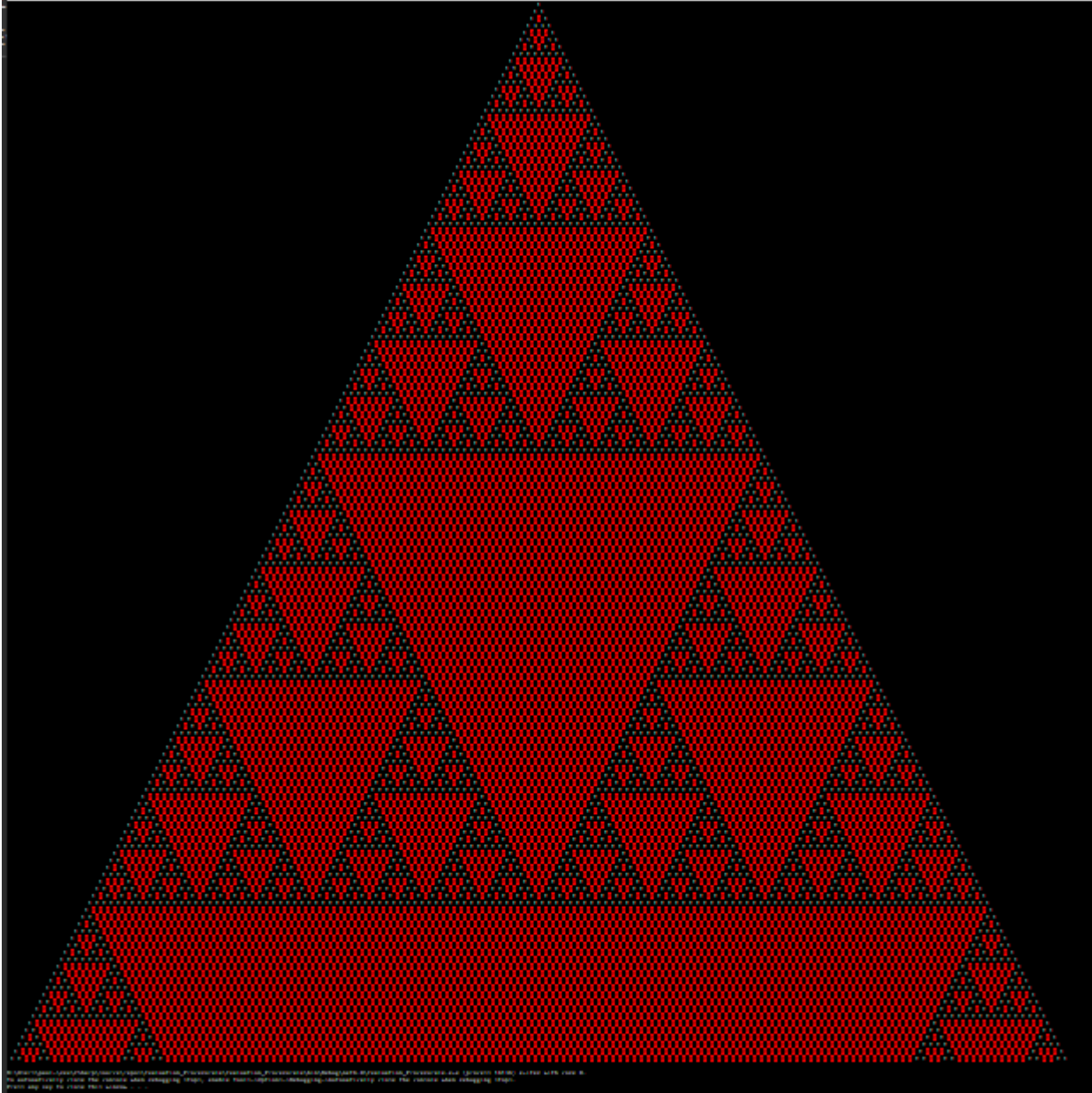




Afficher les entiers pairs de manière différentes des nombres impaires dans un triangle isocèle. Pour changer les couleurs de la console vous pouvez utiliser `Console.BackgroundColor = ConsoleColor.White;` et `Console.ForegroundColor = Console.ForegroundColor = ConsoleColor.White;`

Les nombres au-delà de 9 peuvent être représenté par un symbole. J'ai utilisé 'P' pour pair et 'I' pour impair.

Voici le résultat avec 150 lignes :



--- FIN DU DOCUMENT ---