

LIFAP1 – CC mi-parcours – Séquence 1

Contrôle Continu (Durée totale : 1h)

Mardi 2 novembre 2021

*Recommandations : Les documents, calculatrice, téléphone portable sont interdits. La qualité de l'écriture et de la présentation seront prises en compte dans la note finale. Vous veillerez à **respecter** les notations et les règles d'écriture des algorithmes vues en cours et en TD. Un soin tout particulier devra être apporté à l'écriture des entêtes des différents sous-programmes.*

NOM :

.....

PRENOM :

.....

Numéro Etudiant :

.....

Partie A – Algorithmique

La suite de Padovan est une suite d'entiers définie par récurrence par $\mathcal{P}_{n+3} = \mathcal{P}_{n+1} + \mathcal{P}_n$ pour tout entier n positif.

Les trois valeurs initiales sont : $\mathcal{P}(0) = \mathcal{P}(1) = \mathcal{P}(2) = 1$

1. Écrire l'algorithme d'une **procédure** `saisie_valeur` qui "retourne" une valeur choisie par l'utilisateur strictement supérieure à 2 et inférieure à une constante MAX définie au préalable. La saisie devra être recommencée tant que la valeur ne respecte pas les contraintes.

2. Écrire l'algorithme d'une **fonction** `terme_padovan` qui calcule le $n^{\text{ième}}$ terme de la suite de Padovan, n étant passé en paramètre de la fonction, **sans passer par la construction d'un tableau**.

3. **Sans utiliser la fonction précédente**, écrire l'algorithme d'une **procédure** `tableau_padovan` qui remplit un tableau avec les n premiers termes de la suite de Padovan n étant passé en paramètre. On pourra utiliser la constante `MAX` pour la définition du tableau.

4. En utilisant les sous-programmes précédents, écrire l'algorithme du programme principal permettant
 - a. de saisir une valeur n comprise entre 3 et MAX,
 - b. de calculer le $n^{\text{ième}}$ terme de la suite en utilisant respectivement les sous-programmes écrits en 2 et 3,
 - c. et d'afficher un message confirmant, après les avoir calculées, que ces deux valeurs sont bien identiques.

Partie B – Langage C/C++

On dispose d'un tableau recensant les statistiques de performance d'un coureur sur une distance de 10 km. Dans le tableau ci-après, la première ligne représente les kilomètres, la seconde ligne la vitesse moyenne sur ce kilomètre, et la troisième le nombre de calories brûlées sur ce même kilomètre.

Km	1	2	3	4	5	6	7	8	9	10
Vitesse	11.5	12.3	12.3	11.9	13.0	11.7	12.6	12.8	13.5	11.2
Calories	49	55	58	57	64	67	77	80	92	75

1. Définir en C/C++ un tableau pouvant contenir ces informations. On pourra ne mémoriser que les informations des 2 dernières lignes du tableau précédent : vitesse et calories.

2. Écrire en langage C/C++ un **sous-programme** `extraire_stats` qui en un seul parcours de ce tableau calcule et "retourne" la vitesse moyenne sur la distance totale (`vmoy`), le kilomètre le plus rapide (`fast`), le kilomètre le plus lent (`slow`) et le nombre total de calories brûlées (`nrj`) pendant la séance.

3. On dispose d'une procédure `affiche_tab` qui affiche le contenu d'un tableau 2D de 10 réels et d'une procédure `remplit_tab` qui remplit le contenu d'un tableau 2D de 10 réels. Écrire en C/C++ le programme principal qui remplit le tableau de performances, l'affiche puis affiche les statistiques obtenues en 2.