

LIFAP1 – CC mi-parcours – Séquence 5

Contrôle Continu (Durée totale : 1h)

Vendredi 5 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

Évolution d'une somme placée en banque.

On place à la banque un capital de départ de 500 €. Chaque année le capital augmente avec un taux d'intérêt de 2 % et chaque année on rajoute 100 €.

La valeur du capital à l'année n est obtenu par la suite u_n définie par :

$$u_0 = 500$$

$$u_{n+1} = 1,02 \times u_n + 100$$

1. Écrire l'algorithme d'un **sous-programme** `tab_suite` qui remplit un tableau de taille `MAX` (`MAX` constante définie au préalable à la valeur 100) avec les `MAX` premiers termes de la suite.

2. Écrire l'algorithme d'un **sous-programme** `capital_superieur` qui, à partir du tableau précédent retourne le nombre d'années nécessaires pour dépasser une certaine valeur de `capital` passé en paramètre. Si le montant souhaité reste supérieur à la valeur de la dernière case du tableau, la fonction renverra `-1`.

3. Écrire l'algorithme du **programme principal** qui demande à l'utilisateur le capital à atteindre et lui affiche le nombre d'années nécessaires pour obtenir ce capital. On passera par la construction du tableau contenant toutes les valeurs de la suite.

Partie B – Langage C/C++

Les carrés bordés

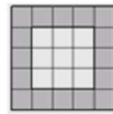
On veut déterminer le nombre de carrés blancs et le nombre de carrés gris dans des configurations de carré différentes, tels que ceux-ci-dessous.



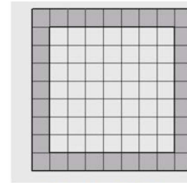
$n = 3$



$n = 4$



$n = 5$



$n = 9$

Pour un carré de taille n , il y a $4(n-1)$ carrés noirs et $(n - 2)^2$ carrés blancs ; la somme des deux doit être égale à n^2 .

- 1- Écrire en langage C/C++ un sous-programme `compte_blancs_noirs` qui calcule et "retourne" le nombre de carrés blancs et le nombre de carrés noirs en fonction de la taille du carré, n , passée en paramètre.

- 2- Définir en C/C++ une constante `MAX` ayant comme valeur 13.

- 3- Écrire en langage C/C++ un sous-programme qui remplit un tableau 2D de taille 2 lignes et `MAX` colonnes contenant dans la première ligne le nombre de carrés noirs et dans la seconde le nombre de carrés blancs.

- 4- Écrire en langage C/C++ une **fonction booléenne** `noirs_plus_blancs` qui retourne vrai si pour une valeur `n` passée en paramètre la somme du nombre de carré noirs et de carrés blancs est bien égale à n^2 , faux sinon. On utilisera impérativement le tableau construit à la question précédente.

- 5- Écrire en langage C/C++ une **fonction** `saisie_taille_carre` qui demande à l'utilisateur de choisir une valeur comprise entre 3 et 15 (inclus) et la retourne au programme principal. La saisie sera recommencée tant que la valeur ne satisfait pas les contraintes.

- 6- Écrire en langage C/C++ le programme principal qui permet de construire le tableau contenant pour toutes les valeurs de `n` comprises entre 3 et 15 le nombre de carrés noirs et le nombre de carrés blancs, puis de saisir une valeur `n` comprise entre 3 et 15 et de vérifier que la somme des carrés blancs et des carrés noirs est bien égale à n^2 .