

LIFAP1 – CC1 – Séquence 3 – Sujet A

Durée totale : 1h – Mardi 30 septembre 2025

***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.*

Partie A – Algorithmique

Un nombre est **abondant** si la somme des diviseurs stricts est strictement supérieure à lui-même. Exemple 12 est abondant car la somme de ses diviseurs stricts est 16 ($=1+2+3+4+6$)

- 1- Ecrire l'algorithme d'un **sous-programme** `est_abondant` qui retourne vrai si `n` passé en paramètre est abondant, faux sinon.

NOM

.....

PRENOM

.....

Numéro Etudiant

.....

Groupe TD

.....

- 2- Ecrire l'algorithme du **programme principal** qui, en utilisant le sous-programme précédent, affiche tous les nombres abondants inférieurs à 100.

Partie B – Langage C/C++

On souhaite écrire un programme qui nous permette d'afficher le motif ci-contre : un triangle de chiffres croissants. L'utilisateur choisira la hauteur du triangle (un entier strictement positif).

```
1
2 3
4 5 6
7 8 9 10
11 12 13 14 15
```

Dans l'exemple, hauteur = 5.

- 1- Ecrire en langage C/C++ un **sous-programme** `triangle` qui à partir de la hauteur du triangle passée en paramètre affiche le motif voulu.

- 2- Ecrire le **programme principal** qui demande à l'utilisateur un entier strictement positif. On recommencera la saisie tant que la valeur n'est pas strictement positive. Utiliser ensuite le sous-programme précédent pour afficher le motif correspondant.

- 3- Un nombre est dit "spécial" s'il est **pair et multiple de 3** ou **s'il est strictement inférieur à 10**. Ecrire en langage C/C++ un **programme principal** qui affiche tous les nombres spéciaux inférieurs à 20.