



Objectifs

- **Réaliser** l'itération 2 de l'application Gestion Centralisée de logs
 - **Améliorer** l'interaction utilisateur
- **Maîtriser** les outils DevOps

Contenu technique

- Programmer une alternative complexe
- Programmer une boucle

Durée 3h itération, 2h exercices (à faire à la maison)

I. Améliorer l'interaction utilisateur

Votre première itération est fonctionnelle. Nous allons améliorer celle-ci en ajoutant des fonctionnalités au programme.

I.1 Alternative complexe.

Pour traiter l'ensemble des choix de menu, l'alternative `if...else` devient très lourde.

Tâche 1. Utiliser l'instruction **switch case** pour améliorer la lisibilité de votre code.

Résultat attendu :

- ✓ Les différents choix du menu sont sélectionnés en utilisant un switch case

I.2 Répétition

Le langage C++ offre 3 possibilités de répétitions d'un traitement (bloc de programme).

Votre programme doit pouvoir être répété autant que l'utilisateur le souhaite.

Q1. En consultant le cahier des charges, déterminez et donner l'exigence (numéro et description) correspondante à la répétition du programme.

Q2. Quelle est la touche du clavier utilisée pour quitter le programme ?

Tâche 2. Utiliser la boucle de votre choix pour traiter le cas ci-dessus.

Résultat attendu :

- ✓ Le menu se répète tant que l'utilisateur ne choisit pas de quitter le programme



II. Outils DevOps

II.1 Formatage du code

Tâche 3. Suivre la page [Standard de codage](#) pour formater correctement votre code

Résultats attendus :

- ✓ Le code doit être correctement indenté
- ✓ Les noms de variables doivent respecter la convention de nommage

II.2 Documentation du code

Tâche 4. Suivre la page [Documenter son code](#) pour documenter correctement votre code

Résultats attendus :

- ✓ Le fichier `main.cpp` doit être correctement documenté
- ✓ La fonction `main()` doit être correctement documentée
- ✓ Les points de fonctionnement délicats du programme doivent être explicités

II.3 Sauvegarde de votre travail

Tâche 5. Suivre la page [Git premiers pas](#) pour sauvegarder votre travail sur le serveur GIT de la section.

Résultats attendus :

- ✓ Un dépôt **<VOTRE_NOM>_Gestion_Log** doit être configuré sur le serveur de la section
- ✓ Un tag (v2.0 avec un message indiquant la fin de l'itération) doit être créé sur le serveur.
- ✓ Votre travail (**liste fichiers ci-dessous**) doit être sauvegardé sur ce dépôt
 - `CMakeLists.txt`
 - `Doxyfile`
 - `main.cpp`
 - `readme.md`
 - `.gitignore`
 - `.gitlab-ci.yml`

II.4 Pipeline d'intégration

Le projet modèle que vous utilisez est livré avec un pipeline d'intégration DevOps. Il contient 2 étapes :

- Compilation de votre projet
- Tests SAST



Tâche 6. Vérifier sur Gitlab que le pipeline est correctement exécuté.

Q3. Intégrer un screenshot du résultat dans votre compte rendu.

III. Entraînement

Des exercices vous sont proposés pour maîtriser les notions abordées dans cette itération. Ils sont à faire à la maison, sauf si vous êtes en avance sur le planning de l'itération.

Q4. Faire au moins 3 exercices (au choix selon vos besoins)

Le texte des exercices sont sur le site du [BTS CIEL](#), code à remettre avec votre compte rendu.

III.1 Affectation de variables

Exercice 2.a: Permutation de deux variables.

III.2 Afficher un texte, entrer une variable

Exercice 3.b : Déclarer, afficher

Exercice 3.d : Par ici la monnaie

III.3 Conditions et alternatives

Exercices 4.a : Positif ?

Exercice 4.c : Voyelle ou consonne ?

III.4 Les itérations

Exercice 5.a : while do for

Exercice 5.d : Dessin

IV. Livrable

Sur le moodle de la section

→ Votre compte rendu de TP avec les réponses aux questions de celui-ci

Sur le serveur GIT de la section

→ Le projet **<VOTRE_NOM>_Gestion_Log** avec le tag correspondant à la fin de l'itération2.