

## **Iteration 3 - Fonctions et Modules**

### **Etape 1 : Sous-fonction afficherMenu()**

Code en C++ :

```

/**
 * @file main.cpp
 * @brief Programme de gestion de logs - Iteration 3
 * @author HAUTOT Nolan - LEFRANC Robin
 * @date 16/09/2025
 * @version 3.0
 */

/**
 * @brief Fonction principale du programme
 * @details Affiche un menu en boucle jusqu'à ce que
 * l'utilisateur choisisse de quitter (choix=0)
 * @return 0 si le programme se termine normalement
 */

#include <iostream>
#include <string>

using namespace std;

void afficherMenu(); // Prototype de la fonction afficherMenu()

int main()
{
    // Variable pour stocker le choix
    int choix = 0;

    do // On effectue une boucle do-while
    {
        // Affichage du Menu "Gestion de logs"
        afficherMenu(); // On appelle la fonction

        // Demander le choix de l'utilisateur
        cin >> choix;

        switch (choix) // Ici on utilise switch à la place de if/else if pour une meilleure lis:
        {
            case 1: // Les 'case' sont l'équivalent des 'if'.
                cout << "Affichage des evenements systeme...";
                break; // Le 'break' est obligatoire afin d'éviter une execution continue.
            case 2:
                cout << "Affichage des connexions reseau...";
                break;

```

```

    case 3:
        cout << "Affichage des actions utilisateur...";
        break;
    case 4:
        cout << "Affichage des connexions utilisateur...";
        break;
    case 0:
        cout << "Au revoir !";
        break;
    default: // Le 'default' est l'equivalent du 'else' final.
        cout << "Erreur ! Le choix " << choix << " n'est pas un choix valide !";
}

if (choix != 0)
{
    cout << "\nAppuyer sur Entree...";
    cin.ignore(); // Vide le buffer
    cin.get();    // Attend Entrée
}

} while (choix != 0); // Ici tant que choix est différent de 0, on continue la boucle.

return 0;
}

void afficherMenu() // On créer la fonction et on colle le code pour afficher le menu.
{
    cout << "" << endl;
    cout << "===== " << endl;
    cout << "HAUTOT Nolan - LEFRANC Robin | CIEL - Gestion de logs" << endl;
    cout << "===== " << endl;
    cout << "" << endl;
    cout << "Menu principal :" << endl;
    cout << "" << endl;
    cout << "1 - Voir les evenements systeme" << endl;
    cout << "2 - Voir les connexions reseau" << endl;
    cout << "3 - Voir les actions utilisateur" << endl;
    cout << "4 - Voir les connexions utilisateur" << endl;
    cout << "0 - Quitter" << endl;
    cout << "" << endl;
    cout << "Votre choix : ";
}

```

*Question 1* : Pourquoi utiliser `void` comme type de retour ?

- On utilise `void` comme type de retour pour ne retourner aucune valeur.

*Question 2* : Où placer le prototype dans le fichier ?

- On place le prototype avant `main()` .

*Question 3* : Comment tester que la fonction fonctionne bien ?

- Pour tester si la fonction fonctionne bien, il suffit de compiler le code avec la fonction et de le lancer. Si tout se passe bien alors la fonction marche. Dans le cas contraire il faut résoudre les problèmes.

## **Etape 2 : Sous-fonction choisirLog()**

Code en C++ :

```

/**
 * @file main.cpp
 * @brief Programme de gestion de logs - Iteration 3
 * @author HAUTOT Nolan - LEFRANC Robin
 * @date 16/09/2025
 * @version 3.0
 */

/**
 * @brief Fonction principale du programme
 * @details Affiche un menu en boucle jusqu'à ce que
 * l'utilisateur choisisse de quitter (choix=0)
 * @return 0 si le programme se termine normalement
 */

#include <iostream>
#include <string>

using namespace std;

void afficherMenu(); // Prototype de la fonction afficherMenu()
void choisirLog(int choix); // Prototype de la fonction choisirLog() avec comme paramètre int cl

int main()
{
    // Variable pour stocker le choix
    int choix = 0;

    do // On effectue une boucle do-while
    {
        // Affichage du Menu "Gestion de logs"
        afficherMenu(); // On appelle la fonction afficherMenu()

        // Demander le choix de l'utilisateur
        cin >> choix;

        choisirLog(choix); // On appelle la fonction choisirLog(choix)

        if (choix != 0)
        {
            cout << "\nAppuyer sur Entree...";
            cin.ignore(); // Vide le buffer
            cin.get();    // Attend Entrée
        }
    }
}

```

```
}
```

```
} while (choix != 0); // Ici tant que choix est différent de 0, on continue la boucle.
```

```
return 0;
```

```
}
```

```
void afficherMenu() // On créer la fonction et on colle le code pour afficher le menu.
```

```
{
```

```
    cout << "" << endl;
```

```
    cout << "===== " << endl;
```

```
    cout << "HAUTOT Nolan - LEFRANC Robin | CIEL - Gestion de logs" << endl;
```

```
    cout << "===== " << endl;
```

```
    cout << "" << endl;
```

```
    cout << "Menu principal :" << endl;
```

```
    cout << "" << endl;
```

```
    cout << "1 - Voir les evenements systeme" << endl;
```

```
    cout << "2 - Voir les connexions reseau" << endl;
```

```
    cout << "3 - Voir les actions utilisateur" << endl;
```

```
    cout << "4 - Voir les connexions utilisateur" << endl;
```

```
    cout << "0 - Quitter" << endl;
```

```
    cout << "" << endl;
```

```
    cout << "Votre choix : ";
```

```
}
```

```
void choisirLog(int choix) // On créer la fonction et on colle le code pour gérer le choix de 1
```

```
{
```

```
    switch (choix) // Ici on utilise switch à la place de if/else if pour une meilleure lisibil:
```

```
    {
```

```
        case 1: // Les 'case' sont l'equivalent des 'if'.
```

```
            cout << "Affichage des evenements systeme...";
```

```
            break; // Le 'break' est obligatoire afin d'eviter une execution continue.
```

```
        case 2:
```

```
            cout << "Affichage des connexions reseau...";
```

```
            break;
```

```
        case 3:
```

```
            cout << "Affichage des actions utilisateur...";
```

```
            break;
```

```
        case 4:
```

```
            cout << "Affichage des connexions utilisateur...";
```

```
            break;
```

```
        case 0:
```

```
            cout << "Au revoir !";
```

```

        break;
    default: // Le 'default' est l'équivalent du 'else' final.
        cout << "Erreur ! Le choix " << choix << " n'est pas un choix valide !";
    }
}

```

*Question 1* : Pourquoi passer 'choix' en paramètre plutôt qu'une variable globale ?

- Car cela nous permet de modifier le paramètre choix à l'intérieur de la fonction. On reçoit le choix de l'utilisateur et on traite l'information.

*Question 2* : Que se passe-t-il si on modifie 'choix' dans la fonction ?

- Il ne se passe rien pour la variable originale `choix` car la variable mise dans la fonction est une copie de celle-ci.

*Question 3* : Où doit rester la gestion de la pause (`cin.ignore/cin.get`) ?

- La gestion de la pause doit rester dans `main()`.

## Etape 3 : Création du module Menu

Code du `menu.h` :

```

// On inclut les include guards
#ifndef MENU_H
#define MENU_H

// On ajoute les bibliothèques nécessaires
#include <iostream>

// On déclare les prototypes
void afficherMenu();
void choisirLog(int choix);

#endif // MENU_H

// ifndef, define et endif empêchent l'inclusion multiple du même fichier .h. Cela pourrait provoquer des erreurs de compilation.

```

Code du `menu.cpp` :

```

#include "menu.h"
#include <cstdlib>

using namespace std;

void afficherMenu() // On créer la fonction et on colle le code pour afficher le menu.
{
    system("cls");
    cout << "" << endl;
    cout << "===== " << endl;
    cout << "HAUTOT Nolan - LEFRANC Robin | CIEL - Gestion de logs" << endl;
    cout << "===== " << endl;
    cout << "" << endl;
    cout << "Menu principal :" << endl;
    cout << "" << endl;
    cout << "1 - Voir les evenements systeme" << endl;
    cout << "2 - Voir les connexions reseau" << endl;
    cout << "3 - Voir les actions utilisateur" << endl;
    cout << "4 - Voir les connexions utilisateur" << endl;
    cout << "0 - Quitter" << endl;
    cout << "" << endl;
    cout << "Votre choix : ";
}

void choisirLog(int choix) // On créer la fonction et on colle le code pour gérer le choix de 1
{
    switch (choix) // Ici on utilise switch à la place de if/else if pour une meilleure lisibil:
    {
        case 1: // Les 'case' sont l'equivalent des 'if'.
            cout << "Affichage des evenements systeme...";
            break; // Le 'break' est obligatoire afin d'eviter une execution continue.
        case 2:
            cout << "Affichage des connexions reseau...";
            break;
        case 3:
            cout << "Affichage des actions utilisateur...";
            break;
        case 4:
            cout << "Affichage des connexions utilisateur...";
            break;
        case 0:
            cout << "Au revoir !";
            break;
    }
}

```



```
default: // Le 'default' est l'equivalent du 'else' final.  
    cout << "Erreur ! Le choix " << choix << " n'est pas un choix valide !";  
}  
}
```

Code du main.cpp :

```

/**
 * @file main.cpp
 * @brief Programme de gestion de logs - Iteration 3
 * @author HAUTOT Nolan - LEFRANC Robin
 * @date 16/09/2025
 * @version 3.0
 */

/**
 * @brief Fonction principale du programme
 * @details Affiche un menu en boucle jusqu'à ce que
 * l'utilisateur choisisse de quitter (choix=0)
 * @return 0 si le programme se termine normalement
 */

#include <iostream>
#include <string>

// On inclut le module Menu qu'on a créer
#include "Menu/menu.h"

using namespace std;

int main()
{
    // Variable pour stocker le choix
    int choix = 0;

    do // On effectue une boucle do-while
    {
        // Affichage du Menu "Gestion de logs"
        afficherMenu(); // On appelle la fonction afficherMenu()

        // Demander le choix de l'utilisateur
        cin >> choix;

        choisirLog(choix); // On appelle la fonction choisirLog(choix)

        if (choix != 0)
        {
            cout << "\nAppuyer sur Entree...";
            cin.ignore(); // Vide le buffer
            cin.get();    // Attend Entrée
        }
    } while (choix != 0);
}

```

```
    }  
  
    } while (choix != 0); // Ici tant que choix est différent de 0, on continue la boucle.  
  
    return 0;  
}
```

On divise le `main.cpp` en 3 fichiers : `menu.h` , `menu.cpp` et `main.cpp` .

Dans le fichier `menu.h` on déclare tous les prototypes dont on aura besoin.

Dans le fichier `menu.cpp` on crée toutes les fonctions dont on aura besoin.

Enfin dans le fichier `main.cpp` on met le reste du code.

## Etape 4 : Documentation des fonctions

Code de `menu.h` après documentation :

```

/**
 * @file menu.h
 * @brief Module de gestion du menu principal
 * @author HAUTOT Nolan - LEFRANC Robin
 * @date 16/09/2025
 */

// On inclut les include guards
#ifndef MENU_H
#define MENU_H

// On ajoute les bibliothèques nécessaires
#include <iostream>

// On déclare les prototypes
/**
 * @brief Affiche le menu principal du programme
 * @details Affiche le titre et toutes les options disponibles
 */
void afficherMenu();

/**
 * @brief Traite le choix de l'utilisateur
 * @param choix Numéro de l'option sélectionnée (0-3)
 * @details Utilise un switch/case pour afficher les logs correspondants
 */
void choisirLog(int choix);

#endif // MENU_H

// ifndef, define et endif empêchent l'inclusion multiple du même fichier .h. Cela pourrait provoquer des erreurs de compilation.

```

Code de menu.cpp après documentation :

```

/**
 * @file menu.cpp
 * @brief Implémentation du module de gestion du menu
 * @author HAUTOT Nolan - LEFRANC Robin
 * @date 16/09/2025
 */

#include "menu.h"
#include <cstdlib>

using namespace std;

void afficherMenu() // On créer la fonction et on colle le code pour afficher le menu.
{
    system("cls");
    cout << "" << endl;
    cout << "===== " << endl;
    cout << "HAUTOT Nolan - LEFRANC Robin | CIEL - Gestion de logs" << endl;
    cout << "===== " << endl;
    cout << "" << endl;
    cout << "Menu principal :" << endl;
    cout << "" << endl;
    cout << "1 - Voir les evenements systeme" << endl;
    cout << "2 - Voir les connexions reseau" << endl;
    cout << "3 - Voir les actions utilisateur" << endl;
    cout << "4 - Voir les connexions utilisateur" << endl;
    cout << "0 - Quitter" << endl;
    cout << "" << endl;
    cout << "Votre choix : ";
}

void choisirLog(int choix) // On créer la fonction et on colle le code pour gérer le choix de 1
{
    switch (choix) // Ici on utilise switch à la place de if/else if pour une meilleure lisibil:
    {
        case 1: // Les 'case' sont l'équivalent des 'if'.
            cout << "Affichage des evenements systeme...";
            break; // Le 'break' est obligatoire afin d'éviter une execution continue.
        case 2:
            cout << "Affichage des connexions reseau...";
            break;
        case 3:
            cout << "Affichage des actions utilisateur...";
    }
}

```

```
        break;
    case 4:
        cout << "Affichage des connexions utilisateur...";
        break;
    case 0:
        cout << "Au revoir !";
        break;
    default: // Le 'default' est l'equivalent du 'else' final.
        cout << "Erreur ! Le choix " << choix << " n'est pas un choix valide !";
    }
}
```