



**Ilyas R’kiouak**

**Julian Wickeur**

**06/01/2023**

**Table des matières**

[Introduction 2](#_heading=h.30j0zll)

[Problème 2](#_heading=h.1fob9te)

[Equipe : 4](#_heading=h.3znysh7)

[Méthode de travail 4](#_heading=h.2et92p0)

[Solution 5](#_heading=h.tyjcwt)

[Exemple après création d’un graphique (avant la coloration): 5](#_heading=h.3dy6vkm)

[La fonctions création du graph : (ligne code 30 à 81) 6](#_heading=h.1t3h5sf)

[La fonctions lecture de fichier : (ligne code 84 à 221) 7](#_heading=h.4d34og8)

[La fonction modifier un fichier : (ligne 222 à 305) 8](#_heading=h.2s8eyo1)

[La fonctions Welsh-Powell : (ligne 306 à 469) 8](#_heading=h.17dp8vu)

[La fonction d’affichage de la coloration : (ligne 306 à 469) 9](#_heading=h.3rdcrjn)

[Présentation des différents algorithmes 10](#_heading=h.26in1rg)

[L'algorithme de Welsh-Powell 10](#_heading=h.lnxbz9)

[L'algorithme de Hill-Climbing 11](#_heading=h.35nkun2)

[Diagramme d’activité : 12](#_heading=h.1ksv4uv)

[Diagramme d’activité des fonctions : 12](#_heading=h.44sinio)

[Le diagramme d'activités de notre programme 14](#_heading=h.2jxsxqh)

[Conclusion 15](#_heading=h.z337ya)

[Annexes 15](#_heading=h.3j2qqm3)

## Introduction

### Problème

La coloration de graphe est un problème de graph théorie qui consiste à affecter des couleurs distinctes aux sommets d'un graphe de manière qu'aucun sommet adjacent (c'est-à-dire lié par une arête) n'ait la même couleur.

Le nombre minimum de couleurs nécessaires pour colorer le graphe est appelé chromatique du graphe.

Ce problème est important en informatique, en particulier en intelligence artificielle, car elle est utilisée pour modéliser de nombreux problèmes pratiques, tels que la planification des tâches, la gestion de la bande passante dans les réseaux informatiques et l'allocation des fréquences dans les réseaux de communication sans fil.

Il existe de nombreuses variantes de la coloration de graphe, telles que la coloration bicolore able (également appelée coloration bipartite), qui consiste à colorier les sommets d'un graphe en deux couleurs de manière à ce que tous les sommets soient de couleurs différentes, et la coloration de graphe précisément k, qui consiste à colorier les sommets d'un graphe avec exactement k couleurs de manière à ce qu'aucun sommet adjacent n'ait la même couleur.

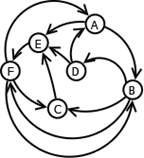
Graphiquement, un graphe se présente avec des ronds (symbolisant les sommets) et des flèches reliant des sommets et symbolisant les arcs.

Figure 1- Exemple de graphe

L’objectif du problème de coloration de graphe et de trouver une affectation des sommets du graphe de sorte que deux sommets reliées par un arc soit affectées dans des groupes différents et que le **nombre de groupes** soit le plus petit possible. Les groupes peuvent être identifier à une couleur ; d’où le nom de problème de coloration de graphe.

Ce problème théorique présente de nombreuses applications (optimisation lors de la compilation de programme, transport de matière première, …). Ce problème et l’approche de Welsh-Powell ont été au programme de spécialité - mathématiques des séries ES.

### Equipe :

Julian Wickeur - Étudiant en L3 SDN à l'Université Chatolique de Lille

Ilyas R’kiouak - Étudiant en L3 SDN à l'Université Chatolique de Lille

### Méthode de travail

1. Étape 1: Définir les objectifs du projet et la stratégie de travail. Nous avons pris le temps de discuter de projet pour déterminer ce que vous voulez accomplir avec notre programme et comment nous allons y parvenir.
2. Étape 2: Établir un plan de travail détaillé. Nous allons décider quelles tâches qu’on va réaliser et dans quel ordre,
3. Étape 3: Développer le programme. Travaillez en parallèle sur les différentes parties du programme. Nous avons assuré de nous coordonner efficacement pour éviter les doublons ou les conflits.
4. Étape 4: Tester et déboguer le programme. Lorsque nous avons terminé de développer le programme, nous allons le tester soigneusement pour nous assurer qu'il fonctionne correctement et qu'il n'y a pas de bugs.
5. Étape 5: Rédiger la documentation et la présentation finale. Nous allons prendre le temps de rédiger une documentation détaillée du programme pour présenter nos résultats.

## Solution

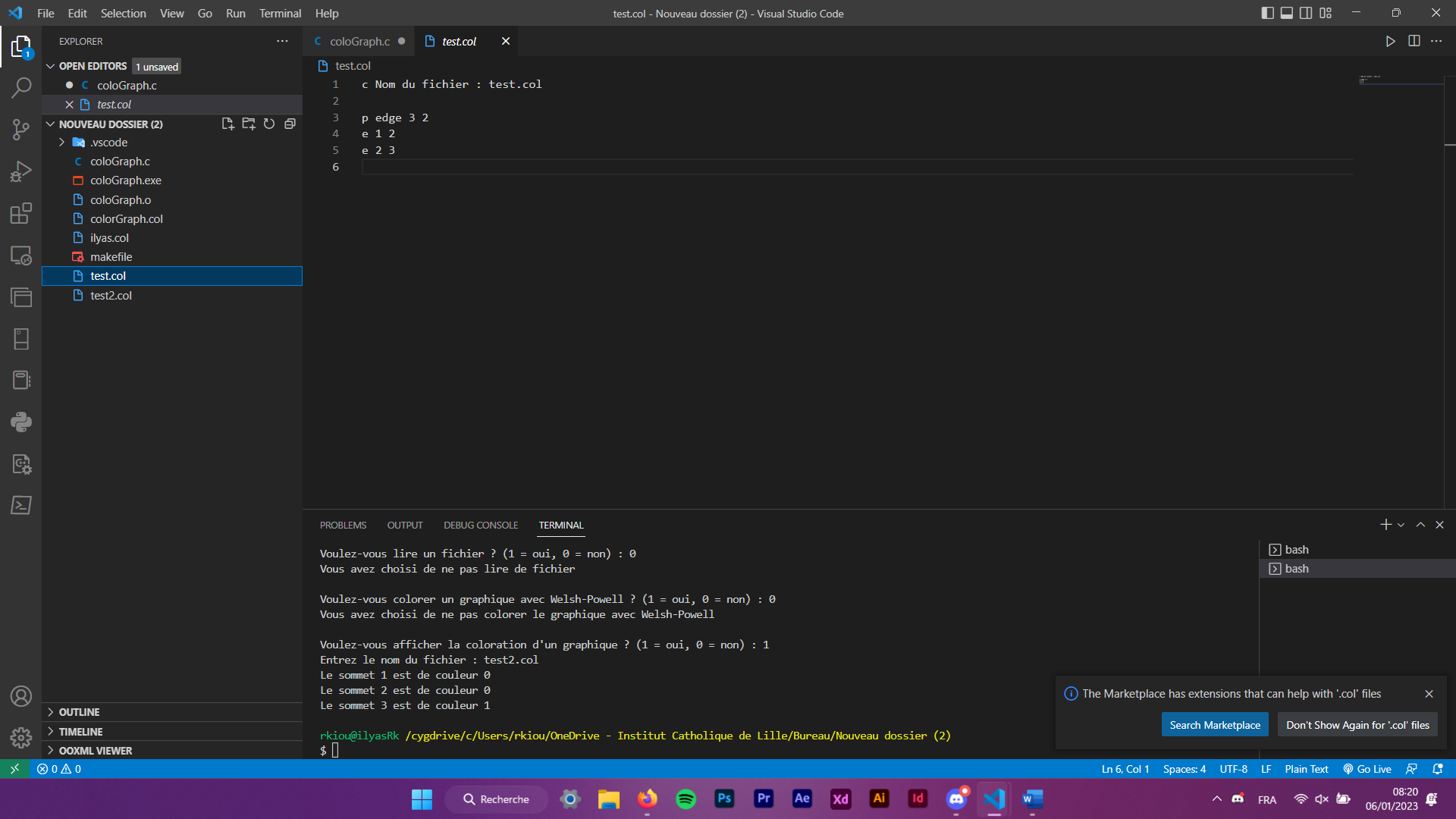
Avant de trouver une solution à notre projet, nous devons bien comprendre la problématique,

Pour débuter le projet, nous nous sommes demandé à quoi aller ressembler les fichiers des graphiques. Pour cela, on s’est appuyé sur les ressources qui étaient disponibles dans le l'énoncé du projet ([Vertex Coloring - Graph Coloring Benchmarks](https://sites.google.com/site/graphcoloring/vertex-coloring)).

Grâce à cela, nous avons pu définir un format de création et lecture de graph prédéfini pour le fonctionnement de notre programme :

* Les lignes qui commencent par un “c” sont les commentaires.
* Ils n’ont pas d’impact sur la lecture effectué par le programme
* La ligne qui commence par “p edge” est le nombre de sommets et de liaisons du graphique
* Cette ligne sera la seule du fichier qui commencera par “p”.
* Le 1er chiffre correspond aux nombres de sommets du graphique.
* Le 2e chiffre correspond aux nombres de liaisons du graphique.
* Ceux qui commencent par un “e” sont les liaisons :
* Le 1er chiffre correspond au nom du sommet où la liaison commence.
* Le 2e chiffre correspond au nom du sommet où la liaison se termine.

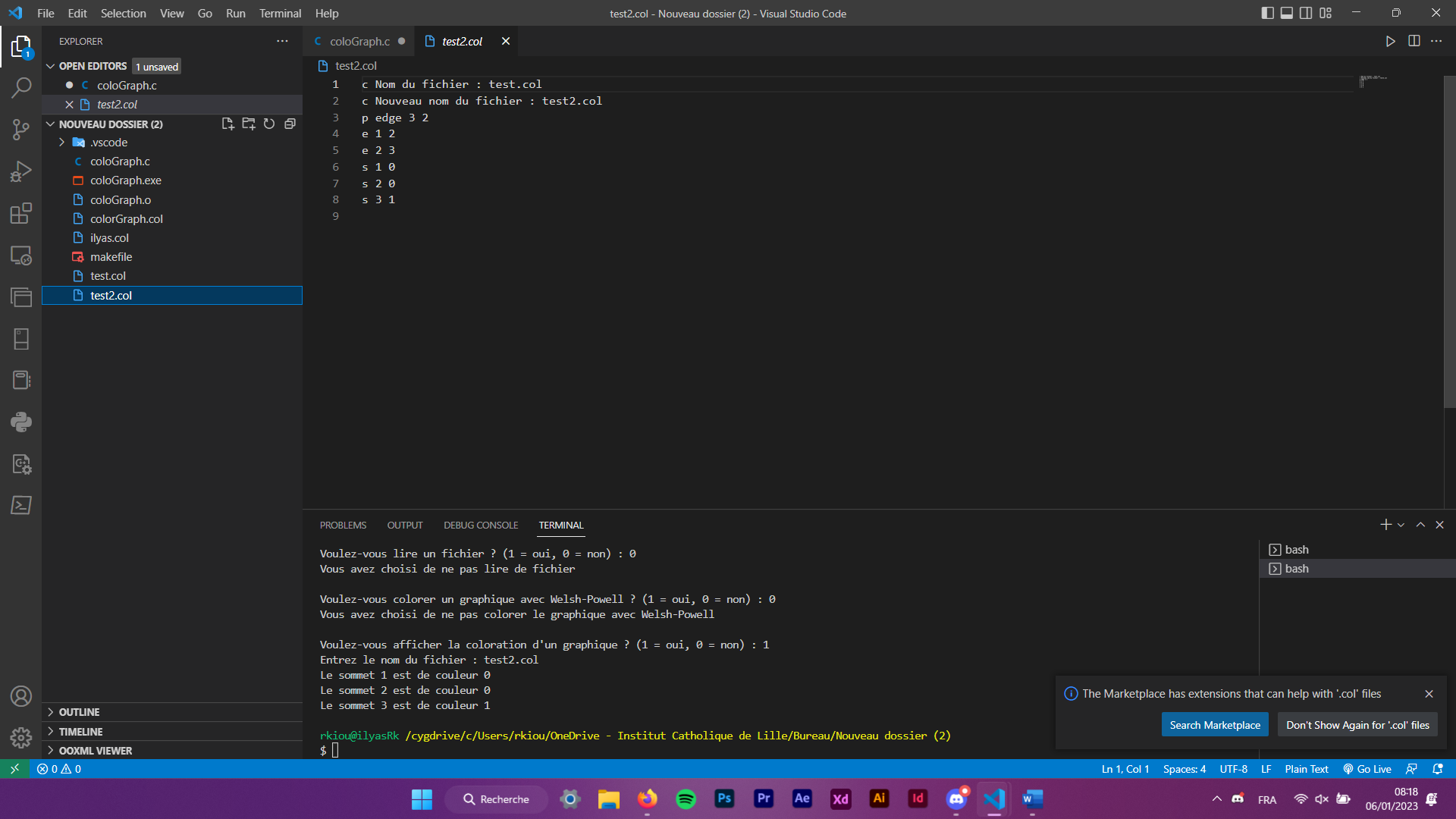
### Exemple après création d’un graphique (avant la coloration) :



**Après la coloration :**

Pour la coloration du graphique, nous avons un nouveau type sur le format du fichier :

* Une nouvelle ligne de commentaire est écrite sur le nouveau fichier avec son nom.
* Les lignes “s” correspondent à la coloration d’un sommet. Le premier chiffre est le nom du sommet en question et le deuxième correspond à la couleur de ce sommet.



Ensuite nous avons commencé à travailler sur la première partie de notre programme.

### La fonctions création du graph : (ligne code 30 à 81)

Etape 1 :

Le programme demande à l’utilisateur s’il veut créer un fichier. Pour répondre au programme, 2 choix nous est proposé :

* Accepter la requête, on écrit 1.
* Refuser la requête, on écrit 0.

Si la réponse est oui, l'utilisateur doit entrer le nom du fichier qui sera créé. Le fichier aura par défaut un format “.col”.

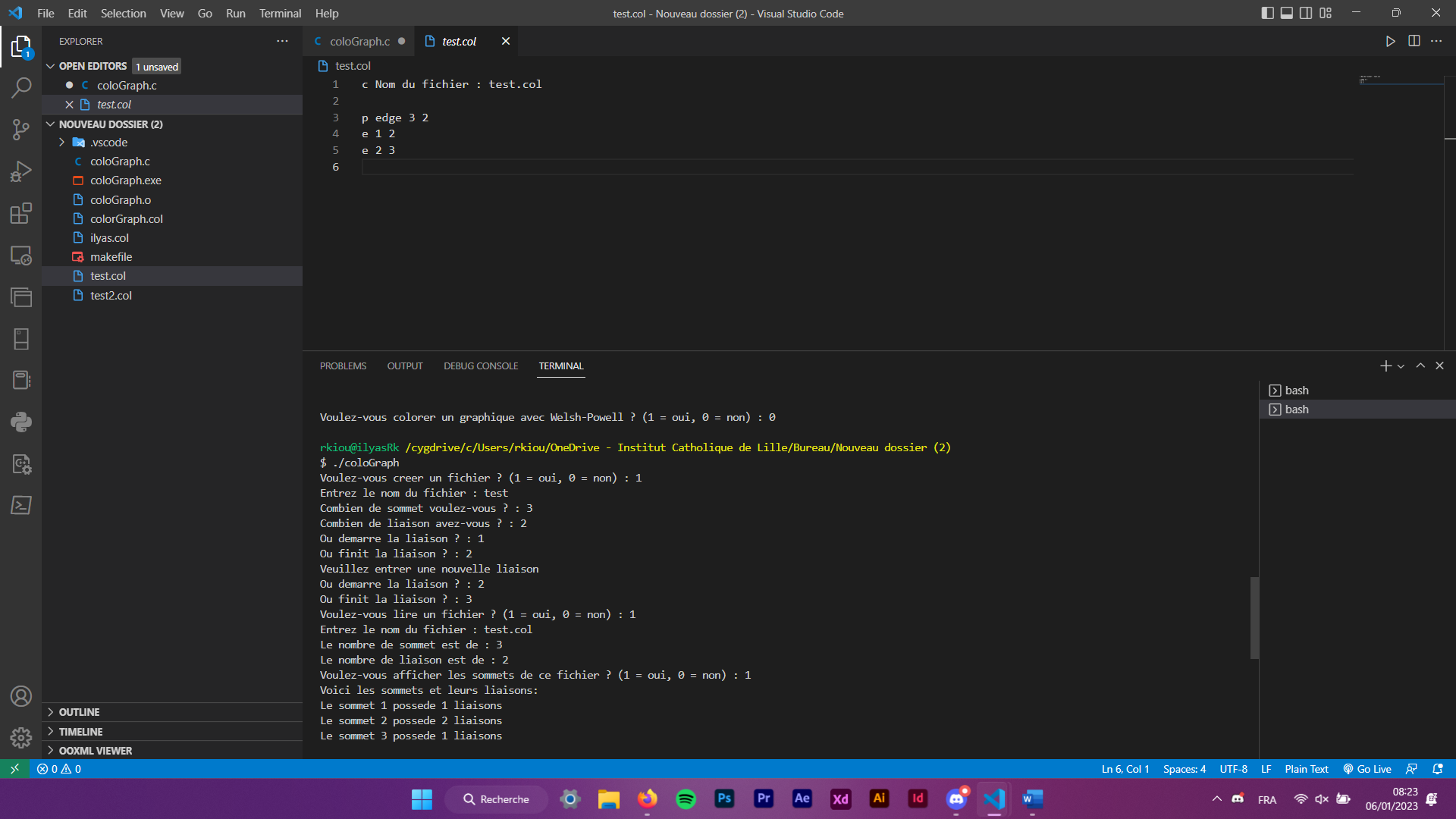
Sinon la demande est terminée et le terminal nous indique que nous avons décidé de ne pas créer de fichier.

Etape 2 :

Le programme demande combien de sommets et de liaisons l’utilisateur souhaite avoir son graphique.

Etape 3 :

Le programme demande où démarre la première liaison et où elle se termine. Ensuite cette question se répète autant de fois qu’il y a de liaisons qui avaient été définies précédemment.



### La fonctions lecture de fichier : (ligne code 84 à 221)

Etape 1 :

Le terminal propose à l’utilisateur s’il veut lire un fichier existant (le fichier doit se trouver dans le même dossier que le programme).

Si oui, l’utilisateur doit entrer le nom du fichier qu’il veut lire. (Ne pas oublier d’écrire le nom du fichier entier avec son format. Ex : “test.col”)

Sinon le terminal nous indique que l’utilisateur a choisi de ne pas lire de fichier. La fonctionnalité de lecture se ferme.

Etape 2 :

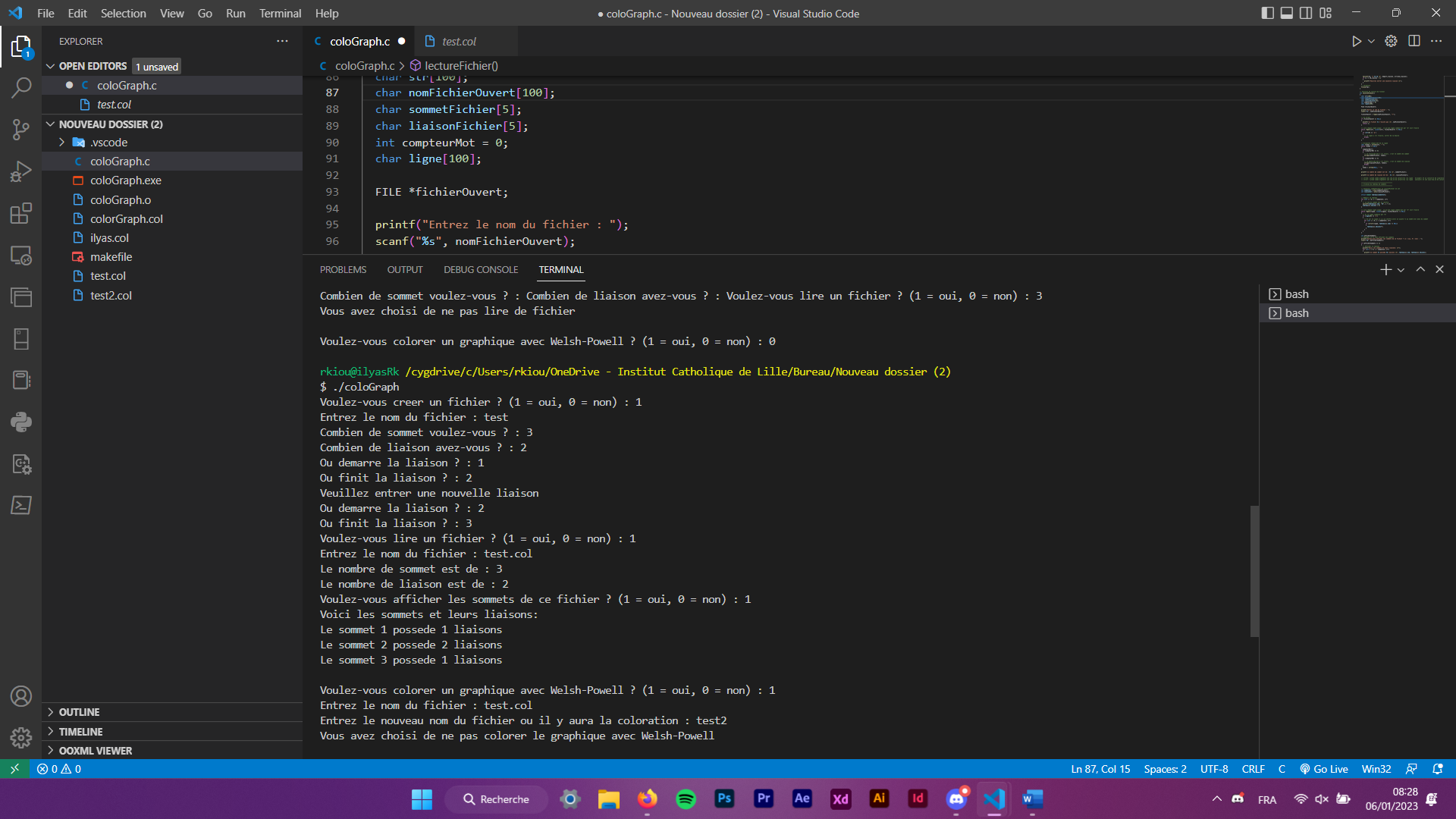
Le programme récupère les informations du fichier entré et nous affiche le nombre de sommets et de liaisons du graphique.

Etape 3 :

Le terminal propose à l’utilisateur s’il veut afficher les sommets du graphique en lecture.

Si oui, le programme analyse chaque sommet du graphique et il affiche le nombre de liaisons que chaque sommet possède.

Sinon le terminal nous indique que l’utilisateur a choisi de ne pas lire les sommets. La fonctionnalité de lecture se ferme.



### La fonction modifier un fichier : (ligne 222 à 305)

Malgré nos efforts, nous n'avons pas réussi à terminer le développement de la fonction qui devait permettre de modifier un fichier. Nous avons toutefois réussi à avancer dans son développement et avons laissé en commentaire le code d'essai que nous avons rédigé. La complexité de la tâche et le manque de temps ont malheureusement empêché sa finalisation.

### La fonctions Welsh-Powell : (ligne 306 à 469)

Etape 1:

Le terminal propose à l’utilisateur s'il veut colorier le graphe avec algorithme de Welsh-Powell

Si oui, l’utilisateur doit entrer le nom du fichier qu’il veut lire. (Ne pas oublier d’écrire le nom du fichier entier avec son format. ex : “test.col”).

Etape 2:

Ensuite le terminal va demander à l'utilisateur de rentrer le nouveau nom du fichier où il y aura la coloration.

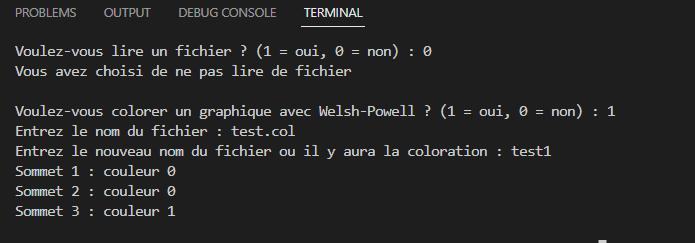
Explication :

* Le programme va créer une copie du fichier sélectionné par l’utilisateur.
* Cette copie sera éditée pour avoir les informations de colorations du fichier d’origine.
* Un nouveau type de ligne sera créé pour accueillir les données de couleurs. (les lignes qui commencent par “s”)
* Une nouvelle ligne de commentaire “c” sera écrite sur le fichier copié avec le nouveau nom de ce fichier.

Etape 3:

Le terminal affiche les sommets et la couleur associé aux sommets correspondants.

On pourrait s’attendre à qu’il nous affiche une vraie couleur comme “rouge” par exemple mais il revoit seulement la valeur correspondant à sa couleur qui est enregistrée dans le fichier. Lors de ce projet, nous ne nous sommes pas penchés sur la traduction la valeur trouvé et l’affichage qu’il va retourner.

****

**La fonctions Hill-Climbing : (ligne 472 à 533)**

Malheureusement, nous n'avons pas pu compléter l'implémentation de l'algorithme de Hill-Climbing pour la coloration de graphe en C. Cependant, nous avons commencé à développer une solution qui est actuellement en commentaire dans le code.



### La fonction d’affichage de la coloration : (ligne 306 à 469)

Etape 1:

Le terminal propose à l’utilisateur s’il veut afficher la coloration d’un graphe.

Si oui, l’utilisateur doit entrer le nom du fichier qu’il veut lire. (Ne pas oublier d’écrire le nom du fichier entier avec son format. Ex : “test.col”)

Etape 2:

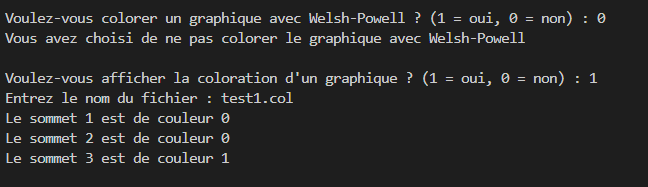
Le terminal affiche les sommets et les couleurs.

Erreur remarquée :

* Juste après la colorisation d’un graphique. Le fichier qui a été colorié n’arrive pas à être disponible par la fonction d’affichage de coloration d’un graphique.
* Il faut relancer le programme et ensuite ce fichier pourra être lu par la fonction d’affichage de coloration d’un graphique.

Hypothèse :

* L’erreur doit avoir surement un lien avec le fait qu’on créé un copie d’un fichier et qu’il est donc disponible à la lecture qu’à la fin du programme.

****

## Présentation des différents algorithmes

### L'algorithme de Welsh-Powell

L'algorithme de Welsh-Powell est un algorithme de coloration de graphes qui utilise un nombre minimal de couleurs pour colorer le graphe de manière à ce qu'aucun sommet adjacent n'ait la même couleur.

Voici les étapes de l'algorithme de Welsh-Powell :

* Parcours en profondeur du graphe : l'algorithme commence par parcourir le graphe en utilisant un parcours en profondeur (DFS). Cela permet de s'assurer que les sommets seront traités dans l'ordre approprié lors de l'attribution des couleurs.
* Tri des sommets par degré décroissant : une fois le parcours en profondeur terminé, l'algorithme trie les sommets du graphe par degré décroissant. Le degré d'un sommet est le nombre d'arêtes qui y sont adjacentes.
* Attribution de couleurs : l'algorithme commence par attribuer une couleur au sommet ayant le plus grand degré. Il vérifie ensuite quelles couleurs sont disponibles pour les sommets adjacents, et attribue la couleur la plus petite possible au sommet. Cette étape est répétée jusqu'à ce que tous les sommets aient été colorés.
* Répétition de l'attribution de couleurs : si le graphe n'est pas encore coloré de manière propre (c'est-à-dire que deux sommets adjacents n'ont pas la même couleur), l'algorithme répète l'étape 3 jusqu'à ce que le graphe soit coloré de manière propre. Cela garantit que le nombre de couleurs utilisées est minimal.

Une fois que le graphe a été coloré, l'algorithme affiche la couleur de chaque sommet.

### L'algorithme de Hill-Climbing

L'algorithme de Hill-Climbing est un algorithme de recherche de solution qui consiste à explorer l'espace de recherche de manière itérative et à trouver la solution optimale en modifiant légèrement la solution actuelle à chaque itération.

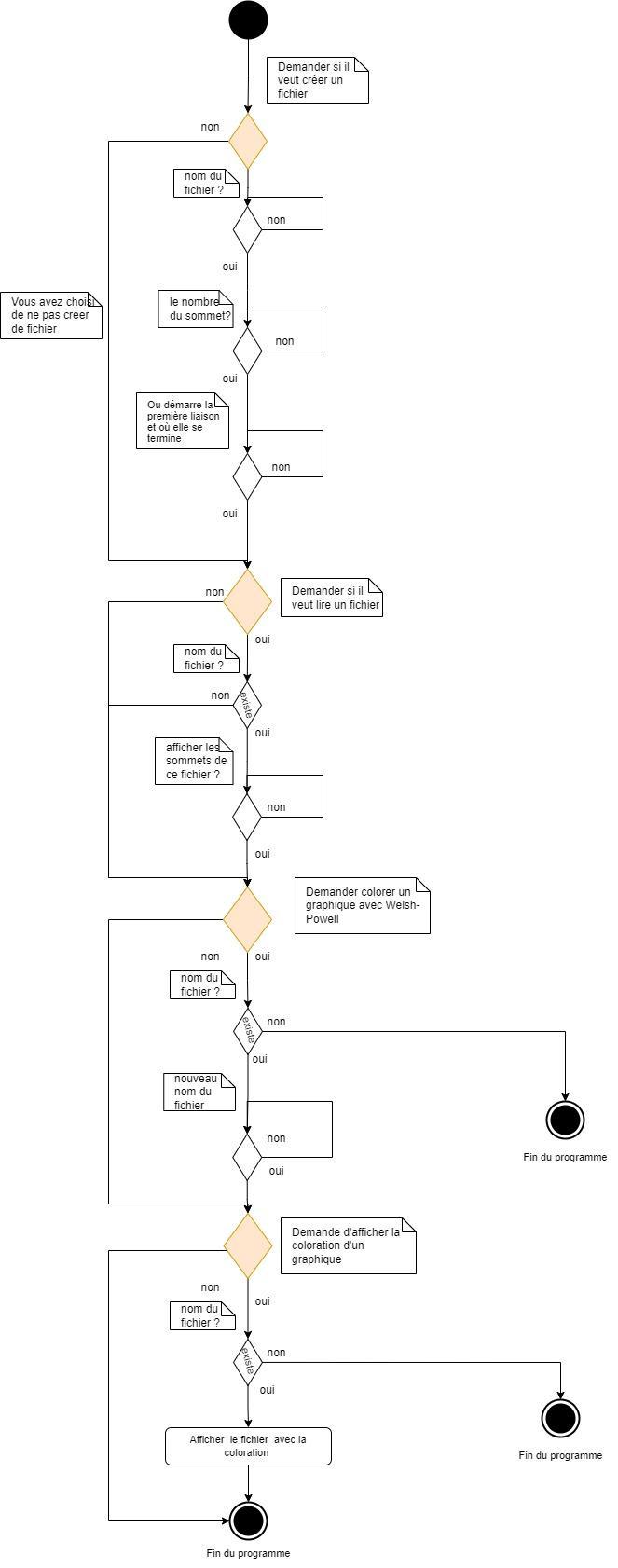
Voici les étapes de l'algorithme de Hill-Climbing :

1. Initialisation de la solution actuelle et de la meilleure solution trouvée jusqu'à présent : l'algorithme génère une solution aléatoire et l'évalue pour initialiser la solution actuelle et la meilleure solution trouvée jusqu'à présent.
2. Itération de l'algorithme : l'algorithme répète l'étape suivante jusqu'à ce qu'un nombre maximum d'itérations soit atteint.
3. Modification de l'état actuel : l'algorithme modifie légèrement l'état actuel en modifiant aléatoirement un élément de l'espace de recherche.
4. Évaluation de la nouvelle solution : l'algorithme évalue la nouvelle solution en calculant son coût.
5. Décision de conserver la nouvelle solution : l'algorithme décide si la nouvelle solution doit être conservée ou non en utilisant une fonction de probabilité dépendante de la température. Si la nouvelle solution est meilleure que la solution actuelle ou si la probabilité de la conserver est supérieure à un nombre aléatoire compris entre 0 et 1, l'algorithme conserve la nouvelle solution. Sinon, il restaure l'état précédent.
6. Mise à jour de la meilleure solution trouvée jusqu'à présent : si la nouvelle solution est meilleure que la meilleure solution trouvée jusqu'à présent, l'algorithme met à jour la meilleure solution et la stocke.
7. Mise à jour de la température : l'algorithme met à jour la température en la multipliant par un facteur de décroissance défini par l'utilisateur.

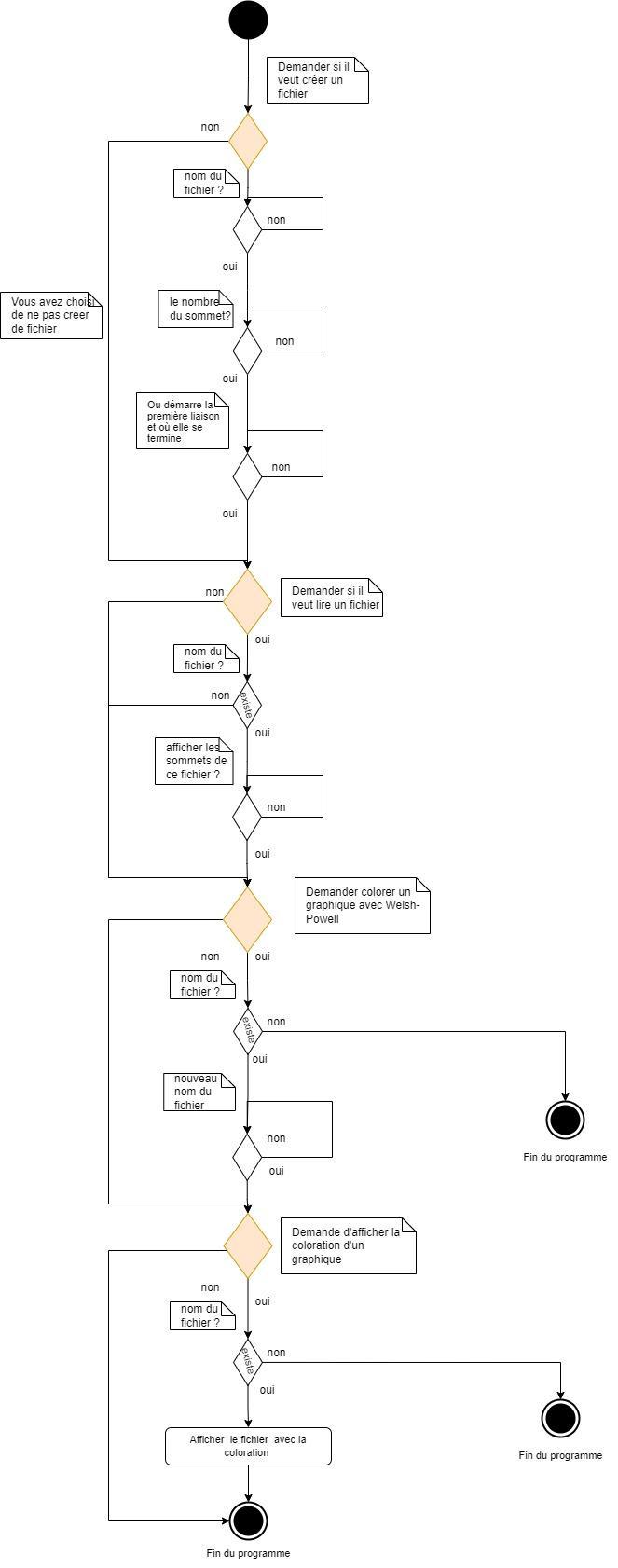
Une fois que l'algorithme a terminé ses itérations, il affiche la meilleure solution trouvée et son coût.

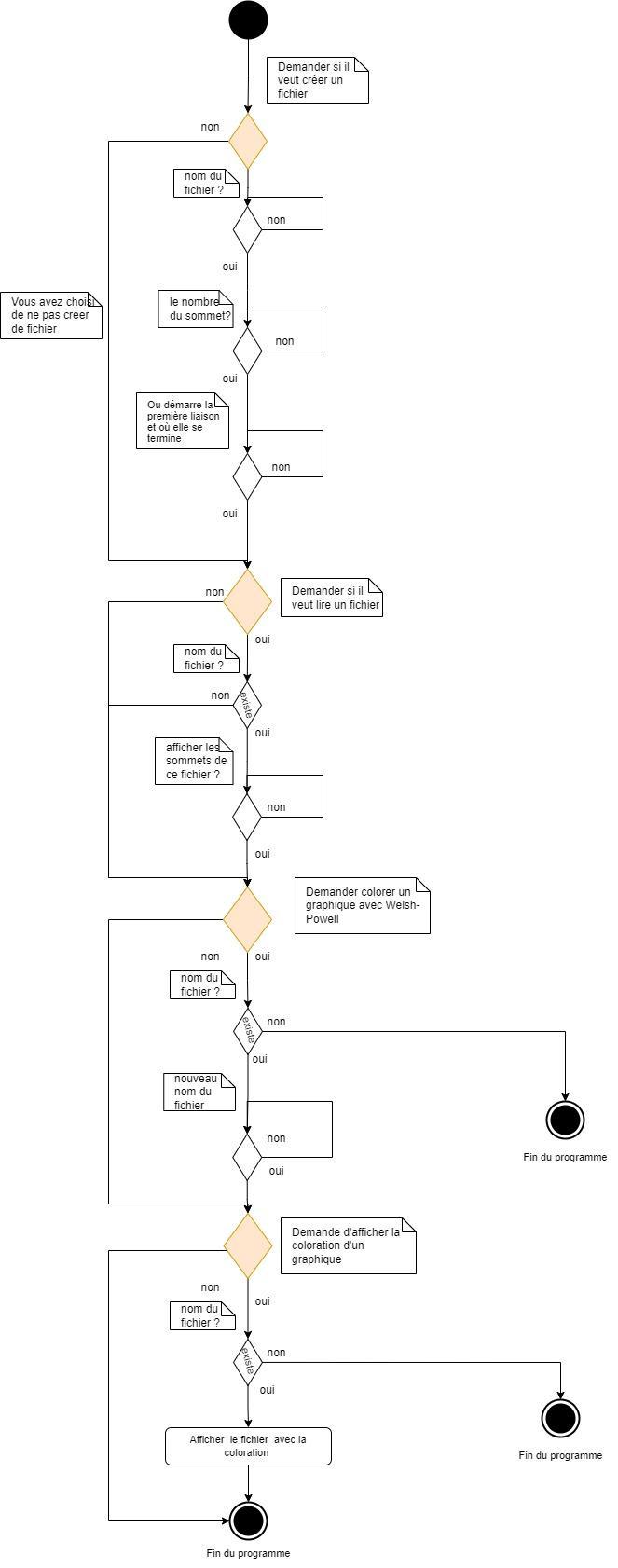
## Diagramme d’activité :

### Diagramme d’activité des fonctions :

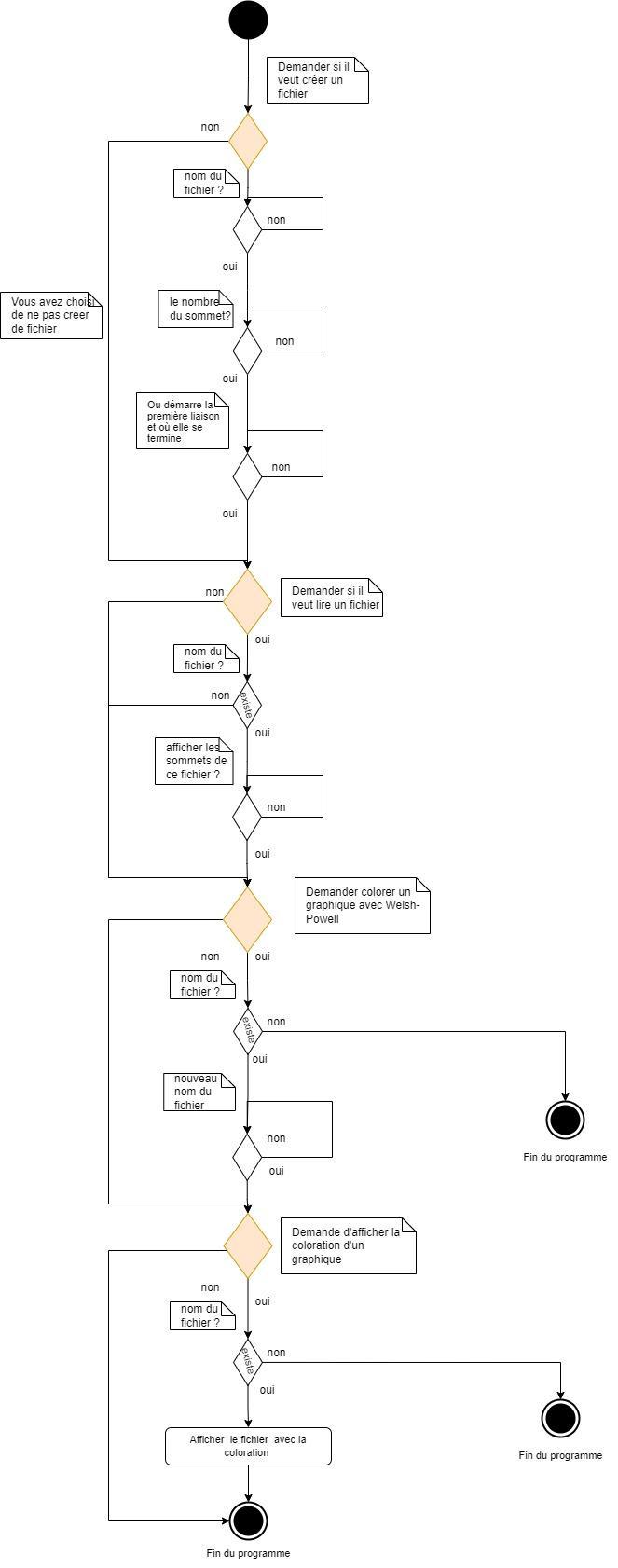
**La fonctions création du graph :**

**La fonctions lecture de fichier :**



**La fonctions Welsh-Powell**

**La fonction d’affichage de la coloration**



### Le diagramme d'activités de notre programme

## Conclusion

En réalisant ce projet, nous avons développé un programme capable de créer, lire, modifier et afficher des graphes. Nous avons également implémenté deux algorithmes de coloration de graphe, à savoir Welsh-Powell et Hill-Climbing, qui nous ont permis de trouver des colorations optimales ou approximatives pour des graphiques donnés. Grâce à ces fonctionnalités, notre programme est capable de résoudre de manière efficace le problème de coloration de graphe, ce qui peut avoir de nombreuses applications pratiques dans différents domaines tels que l'optimisation de la compilation de programmes ou le transport de matières premières.

En travaillant sur ce projet, nous avons acquis de solides connaissances en recherche opérationnelle et coloration de graphe, qui nous permettront de poursuivre notre apprentissage et de développer de nouvelles compétences.

Nous sommes fiers de ce que nous avons accompli et espérons pouvoir mettre nos connaissances en pratique dans de futurs projets.

## Annexes

[https://fr.wikipedia.org/wiki/Coloration\_de\_graphe#](https://fr.wikipedia.org/wiki/Coloration_de_graphe)

<https://fr.wikipedia.org/wiki/Coloration_des_ar%C3%AAtes_d%27un_graphe>

<https://fr.wikipedia.org/wiki/Coloration_%C3%A9quitable>

<https://sites.google.com/site/graphcoloring/vertex-coloring>