

# AVL Debugging - TP 1

Hatim M'RABET EL KHOMSSI - M2 GL

mrabet.hatim@gmail.com

## Bug 1: Aucun affichage

1- Rapport de Bug :

Information Technique :

Je lance le programme sur un Linux Ubuntu 20.04.5 LTS, et jecompile avec gcc (Ubuntu 9.4.0-1ubuntu1~20.04.1) et le code source est dans le fichier [anagrams-IAGL.c](#) et [anagrams-IAGL.h](#).

Description:

Le programme est censé afficher les groupes d'anagrammes à partir d'une phrase, mais lors de l'exécution, il n'affichent rien sur l'écran, et le programme se termine sans erreur.

le résultat attendu :

```
{chien niche chine}
{marche charme}
{une nue}
{limace malice}
```

Les étapes pour reproduire le bug:

```
gcc -g anagrams-IAGL.c -o anagrams-IAGL
./anagrams-IAGL
```

Gravité:

le bug est majeur, car le programme ne fait pas ce qu'il est censé faire, donc il perd sa fonctionnalité

2- Enquête

Pour chercher le problème, j'ai utilisée gdb pour regarder la valeur de `anagrams_size` dans la fonction `print_words_anagrams` qui n'a jamais dépassé 1, et par la suite, j'ai remarqué qu'il y a une condition `if` sans double égale, qui surement pose un problème car ca fait une affectation au lieu d'une comparaison.

3- Correction du bug

Dans la fonction `word_in_array` du fichier [anagrams-IAGL.c](#), on a une condition if qui au lieu de faire une comparaison, elle faisait de l'affectation, donc on a remplacé `=` par `==`. Apres cette modification, on a eu l'affichage suivant:

```
{
chien
niche
chine
Segmentation fault (core dumped)
```

4- Pour tester, j'ai lancé le programme qui m'a affiché quelques valeurs, et un probleme de segementation fault, ce qui veut dire que le bug d'affichage est corrigé.

## Bug 2: Segmentation Fault

1- Rapport de Bug

Information Technique:

Je lance le programme sur un Linux Ubuntu 20.04.5 LTS, et jecompile avec gcc (Ubuntu 9.4.0-1ubuntu1~20.04.1) et le code source est dans le fichier [anagrams-IAGL.c](#) et [anagrams-IAGL.h](#).

Description:

le programme est censé afficher les groupes d'anagrammes à partir d'une phrase, mais lors de l'exécution il affiche un message d'erreur `Segmentation fault (core dumped)`.

le résultat attendu :

```
{chien niche chine}
{marche charme}
{une nue}
{limace malice}
```

mais on a :

```
{
chien
niche
chine
Segmentation fault (core dumped)
```

Les étapes pour reproduire le bug:

```
gcc -g anagrams-IAGL.c -o anagrams-IAGL
./anagrams-IAGL
```

Gravité:

le bug est critique, car le programme crash, et on ne peut pas l'utiliser.

2- Enquête

Le segmentation fault vient souvent à cause d'un access au memoire non initialisé ou interdit. Pour cela, j'ai regardé où on a manipuler des pointeurs. et dans la fonction `print_words`, j'ai regardé les valeurs du `i` et de `words_size` avec gdb dans la boucle et j'ai remarquer qu'il essaie d'accéder à la case 4 d'un tableau de 4 case, ce qui n'est pas possible.

3- Correction du bug

Dans la fonction `print_words` du fichier `anagrams-IAGL.c`, on a une boucle qui parcourait le tableau `words` de 0 à `words_size` donc on essaye d'accéder à une case qui n'existe pas, ce qui déclenche l'erreur `Segmentation fault (core dumped)`. Pour cela, on a remplacer la condition de la boucle `i <= words_size` par `i < words_size`.

Apres cette modification, on a eu l'affichage suivant.

```
{
chien
niche
chine
}
{
marche
charme
}
{
chien
niche
chine
}
{
une
nue
}
{
limace
malice
}
{
de
de
}
{
chien
niche
chine
}
{
une
nue
}
{
de
de
}
{
limace
malice
}
{
marche
charme
}
```

4- Pour tester, il faut relancer le programme avec a même maniere, et je remarque que ca affiche plus d'anagrammes, et on a aucune erreur, ce qui veut dire que le bug du segmentation fault est réglé.

Bug 3: Affichage redondant

1- Rapport de Bug

Information Technique:

Je lance le programme sur un Linux Ubuntu 20.04.5 LTS, et je compile avec gcc (Ubuntu 9.4.0-1ubuntu1~20.04.1) et le code source est dans le fichier `anagrams-IAGL.c` et `anagrams-IAGL.h`.

Description:

le programme est sencé afficher les groupes d'anagrammes à partir d'une phrase, mais lors de l'execution il affiche des groupes d'anagrammes redondants.

le résultat attendu :

```
{chien niche chine}
{marche charme}
{une nue}
{limace malice}
```

Mais on a :

```
{
chien
niche
chine
}
{
marche
charme
}
{
chien
niche
chine
}
{
une
nue
}
{
limace
```

```
malice
}
{
de
de
}
{
chien
niche
chine
}
{
une
nue
}
{
de
de
}
{
limace
malice
}
{
marche
charme
}
```

Les étapes pour reproduire le bug:

```
gcc -g anagrams-IAGL.c -o anagrams-IAGL
./anagrams-IAGL
```

Gravité:

le bug est majeur, car le programme n’affiche pas les valeurs attendues.

2- Pour ce bug, j’avais déjà un présentiment par rapport à la condition if que j’ai fixé dans le premier bug, et qui pour moi il comparait pas vraiment les 2 chaines de caractères. Avec gdb, j’ai affiché la valeur des 2 mots, et puis, je regarde si quand les 2 chaines identiques ca passe en true ou pas.

3- Correction du bug

Dans la fonction `word_in_array` on a une comparaison entre 2 pointeurs vers des chaines de caracteres, donc au lieu d'utiliser `array[i] == word` qui compare l'adresse des 2 mots, on a utilisé `strcmp(array[i], word) == 0` pour comparer les 2 chaines de caracteres.

ce qui nous a permit d'avoir l’affichage suivant :

```
{
chien
niche
chine
}
{
marche
charme
}
{
une
nue
}
{
limace
malice
}
{
de
de
}
```

4- Pour tester que c’était vraiment ca le problème, on relance le programme de la même manière, et on regarde le resultat. sinon, j’ai retester de la meme maniere la condition pour voir si ma modification a changé le comportement du code, et si sa fonctionne de la maniere voulu.

### Bug 4: Même mots dans le même groupe

1- Rapport de Bug

Information Technique:

Je lance le programme sur un Linux Ubuntu 20.04.5 LTS, et on compile avec gcc (Ubuntu 9.4.0-1ubuntu1~20.04.1) et le code source est dans le fichier [anagrams-IAGL.c](#) et [anagrams-IAGL.h](#).

Description:

le programme est sencé afficher les groupes d’anagrammes à partir d'une phrase, mais lors de l'execution, je remarque qu’il affiche un groupe avec 2 fois le meme mot “de”.

le résultat attendu :

```
{chien niche chine}
{marche charme}
{une nue}
{limace malice}
```

mais on a :

```
{
chien
niche
chine
}
{
marche
charme
```

```
}
{
une
nue
}
{
limace
malice
}
{
de
de
}
```

Les étapes pour reproduire le bug:

```
gcc -g anagrams-IAGL.c -o anagrams-IAGL
./anagrams-IAGL
```

Gravité:

Le bug est majeur, car le programme n'affiche pas les valeurs attendues.

2- Enquête

J'ai regardé dans le code la fonction qui ajoute un anagramme, et j'ai remarqué qu'il n y a pas de fonction de verification si le meme mots existe deja, du coup, normalement c'est ca le probleme.

3- Correction du bug

Dans la fonction `print_words_anagrams`, j'ai ajouté une autre condition lors de l'ajout d'un mot à une anagramme, pour verifier si le mot n'est pas déjà dans le groupe d'anagramme.

```
if(strcmp(sig, signature) == 0 && !word_in_array(anagrams, anagrams_size, words[j]))
```

ce qui nous a permit d'avoir l'affichage suivant :

```
{
chien
niche
chine
}
{
marche
charme
}
{
une
nue
}
{
limace
malice
}
```

4- Pour tester, je modifie le code et je regarde avec gdb, si quand on a deja le meme mot, il va l'ajouter ou pas, et d'apres, mon observation ca fonctionnait, et on a le résultat voulu.

Bug 5: Mauvaise affichage

1- Rapport de Bug

Information Technique:

Je lance le programme sur un Linux Ubuntu 20.04.5 LTS, et je compile avec gcc (Ubuntu 9.4.0-1ubuntu1~20.04.1) et le code source est dans le fichier [anagrams-IAGL.c](#) et [anagrams-IAGL.h](#).

Description:

le programme est sencé afficher les groupes d'anagrammes à partir d'une phrase avec un groupe par ligne, mais lors de l'execution il affiche les groupes d'anagrammes avec un mot par ligne.

le résultat attendu :

```
{chien niche chine}
{marche charme}
{une nue}
{limace malice}
```

mais on a :

```
{
chien
niche
chine
}
{
marche
charme
}
{
une
nue
}
{
limace
malice
}
```

Les étapes pour reproduire le bug:

```
gcc -g anagrams-IAGL.c -o anagrams-IAGL
./anagrams-IAGL
```

Gravité:

Le bug est mineur, car le programme affiche les valeurs attendues mais c'est juste la manière d'affichage qui pose problème.

## 2- Enquête

Le problème d'affichage est lié à un saut de ligne qui vient après chaque mot, ce qui peut être dans la fonction `print_words`.

## 3- Correction du bug

Pour résoudre ce problème, on a juste enlever un saut de ligne `\n` après l'ouverture d'une accolade `{` dans la fonction `print_words_anagrams` et enlever celle dans la fonction `print_words`, ce qui nous a permis d'avoir l'affichage attendu suivant :

```
{chien niche chine}
{marche charme}
{une nue}
{limace malice}
```

4- Pour tester, j'ai relancé le code, et j'ai eu le même résultat attendu.