

1. Corrections

Erreurs détectées par Valgrind sur l'ensemble des 15 fichiers de configuration de code de l'application ASCQ_ME

En travaillant sous Windows, dans le sous-système Linux de Windows (WSL), nous avons commencé par télécharger l'application ASCQ_ME fourni pour cet étude de cas.

Une fois compilée via le Makefile existant(en exécutant la commande <<make>> dans l'invite de commande) , nous avons commencé à tester les fichiers en appelant l'exécutable avec chacun des 15 fichiers de configurations fournis par défaut dans le répertoire <<_configuration>>.

Dans le terminal des commandes, nous avons exécuté la commande suivante pour chaque fichier de configuration :

```
valgrind --show-leak=all --leak-check=full -v
```

```
./ascq_me_configuration/NOMFICHIER.config
```

Par exemple, pour le fichier <<default_configuration>> :

```
valgrind --show-leak=all --leak-check=full -v
```

```
./ascq_me_configuration/default_configuration.config
```

À l'aide de l'affichage de Valgrind pour chaque fichier, nous avons pu détecter les erreurs qui se sont trouvés dans le code de l'application, en suivant ligne par ligne l'affichage des erreurs de mémoire et d'autres problèmes avec les sections : "at..." - indication qui montre où l'erreur a été détectée dans le code; et "by..." - indication qui est généralement utilisée pour indiquer la source de l'erreur, elle montre quelle partie du code a causé l'erreur.

Nous avons identifier les erreurs suivantes et nous les avons corrigés comme suit :

ERREUR 1:

Erreur d'allocation mémoire, dans le fichier <<**digest.cpp**>> à la ligne 126.

```
// C'est parti pour le grand n'importe quoi
Peptides = (peptide_t *) malloc (32 * sizeof (char));
//Peptides = (peptide_t *) malloc (NbPeptides * sizeof (peptide_t));
```

Pour la corrigée, nous avons mis en commentaire la ligne 126 avec l'allocation mémoire mal faite, et nous avons enlever les commentaires de la ligne 127 en laissant la bonne allocation mémoire à prendre sa place.

Le résultat pour cette partie de code est le suivant :

```
// C'est parti pour le grand n'importe quoi
//Peptides = (peptide_t *) malloc (32 * sizeof (char));
Peptides = (peptide_t *) malloc (NbPeptides * sizeof (peptide_t));
```

ERREUR 2:

Erreur d'allocation mémoire (dépassement de la taille mémoire d'allocation) , dans le fichier <<**isodistrib.cpp**>> à la ligne 1531.

```
for(i=0;i<N_SPEC;i++)
{
t_exp_spec[i].Re=-1.0; /* init with -1 flag for futur interpolation */
// Pourquoi pas ?
t_exp_spec[i+1].Im=0.0;
}
```

Nous avons corrigé en remarquant que la taille de tableau est dépassée pour "i+1", donc l'allocation mémoire va échouée. Du coup c'est nécessaire de transformer l'instruction en appelant l'élément "i "de tableau, comme suit :

```
for(i=0;i<N_SPEC;i++)
{
t_exp_spec[i].Re=-1.0; /* init with -1 flag for futur interpolation */
// Pourquoi pas ?
t_exp_spec[i].Im=0.0;
}
```

ERREUR 3:

Erreur d'allocation mémoire dans le fichier <<**ascq_me_configuration.cpp**>> à la ligne 377, un problème avec realloc().

```
// BOUH !!!!
//free(unlimited_buffer);
```

Pour libéré l'espace mémoire alloué pour le buffer, on doit faire l'instruction "free", donc on a enlevé le commentaire :

```
// BOUH !!!!
free(unlimited_buffer);
```

ERREUR 4:

Erreur de type invalide free() / realloc(), dans le fichier <<**formula.cpp**>> à la ligne 559.

```
free(copy);
// Je suis un fou !!!
free(copy);
```

L'espace mémoire est libéré 2 fois, ce qui génère l'erreur. Donc, nous avons mis en commentaire le deuxième free() :

```
free(copy);  
// Je suis un fou !!!  
//free(copy);
```

ERREUR 5:

Erreur d'allocation mémoire de type invalide free() / realloc(), dans le fichier <<seqio.cpp>> à la ligne 273.

```
// C'est pas bien !!!  
/*if(SeqBuffer!=NULL)  
{  
    free(SeqBuffer);  
}*/
```

On l'a corrigé en supprimant les commentaires, pour que l'instruction free() puisse s'exécuter dans le code, pour libérer l'espace mémoire associé au SeqBuffer :

```
// C'est pas bien !!!  
if(SeqBuffer!=NULL)  
{  
    free(SeqBuffer);  
}
```