TP2 – Quality review

EXERCICE 1

Récupérez le fichier « quality.c » sur l'ENT

QUESTION 1

Copier le résultat initial obtenu dans votre rapport.

```
PS X:\Semestre_8\MOQL\TP2\02 - Quality Review> C:\MOQL\vera++\bin\vera++ quality.c
quality.c:1: no copyright notice found
quality.c:4: too many consecutive empty lines
quality.c:6: comma should be followed by whitespace
quality.c:8: trailing whitespace
quality.c:11: trailing whitespace
quality.c:11: horizontal tab used
quality.c:12: keyword 'for' not followed by a single space
quality.c:12: comma should be followed by whitespace
quality.c:12: comma should be followed by whitespace
quality.c:12: comma should be followed by whitespace
quality.c:12: comma should be followed by whitespace
quality.c:12: full block {} expected in the control structure
quality.c:13: keyword 'while' not followed by a single space
quality.c:13: keyword 'while' not followed by a single space
quality.c:13: keyword 'while' not followed by a single space
quality.c:13: comma should be followed by whitespace
quality.c:14: line is longer than 100 characters
quality.c:14: keyword 'for' not followed by a single space
quality.c:14: keyword 'for' not followed by a single space
quality.c:14: keyword 'if' not followed by a single space
quality.c:14: closing curly bracket not in the same line or column
quality.c:14: closing curly bracket not in the same line or column
quality.c:14: full block {} expected in the control structure
quality.c:14: full block {} expected in the control structure
```

QUESTION 2

Améliorez les résultats de clarté et maintenabilité afin de faire disparaitre les différents messages et faire gagner le texte en lisibilité.

Retester régulièrement le code en le recompilant avec le build.bat

Après plusieurs tentatives, nous n'avons plus aucune erreur :

```
PS X:\Semestre_8\MOQL\TP2\02 - Quality Review> C:\MOQL\vera++\bin\vera++ quality.c
PS X:\Semestre_8\MOQL\TP2\02 - Quality Review>
```

Voici notre code modifié :

```c
/*
 * Copyright (c) 2024 BLIGNY PIVON
 */

#include <stdio.h>
#include <string.h>

int main(int argc, char **argv)
{
    long arg[100];
    int i, j, base = 10, start = 0, end = strlen(argv[1]);

    for (i = 0; i < end; i++)
    {
        arg[i] = argv[1][i] - '0';
    }

    while (end - start > 2)
    {
        while (arg[end - 1] == 0)
        {
            printf("%d\n", base);
            end--;
        }

        while (arg[start] == 0)
        {
            start += 1;
        }
        base += 1;
        for (i = start + 1; i < end; i++)
        {
            for (j = i; j > start; j--)
            {
                if ((arg[j] -= arg[j - 1]) < 0)
                {
                    arg[j] += base;
                    arg[j - 1] -= 1;
                }
            }
        }
    }
    return 0;
}
```

EXERCICE 2

Compilez le fichier original « quality.c », puis exécutez-le avec quelques valeurs numériques. Repassez CPPCheck afin de compléter l'amélioration du code

Avec CPPCheck, nous avons quelques erreurs de style :

Fichier	Sévérité	Ligne	Résumé	Since date	Tag
quality.c					
qua...	erreur de style	8	MISRA 2.7 (Advi...	01/02/2024	
qua...	erreur de style	18	MISRA 12.1 (Ad...	01/02/2024	
qua...	erreur de style	11	MISRA 12.3 (Ad...	01/02/2024	
qua...	erreur de style	22	MISRA 17.7 (Re...	01/02/2024	
qua...	erreur de style	5	MISRA 21.6 (Re...	01/02/2024	

Voici donc le code corrigé :

```c
/*
 * Copyright (c) 2024 BLIGNY PIVON
 */

#include <string.h>

int main(char **argv)
{
    long arg[100];
    int i;
    int j;
    int base = 10;
    int start = 0;
    int end = strlen(argv[1]);

    for (i = 0; i < end; i++)
    {
        arg[i] = argv[1][i] - '0';
    }

    while ((end - start) > 2)
    {
        while (arg[end - 1] == 0)
        {
            (void)printf("%d\n", base);
            end--;
        }

        while (arg[start] == 0)
        {
            start += 1;
        }
        base += 1;
        for (i = start + 1; i < end; i++)
        {
            for (j = i; j > start; j--)
            {
                if ((arg[j] -= arg[j - 1]) < 0)
                {
                    arg[j] += base;
                    arg[j - 1] -= 1;
                }
            }
        }
    }
    return 0;
}
```