

Main.c

```
/**
 * @file main.c
 * @author Durand Thomas
 * @brief fonction main avec exemple d'utilisation des fonctions demande dans le TD
 * @version 0.1
 * @date 2019-11-12
 *
 * @copyright Copyright (c) 2019
 *
 */
#include <stdio.h>
#include <stdlib.h>
#include "tab1d.h"
#include "chaine.h"

#define N 20

int main(){

int tab1[N],tab2[N],tabRes[N];

for (int i = 0; i < N; i++){
tab1[i] = i;
tab2[i] = i;
tabRes[i] = 0;
}
inverseTab(tab1);
somme(tab1,tab2,tabRes);
afficheTab(tabRes);
char chaine[] = "coucou je m'appelle Thomas";

printf("longueur chaine: %d\n",longueurChaine(chaine));
afficheChaine(chaine);
supprimeChar(chaine);
afficheChaine(chaine);

return 0;
}
```

tab1D.c

```

/**
 * @file tab1d.c
 * @author Durand Thomas
 * @brief ensemble des fonctions de l'exercice 1
 * @version 0.1
 * @date 2019-11-12
 *
 * @copyright Copyright (c) 2019
 *
 */

```

```

#include <stdio.h>
#include <stdlib.h>

```

```

#define N 20

```

```

/**
 * @brief cherche un element dans le tableau et renvoie son id
 *
 * @param tab
 * @param num
 * @return int
 */
int chercheElement(int tab[N], int num){
    for (int i = 0; i < N; i++){
        if (tab[i] == num){
            return i;
        }
    }
    return -1;
}

```

```

/**
 * @brief inverse un tableau
 *
 * @param tab
 */
void inverseTab(int tab[N]){
    int temp;
    for (int i = 0; i < (N / 2); i++){
        temp = tab[i];
        tab[i] = tab[N-i-1];
        tab[N-i-1] = temp;
    }
}

```

```
}  
}
```

```
/**
```

```
 * @brief affiche un tableau
```

```
 *
```

```
 * @param tab
```

```
 */
```

```
void afficheTab(int tab[N]){
```

```
    printf("\n");
```

```
    for (int i = 0; i < N; i++){
```

```
        printf(" %d ",tab[i]);
```

```
    }
```

```
    printf("\n");
```

```
}
```

```
/**
```

```
 * @brief addition deux vecteur et met le resultat dans un troisieme
```

```
 *
```

```
 * @param tab1
```

```
 * @param tab2
```

```
 * @param tabRes
```

```
 */
```

```
void somme(int tab1[N],int tab2[N],int tabRes[N]){
```

```
    for (int i = 0; i < N; i++){
```

```
        tabRes[i] = tab1[i] + tab2[i];
```

```
    }
```

```
}
```

chaine.c

```
/**
```

```

* @file chaine.c
* @author Durand Thomas
* @brief ensemble des fonctions de l'exercice 2
* @version 0.1
* @date 2019-11-12
*
* @copyright Copyright (c) 2019
*
*/

#include <stdio.h>
#include <stdlib.h>

/**
* @brief calcule la longueur d'une string
*
* @param chaine
* @return int
*/
int longueurChaine(char chaine[]){
    int i = 0;
    while (chaine[i] != '\0'){
        i++;
    }
    return i;
}

/**
* @brief supprime les espaces et les tabulations d'une strig
*
* @param chaine
*/
void supprimeChar(char chaine[]){
    int i = 0;
    int j = 0;
    while (chaine[j] != '\0'){
        if (chaine[j] == ' ' || chaine[j] == 9){
            j++;
        }
        if (i != j){
            chaine[i] = chaine[j];
        }
        i++;
        j++;
    }
}

```

```
chaine[i] = '\0';
}

/**
 * @brief affiche une string
 *
 * @param chaine
 */
void afficheChaine(char chaine[]){
printf("%s",chaine);
printf("\n");
}
```