

Rattrapage de Programmation C

ING1 – GI

16 février 2022



-
- *Durée : 2 heures*
 - *Vous devez rédiger votre copie à l'aide d'un **stylo à encre** exclusivement.*
 - *Toutes vos affaires (sacs, vestes, trousse, etc.) doivent être placées à l'avant de la salle.*
 - *Aucun document n'est autorisé.*
 - *Aucune machine électronique ne doit se trouver sur vous ou à proximité, même éteinte.*
 - *Aucun déplacement n'est autorisé.*
 - ***Aucune question au professeur n'est autorisée.** Si vous pensez avoir détecté une erreur, continuez en expliquant les hypothèses que vous faites.*
 - *Aucun échange, de quelque nature que ce soit, n'est possible.*
 - *Le barème est donné à titre indicatif.*
-

Exercice 1 : Questions de cours (5 pts)

1. À quoi servent les séquences d'échappement `\t` et `\0` ? (0.5 pt)
2. Qu'est-ce que doxygen? (0.5 pt)
3. À quoi servent `$@` et `^` dans un Makefile? (0.5 pt)
4. Réalisez l'allocation mémoire d'une matrice d'entiers 5×5 . (1pt)
5. À quoi sert typedef? (0.5 pt)
6. Repérez et expliquez les erreurs se cachant dans ce code : (2 pts)

```
1  int main(int argc, char argv) {
2      int taille, retour;
3      srand(time(NULL));
4      printf("Veuillez entrer la taille du tableau : \n");
5      retour = scanf("%d", taille);
6      if (retour > 0) {
7          int vecteur[taille];
8          for (int *p = vecteur[0]; p < vecteur[taille]; p++) {
9              *p = rand()%10;
10         }
11     }
12     return 0 ;
13 }
```

Exercice 2 : Multiple de 3 ou non (5 pts)

1. Écrire un programme qui demande à l'utilisateur de saisir des entiers stockés dans un tableau dynamique dont la taille sera préalablement saisie par l'utilisateur. (1pt)
2. Écrire une fonction : `int compter_multiples3 (int* T , int n)`. La fonction prend en paramètres le tableau et le nombre d'éléments du tableau et retourne le nombre d'entiers multiples de 3. Appeler cette fonction pour afficher le nombre d'entiers multiples 3 et le nombre d'entiers qui ne sont pas multiples de 3, dans le tableau saisi. (2pts)
3. Écrire une procédure qui modifie le tableau de telle sorte que les nombres multiples de 3 soient tous placés avant les nombres qui ne le sont pas. (2pts)
 - L'ordre des éléments ne sera pas forcément conservé.
 - Les échanges successifs seront affichés. La procédure affichera aussi le tableau final séparé en 2 parties à l'aide du caractère *barre verticale* (`|`).

NB : le caractère *barre verticale* (`|`) n'a pas à être inséré dans le tableau.

L'affichage devra être conforme à l'affichage attendu suivant :

```
Combien d'entiers dans le tableau ? 10
3
4
6
2
1
9
12
2
2
3
5 nombres multiples de 3 et 5 nombres non multiples de 3
3 3 6 2 1 9 12 2 2 4
3 3 6 12 1 9 2 2 2 4
3 3 6 12 9 1 2 2 2 4
Tableau trie : multiples de 3 | non multiples de 3
3 3 6 12 9 | 1 2 2 2 4
```

Exercice 3 : Manipulations de chaînes (5 pts)

Toutes les questions de cet exercice doivent être réalisées sans les fonctions fournies par `string.h`

1. Écrire une fonction pour calculer la longueur d'une chaîne de caractères passé en paramètre. (1 pt)
2. Écrire une procédure pour copier une chaîne de caractères vers une autre (qu'on supposera déjà allouée). (1 pt)
3. Écrire une procédure qui remplace toutes les occurrences d'un caractère donné dans une chaîne de caractères par un autre caractère également passé en paramètre. (1pt)
4. Écrire une fonction pour trouver la première occurrence d'une sous-chaîne de caractères dans une chaîne. La fonction renverra un pointeur vers le début de la sous-chaîne trouvée ou NULL si elle n'est pas trouvée. (2 pts)

Exercice 4 : Album de musique (5 pts)

Un album est décrit par son titre, son groupe, son genre(classique, danse, jazz, pop, rap), sa date de sortie, sa durée, et son format (CD, MP3).

1. Écrire les énumération genre et format ainsi que la structure permettant de représenter le type album musical. (1.5pts)
2. Écrire une fonction qui prendra en entrée une bibliothèque de N albums et un genre de musique g, et retournera le nombre d'albums du genre g apparaissant dans la bibliothèque. (1pt)
3. Écrire une fonction qui prendra en paramètres une bibliothèque de N albums et un titre t, et permettant de retourner l'index du premier album dans la bibliothèque correspondant au titre t. La valeur N est retournée s'il n'existe aucun album intitulé par t dans la bibliothèque. (1pts)
4. Écrire la procédure sauvegarder qui permet d'écrire dans le fichier "artiste.txt" une bibliothèque d'albums passé en paramètre. (1.5pt)

A Annexes

A.1 Extrait de la page de man : `strcmp`

SYNOPSIS

```
#include <string.h>
int strcmp(const char *s1, const char *s2);
```

DESCRIPTION

The **strcmp()** function compares the two strings *s1* and *s2*. The locale is not taken into account. It returns an integer less than, equal to, or greater than zero if *s1* is found, respectively, to be less than, to match, or be greater than *s2*.

RETURN VALUE

The **strcmp()** function returns an integer less than, equal to, or greater than zero if *s1* is found, respectively, to be less than, to match, or be greater than *s2*.

A.2 Extrait de la page de man : fopen

SYNOPSIS

```
#include <stdio.h>
FILE *fopen(const char *pathname, const char *mode);
```

DESCRIPTION

The **fopen()** function opens the file whose name is the string pointed to by *pathname* and associates a stream with it. The argument *mode* points to a string beginning with one of the following sequences:

- r** Open text file for reading. The stream is positioned at the beginning of the file.
- r+** Open for reading and writing. The stream is positioned at the beginning of the file.
- w** Truncate file to zero length or create text file for writing. The stream is positioned at the beginning of the file.
- w+** Open for reading and writing. The file is created if it does not exist, otherwise it is truncated. The stream is positioned at the beginning of the file.
- a** Open for appending (writing at end of file). The file is created if it does not exist. The stream is positioned at the end of the file.
- a+** Open for reading and appending (writing at end of file). The file is created if it does not exist. Output is always appended to the end of the file.

RETURN VALUE

Upon successful completion **fopen()** return a FILE pointer. Otherwise, NULL is returned and *errno* is set to indicate the error.

A.3 Extrait de la page de man : fclose

SYNOPSIS

```
#include <stdio.h>
int fclose(FILE *stream);
```

DESCRIPTION

The **fclose()** function flushes the stream pointed to by *stream* (writing any buffered output data using **fflush(3)**) and closes the underlying file descriptor.

The behaviour of **fclose()** is undefined if the *stream* parameter is an illegal pointer, or is a descriptor already passed to a previous invocation of **fclose()**.

RETURN VALUE

Upon successful completion, 0 is returned. Otherwise, **EOF** is returned and *errno* is set to indicate the error. In either case, any further access (including another call to **fclose()**) to the *stream* results in undefined behavior.

A.4 Extrait de la page de man : printf, fprintf

SYNOPSIS

```
#include <stdio.h>
int printf(const char *format, ...);
int fprintf(FILE *stream, const char *format, ...);
```

DESCRIPTION

The functions in the **printf()** family produce output according to a *format* as described below. The function **printf()** write output to *stdout*, the standard output stream; **fprintf()** write output to the given output *stream*.

All of these functions write the output under the control of a *format* string that specifies how subsequent arguments (or arguments accessed via the variable-length argument facilities of **stdarg(3)**) are converted for output.

RETURN VALUE

Upon successful return, these functions return the number of characters printed (excluding the null byte used to end output to strings).

If an output error is encountered, a negative value is returned.