Introduction to programming (NF05A)
— exercices 2020-2021 —

Université de technologie de Troyes

Tutorial 5

Exercise 1. Files

1. Write a procedure/function to read and store M real vectors of size N (M and N will be entered by the user) in a file with a name given by the user.

2. Write a procedure/function to calculate the sum of the elements of the two vectors in the previous file and add the result vector to this file.

Exercise 2. Structures, pointers and files

The manager of a small library wishes to computerize the management of the loan of books. Each book is identified by an 8 characters rib (string), a title, the name of the main author, the name of the publisher, the year of publication. To these informations we must add the indication of the loan (book on loan or on the shelf) and the date of loan if necessary.

Here is a suggestion for the LibBook structure:

```
typedef enum {Loaned, Available} bookState;

typedef struct {
          char author[20] ;
          char title[100] ;
          char rib[9] ;
          int year ;
          char editor[20] ;
          bookState state;
          date loanDate ;
} LibBook ;
```

1. Write a procedure/function to enter information of a book.

2. Write a procedure/function to enter N books (N will be given by the user) and save them in a file with a file name entered by the user. The first line of the file contains the number of books stored. Each type of information of a book is stored in a line.

3. Write a procedure to read the contents of a library file using a the structure LibBook and dynamic memory allocation.

4. Write a function to search for information about a book. The name of the searched book will be given by the user. If the requested book is found, display this information.

5. Write a procedure to modify some information of a requested book (for example if a book is returned to the library, you have to modify loaned, loanDate,...). Save this information on the same source file.

Exercise 3. Files concatenation

Write a function `int concat(char* sResult, char* sFile1, char* sFile2)` which allows to concatenate the files whose names are sFile1 and sFile2 and put the result in the file whose name is sResult. The function must return 1 if no error is detected, 0 otherwise.

Additional exercises - To go further

Exercise 4. Files encryption.

Write a function `int encrypt(int d, char* source, char* dest)` that encrypt the conent of the file named `source` and write the result into the file named `dest` by replacing each character `c` in the file `source` with the value character `(c + d) modulo 256`. Test this function.

Exercise 5. Statistics on files.

Write a procedure `void stat_freq(char* filename)` which displays all the characters appearing in the file of name `filename` in a descending order according to frequency of appearance in the file. Test this procedure.