

Programming Project

Degree in Industrial Engineering

FREQUENTLY ANSWERED QUESTIONS

1. The provided I/O functions are invoked only once as shown in *main.c*. Read functions are used at the beginning of the program to load file data into main memory structures. Write functions are used at the end of the program to dump data in main memory structures to the disk.

Program start:

```
readUsersFromFile("users.txt", users, &nUsers);
readItemsFromFile("items.txt", items, &nItems);
```

Program end:

```
writeUsersToFile("users.txt", users, nUsers);
writeItemsToFile("items.txt", items, nItems);
```

2. The prototype of the login function is as follows:

```
/** @fn int login(struct User users[], int nUsers)
    @brief Log in the application
    @param[in] users Users array (must have been initialized)
    @param[in] nUsers Number of (valid) elements in users array
    @return Index in the array of the user in [0, MAX_USERS-1].
            MAX_USERS if the administrator user/password was used.
            -1 if wrong after three tries.

    @details The function reads a user login and a password,
             and searches the users array to find a coincidence
             (the administrator user/password is also considered)
             If there is no coincidence after three tries, the
             function returns -1; otherwise, it returns the
             position of the user in the array or a special value
             for the administrator user (MAX_USERS)
**/
int login(struct User users[], int nUsers);
```

The function is invoked as follows:

```
int user = 0; // anonymous: 0, registered: 1, administrator: 2
int cu_id; // id of current logged user (if registered user)
int cu_index; // index of current logged user in the user array
              // (if registered user)
...
```

```
cu_index = login(users, nUsers);
if(cu_index >= 0 && cu_index < MAX_USERS) {
```



```

    // registered user
    cu_id    = cu_index + 1;
    user     = 1;
} else if(cu_index == MAX_USERS) {
    // administrator
    cu_id = 0;
    user  = 2;
} else {
    // not a registered user
    cu_id = -1;
}

```

Notice that a user with identifier n is stored at the position $n-1$ of the array `users`. Analogously, the position k stores the user with id $k+1$. This can be directly extended to the `items` array. Be consistent in your program with regard to the use of indexes and identifiers.

3. Search functions must work when the actual number of elements of the arrays is 0. This can be achieved by testing the end condition at the beginning of the loop; e.g.:

```

int i, pos=-1;
for(i=0; i<nUsers && pos==-1; i++) {
    // test if found
}

int i=0, pos= 1;
do {
    // test if found and increase i
} while (i<nUsers && pos==1);

```

4. Passing string characters as parameters is equivalent to passing regular one-dimension arrays.
5. When adding new elements to an array, do check that the length of the array would not be exceeded.
6. Any function that can be reused must be reused.
7. The use of global variables is completely forbidden. If a function needs a value from any other function (and, in particular, from the main function), a parameter must be used.
8. Watch out the precedence of the unary operators `++` and `--` when using parameters passed by reference (the `*` indirection operator has less precedence than `++` and `--`):

```

// Assuming that N is passed by reference
(*N)++; // The value at memory address N is incremented
        // <--> (*N) = (*N) + 1;

*N++; // N is incremented, the value at memory address N+1

```



```
// is accessed (but nothing is done)
```

9. The showcase must print on the screen the identifier, name and price of each corresponding item. Use conveniently placeholder modifiers to set the width of the printed fields.

Check that it is possible to print the number of items specified in the project description. For example, it may happen that there is only 2 items in the system. In this case, do not try to print the 5 most recent items. Be especially careful if you are implementing the extended version of the project.

10. Test that it is possible to show more information about a given item after presenting a list in the extended version of the project. In particular, do not allow users to list detailed information of items that have not been displayed on the screen and they do not have access to.

To do so, you are recommended to create a boolean array `list[MAX_PRODUCTS]` that marks if the item $i=0, \dots, nItems$ has been shown in the last list.

