Exercises — Null terminated arrays

version #

Assistants C/Unix 2022 <assistants@tickets.assistants.epita.fr>

Copyright

This document is for internal use at EPITA (website) only.

Contents

*https://intra.assistants.epita.fr

1 Null terminated arrays

Files to submit:

- null_terminated_arrays/null_terminated_arrays.c
- null_terminated_arrays/null_terminated_arrays.h

Provided files:

- null_terminated_arrays/null_terminated_arrays.h

Authorized headers: You are only allowed to use the functions defined in the following headers:

- stddef.h
- assert.h
- err.h
- errno.h

1.1 Goal

In this exercise you will manipulate arrays whose end is marked by a `NULL` pointer.

Write the function `reverse_matrix`, which takes an array of string arrays and reverses the main array as well as all the nested arrays, you should not reverse the string itself.

```
void reverse_matrix(const char ***matrix);
```

`matrix` is a null-terminated array of null-terminated string arrays. The strings stored in each array are valid and null-terminated strings.

- `matrix` (`const char ***`) is ended by a `NULL` pointer : the array of arrays.
- `matrix[0]` (`const char **`) is ended by a `NULL` pointer : the array of strings.
- `matrix[0][0]` (`const char *`) is ended by a `\0` character: the string (array of chars).

This means that in order to find the size of an array, you have to find the `NULL` element that marks the end of the array.

1.2 Example

1.2.1 Input:

```
{
  { "1", "2", "3", NULL },
  { "4", "5", NULL },
  { "6", "7", "8", "9", NULL },
  NULL
}
```

1.2.2 Output:

```
{
  { "9", "8", "7", "6", NULL },
  { "5", "4", NULL },
  { "3", "2", "1", NULL },
  NULL
}
```

1.2.3 Remarks

- Notice how the `NULL` still marks the end of the array.
- None of the strings can be `NULL`, because that would be breaking the end marker.
- Do not reverse the string itself.

It is my job to make sure you do yours.