Exercises — String Replace

version **#7be580532266ed398481e31366afcc24b1950c2a**

Assistants C/Unix 2023 <assistants@tickets.assistants.epita.fr>

Copyright

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

Contents

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

File Tree

```
string_replace/
├── string_replace.c  (to submit)
└── string_replace.h  (to submit)
```

Authorized functions : You are only allowed to use the following functions

- malloc(3)
- calloc(3)

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

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

Compilation : Your code must compile with the following flags

- -std=c99 -pedantic -Werror -Wall -Wextra -Wvla

Main function : None

1 Goal

You have to implement a function that copies the string `str` to a newly allocated string while replacing each `c` character with the string `pattern`. The function returns the new string.

```
char *string_replace(char c, const char *str, const char *pattern);
```

- The `c` character cannot be `\0`.
- If `pattern` is `NULL`, every occurrence of the `c` character will be removed in the new string.
- If `str` is `NULL`, `string_replace` returns `NULL`.

2 Example

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

#include "string_replace.h"

void check(char c, const char *s, const char *p)
{
    char *res = string_replace(c, s, p);
```

(continues on next page)

(continued from previous page)

```
    printf("%s\n", res);
    free(res);
}

int main(void)
{
    check('o', "bobo", "");
    check('o', "bobo", "i");
    check('o', "bobo", "oo");

    return 0;
}
```

```
42sh$ gcc -Wall -Wextra -Werror -std=c99 -pedantic -o replace string_replace.c main.c
42sh$ ./replace | cat -e
bb$
bibi$
booboo$
```

The way is lit. The path is clear. We require only the strength to follow it.