



EXERCISES — Generic linked list

version #



Copyright

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

Copyright © 2021-2022 Assistants [<assistants@tickets.assistants.epita.fr>](mailto:assistants@tickets.assistants.epita.fr)

The use of this document must abide by the following rules:

- ▷ You downloaded it from the assistants' intranet.*
- ▷ This document is strictly personal and must **not** be passed onto some-one else.
- ▷ Non-compliance with these rules can lead to severe sanctions.

Contents

1	Generic linked list	3
1.1	Goal	3
1.2	list_prepend	3
1.3	list_length	4
1.4	list_destroy	4

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

1 Generic linked list

Files to submit:

- generic_void_list/list.c

Provided files:

- generic_void_list/list.h

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

- free(3)
- malloc(3)
- calloc(3)
- memcpy(3)

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

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

1.1 Goal

In this exercise, you will have to implement a generic linked list, along with its manipulation operations.

Be careful!

An empty list is represented by a `NULL` pointer.

1.2 list_prepend

- **Authorized functions:** malloc(3), memcpy(3)

```
struct list *list_prepend(struct list *list, const void *value,  
                          size_t data_size);
```

This function must insert a node containing `value` at the beginning of the list. It must return the new list, or `NULL` if an error occurred. You can use `memcpy(3)` to copy `value` into the data field of the list structure.

1.3 list_length

- **Authorized functions:** none.

```
size_t list_length(struct list *list);
```

This function returns the length of the list.

1.4 list_destroy

- **Authorized functions:** free(3)

```
void list_destroy(struct list *list);
```

This function releases all the memory used by list.

It is my job to make sure you do yours.