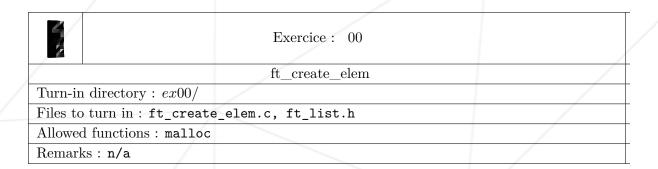
## Chapter III

Exercice 00: ft\_create\_elem

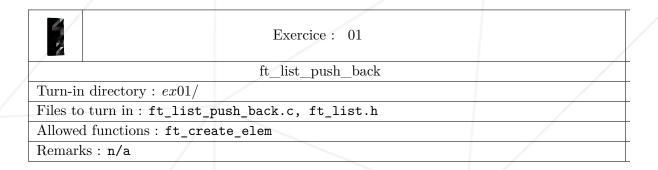


- $\bullet$  Create the function  ${\tt ft\_create\_elem}$  which creates a new element of  ${\tt t\_list}$  type.
- $\bullet$  It should assign  $\mathtt{data}$  to the given argument and  $\mathtt{next}$  to NULL.
- Here's how it should be prototyped :

t\_list \*ft\_create\_elem(void \*data);

#### Chapter IV

Exercice 01: ft\_list\_push\_back



- Create the function ft\_list\_push\_back which adds a new element of t\_list type at the end of the list.
- It should assign data to the given argument.
- If necessary, it'll update the pointer at the beginning of the list.
- Here's how it should be prototyped:

void ft\_list\_push\_back(t\_list \*\*begin\_list, void \*data);

### Chapter V

## Exercice 02: ft\_list\_push\_front

Exercice: 02	
ft_list_push_front	/
Turn-in directory: $ex02/$	
Files to turn in: ft_list_push_front.c, ft_list.h	
Allowed functions: ft_create_elem	
Remarks: n/a	

- Create the function ft\_list\_push\_front which adds a new element of type t\_list to the beginning of the list.
- It should assign data to the given argument.
- If necessary, it'll update the pointer at the beginning of the list.
- Here's how it should be prototyped :

void ft\_list\_push\_front(t\_list \*\*begin\_list, void \*data);

# Chapter VI

Exercice 03: ft\_list\_size

Exercice	: 03
ft_list_	size
Turn-in directory : $ex03/$	
Files to turn in: ft_list_size.c, ft_list.	h
Allowed functions: Nothing	
Remarks : n/a	

- Create the function ft\_list\_size which returns the number of elements in the list
- Here's how it should be prototyped :

int ft\_list\_size(t\_list \*begin\_list);

## Chapter VII

Exercice 04: ft\_list\_last

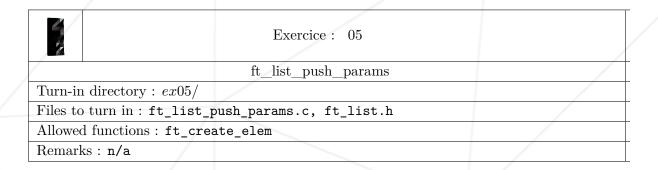
	Exercice: 04	
/	ft_list_last	
Turn-in directory : $e$	x04/	
Files to turn in : ft_	_list_last.c, ft_list.h	
Allowed functions: 1	Nothing	
Remarks : n/a		

- Create the function ft\_list\_last which returns the last element of the list.
- Here's how it should be prototyped :

t\_list \*ft\_list\_last(t\_list \*begin\_list);

### Chapter VIII

Exercice 05: ft\_list\_push\_params



- Create the function ft\_list\_push\_params which creates a new list that includes command-line arguments.
- The first argument should be at the end of the list.
- The first link's address in the list is returned.
- Here's how it should be prototyped:

t\_list \*ft\_list\_push\_params(int ac, char \*\*av);

### Chapter IX

Exercice 06: ft\_list\_clear

Exercice: 06

ft\_list\_clear

Turn-in directory: ex06/

Files to turn in: ft\_list\_clear.c, ft\_list.h

Allowed functions: free

Remarks: n/a

- $\bullet$  Create the function  $\verb|ft_list_clear| which clears all links from the list.$
- It'll then assign the list's pointer to null.
- Here's how it should be prototyped :

void ft\_list\_clear(t\_list \*\*begin\_list);

### Chapter X

Exercice 07: ft\_list\_at

Exercice: 07

ft\_list\_at

Turn-in directory: ex07/

Files to turn in: ft\_list\_at.c, ft\_list.h

Allowed functions: Nothing

Remarks: n/a

- $\bullet$  Create the function  ${\tt ft\_list\_at}$  which returns the Nth element of the list.
- In case of error, it should return a null pointer.
- Here's how it should be prototyped :

t\_list \*ft\_list\_at(t\_list \*begin\_list, unsigned int nbr);