SCALE FOR PROJECT LIBFT

You should evaluate 1 student in this team

Introduction

Please follow the rules below:

- Stay polite, courteous, respectful, and constructive during the evaluation process. The well-being of the community depends on it.

- Identify with the evaluated person or group any potential malfunctions in their work. Take the time to discuss and debate the identified issues.

- You must take into account that there may be slight differences in interpretation between the project instructions, its scope, and its features. Keep an open mind and grade as honestly as possible. The educational value is only valid if the peer evaluation is conducted seriously.

Guidelines

- Only grade what is contained in the cloned Git repository of the student or group.

- Ensure that the Git repository actually belongs to the student or group, that the project is indeed the one expected, and that "git clone" is used in an empty directory.

- Carefully verify that no aliases have been used to deceive you and ensure that you are evaluating the official submission.

- To avoid any surprises, check with the student or group for any potential scripts used to facilitate the evaluation, such as test or automation scripts.

- Do not blindly trust test programs! They could be incorrect.

- Check the Makefile: no wildcards, it should not re-link. If this is the case, the grade is 0. Use make, then make --debug to ensure this.

- If you have not completed the project you are about to evaluate, you must read the entire topic before starting the evaluation.

- Use the available flags to signal an empty submission, a non-functioning program, a Norme error, cheating... In these situations, the evaluation is terminated and the grade is 0, or -42 in the case of cheating. However, except in cases of cheating, you are encouraged to continue the discussion on the work submitted, even if it is incomplete, in order to identify mistakes not to be repeated in the future.

- Throughout the duration of the evaluation, no segfault or other unexpected, premature, or uncontrolled shutdown will be tolerated. In such cases, the final grade will be 0. Use the appropriate flag.

- You should never have to edit a file except for an existing configuration file. If you wish to modify a file, you must clearly state the reasons for the editing and be in agreement with the evaluated student before doing anything.

- You must also verify the absence of memory leaks. Any memory allocated on the heap must be properly freed before the end of the program's execution. You are entitled to use any tool available on the machine such as leaks, valgrind, or e_fence. In the case of memory leaks, check the appropriate flag.

Attachments

📄 subject.pdf

Mandatory Part

You must test the functions present in the libft, either with your own set of tests or with those of the person being evaluated.

Libc functions #1

Test that ALL these functions fulfill their role exactly as they should:

- ft_strlen
- ft_memcpy
- ft_memset
- ft_bzero
- ft_memmove
- ft_memcmp
- ft_memchr Does it work?

Yes	No

Libc functions #2

Test that ALL these functions fulfill their role exactly as they should:

- ft_isalpha
- ft_isdigit
- ft_isalnum
- ft_isascii
- ft_isprint Does it work?

Yes	No

Libc functions #3

Test that ALL these functions fulfill their role exactly as they should:

- ft_toupper
- ft_tolower
- ft_strchr
- ft_strrchr Does it work?

Yes	No

Libc functions #4

Test that ALL these functions fulfill their role exactly as they should:

- ft_strlcat
- ft_strlcpy
- ft_atoi
- ft_strnstr
- ft_strncmp
- ft_calloc
- ft_strdup Does it work?

⊘ Yes	✕ No

Additional functions #1

Test that ALL these functions fulfill their role exactly as they should:

- ft_substr
- ft_strjoin
- ft_putchar_fd
- ft_putstr_fd
- ft_putendl_fd
- ft_putnbr_fd Does it work?

⊘ Yes	✕ No

Additional functions #2

Test that ALL these functions fulfill their role exactly as they should:

- ft_strmapi
- ft_striteri
- ft_strtrim
- ft_split
- ft_itoa Does it work?

⊘ Yes	✕ No

Bonus Part

Bonuses will only be considered if the mandatory part is excellent. This means that the mandatory part must have been completed from start to finish, with perfect error handling even in the case of unexpected use. If all the mandatory points have not been awarded during this defense, no bonus points will be counted.

ft_lstnew

Does it work?

⊘ Yes	✕ No

ft_lstdelone

Does it work?

⊘ Yes	✕ No

ft_lstclear

Does it work?

⊘ Yes	✕ No

ft_lstadd_front

Does it work?

⊘ Yes	✕ No

ft_lstadd_back

Does it work?

⊘ Yes	✕ No

ft_lstsize

Does it work?

⊘ Yes	✕ No

ft_lstlast

Does it work?

⊘ Yes	✕ No

ft_lstiter

Does it work?

⊘ Yes	✕ No

ft_lstmap

Does it work?

⊘ Yes	✕ No

Ratings

Don't forget to check the flag corresponding to the defense

✔ Ok	★ Outstanding project

Empty work   📱 Incomplete work   🍖 Cheat   ♟ Crash   ▲ Concerning situation

💬 Can't support / explain code

Conclusion

<div style="background:#00b8b8;color:#fff;text-align:center;font-weight:bold;padding:8px;">Give this repository a star. ⭐</div>

Empty work   📱 Incomplete work   🍖 Cheat   ♟ Crash   ▲ Concerning situation

💬 Can't support / explain code

Conclusion

<div style="background:#00b8b8;color:#fff;text-align:center;font-weight:bold;padding:8px;">Give this repository a star. ⭐</div>