(https://profile.intra.42.fr/searches)

(https://profile.intra.42.fr)

SCALE FOR PROJECT LIBFT (/PROJECTS/42CURSUS-LIBFT)

You should evaluate 1 student in this team

★

Git repository

Introduction

Please comply with the following rules:

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

- Identify with the student or group whose work is evaluated the possible dysfunctions in their project. Take the time to discuss and debate the problems that may have been identified.

- You must consider that there might be some differences in how your peers might have understood the project's instructions and the scope of its functionalities. Always keep an open mind and grade them as honestly as possible. The pedagogy is useful only and only if the peer-evaluation is done seriously.

Guidelines

- Only grade the work that was turned in the Git repository of the evaluated student or group.

- Double-check that the Git repository belongs to the student(s). Ensure that the project is the one expected. Also, check that 'git clone' is used in an empty folder.

- Check carefully that no malicious aliases was used to fool you and make you evaluate something that is not the content of the official repository.

- To avoid any surprises and if applicable, review together any scripts used to facilitate the grading (scripts for testing or automation).

- Do not blindly trust any scripts. The tester could be wrong!

- Check the makefile: no wildcard, no relink, or the final grade is 0.

- Use make, then make --debug to be sure.

- If you have not completed the assignment you are going to evaluate, you have to read the entire subject prior to starting the evaluation process.

- Use the available flags to report an empty repository, a non-functioning program, a Norm error, cheating, and so forth.
In these cases, the evaluation process ends and the final grade is 0, or -42 in case of cheating. However, except for cheating, student are strongly encouraged to review together the work that was turned in, in order to identify any mistakes that shouldn't be repeated in the future.

- You should never have to edit any file except the configuration file if it exists. If you want to edit a file, take the time to explicit the reasons with the evaluated student and make sure both of you are okay with this.

- You must also verify the absence of memory leaks. Any memory allocated on the heap must be properly freed before the end of execution.
You are allowed to use any of the different tools available on the computer, such as leaks, valgrind, or e_fence. In case of memory leaks, tick the appropriate flag.

Attachments

subject.pdf (https://cdn.intra.42.fr/pdf/pdf/65489/en.subject.pdf)

Mandatory part

You must test each of the functions of the libft with your own tests.

Libc Functions #1

Test every function and make sure they work properly:

- ft_strlen
- ft_memcpy
- ft_memset
- ft_bzero
- ft_memmove
- ft_memcmp
- ft_memchr Is everything working?

✓ Yes ✕ No

Libc Functions #2

Test every function and make sure they work properly:

- ft_isalpha
- ft_isdigit
- ft_isalnum
- ft_isascii
- ft_isprint Is everything working?

✓ Yes ✕ No

LibC functions #3

Test every function and make sure they work properly:

- ft_toupper
- ft_tolower
- ft_strchr
- ft_strrchr Is everything working?

✓ Yes ✕ No

LibC functions #4

Test every function and make sure they work properly:

- ft_strlcat
- ft_strlcpy
- ft_atoi
- ft_strnstr
- ft_strncmp
- ft_calloc
- ft_strdup Is everything working?

✓ Yes ✕ No

Additional functions #1

Test every function and make sure they work properly:

- ft_substr
- ft_strjoin
- ft_putchar_fd
- ft_putstr_fd
- ft_putendl_fd

- ft_putnbr_fd Is everything working?

✓ Yes ✕ No

Additional functions #2

Test every function and make sure they work properly:

- ft_strmapi
- ft_striteri
- ft_strtrim
- ft_split
- ft_itoa Is everything working?

✓ Yes ✕ No

Bonus part

Evaluate the bonus part if, and only if, the mandatory part has been entirely and perfectly done, and the error management handles unexpected or bad usage. In case all the mandatory points were not passed during the defense, bonus points must be totally ignored.

ft_lstnew

Is everything working?

✓ Yes ✕ No

ft_lstdelone

Is everything working?

✓ Yes ✕ No

ft_lstclear

Is everything working?

✓ Yes ✕ No

ft_lstadd_front

Is everything working?

✓ Yes ✕ No

ft_lstadd_back

Is everything working?

✓ Yes ✕ No

ft_lstsize

Is everything working?

✓ Yes ✕ No

ft_lstlast

Is everything working?

✓ Yes ✕ No

ft_lstiter

Is everything working?

✓ Yes ✕ No

ft_lstmap

Is everything working?

✓ Yes ✕ No

Ratings

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

✔ Ok ★ Outstanding project

Empty work Incomplete work Invalid compilation Norme Cheat Crash

⚠ Concerning situation Leaks ⊘ Forbidden function

Conclusion

Leave a comment on this evaluation