Remember that the quality of the defenses, hence the quality of the of the school on the labor market depends on you. The remote defences during the Covid crisis allows more flexibility so you can progress into your curriculum, but also brings more risks of cheat, injustice, laziness, that will harm everyone's skills development. We do count on your maturity and wisdom during these remote defenses for the benefits of the entire community.

SCALE FOR PROJECT DR QUINE (/PROJECTS/DR-QUINE)

You should evaluate 1 student in this team

★

Git repository

`git@vogsphere.msk.21-school.ru:vogsphere/intra-uuid-5860e6e`

Introduction

Please respect the following rules:

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

- Identify with the person (or the group) graded the eventual dysfunctions of the work. Take the time to discuss and debate the problems you have identified.

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

Guidelines

- Only grade the work that is in the student or group's GiT repository.

- Double-check that the GiT repository belongs to the student or the group. Ensure that the work is for the relevant project

and 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 other than the content of the official repository.

- To avoid any surprises, carefully check that both the correcting and the corrected students have reviewed the possible scripts used to facilitate the grading.

- If the correcting student has not completed that particular project yet, it is mandatory for this student to read the entire subject prior to starting the defence.

- Use the flags available on this scale to signal an empty repository, non-functioning program, cheating etc. In these cases, the grading is over and the final grade is 0 (or -42 in case of cheating). However, with the exception of cheating, you are encouraged to continue to discuss your work (even if you have not finished it) in order to identify any issues that may have caused this failure and avoid repeating the same mistake in the future.

Attachments

Subject (https://cdn.intra.42.fr/pdf/pdf/5625/dr_quine.en.pdf)

Preliminaries

Preliminaries

Preliminaries

First, check that the following is true:

- Something has been uploaded to the GiT repository.

- There are valid Makefiles with the usual rules where appropriate.

- If any source file is read (read(), fread(), etc...) in any of the source files, this is considered cheating. In this case, select the CHEAT flag.

- The use of argc/argv is strictly forbidden. If used, select the CHEAT flag.

If any of the above is not true, the defence ends. You are still, however, encouraged to continue to discuss the project.

Yes No

Colleen

Colleen_start

Examine the source code. Make sure it follows the following restrictions:

- Two comments as specified in the subject, including comment within a function.

- A function in addition to main().

- A call to that function from main().

☑ Yes ☒ No

Compilation & checking

Run make.

Once compiled (verify the executable is named correctly), run the following command:
`./Colleen > tmp_Colleen; diff tmp_Colleen Colleen.c`

If there were any errors (diff output), the correction ends here.

☑ Yes ☒ No

Grace

Grace_start

Examine the source code. Make sure it follows the following restrictions:

- A comment as specified in the subject.

- No direct functions, (main() included).

- Three defines.

☑ Yes ☒ No

Compilation & checking

Run make.

Once compiled (verify the executable is named correctly), run the following command:
`rm Grace_kid.c; ./Grace ; diff Grace_kid.c Grace.c`

If there were any errors (diff output), the correction ends here.

☑ Yes ☒ No

Sully

Sully_start

Examine the source code. Make sure it follows the following restrictions:

- An integer initialized to 5.

☑ Yes ☒ No

Compilation & checking

Run make.

Once compiled (verify the executable is named correctly), run the following command:
`mkdir -p tmp; cp Sully tmp/; cd tmp/; ./Sully ; ls -al | grep Sully | wc -l`

The number displayed on the standard output should be 13.
Run the following command:
`diff ../Sully.c Sully_5.c ; diff ../Sully.c Sully_4.c ; diff Sully_5.c Sully_0.c`

The only difference must be the integer set initially, which is decremented at each step.

Verify that the program does not use arguments for main(). If it does, select the
CHEAT flag. The correction ends here.

Verify that with an initial integer of -1, the program should not do anything.
If there are any errors, the correction ends here.

☑ Yes ☒ No

Colleen ASM

Colleen_start

Examine the source code. Make sure it follows the following restrictions:

- Two comments as specified in the subject, including comment within a function.

- A function/routine in addition to the main.

- A call to that function/routine from the main.

☑ Yes ☒ No

Compilation & checking

Run make.

Once compiled (verify the executable is named correctly), run the following command:
`./Colleen > tmp_Colleen; diff tmp_Colleen Colleen.s`

If there were any errors (diff output), the correction ends here.

☑ Yes ☒ No

Grace ASM

Grace_start

Examine the source code. Make sure it follows the following restrictions:

- A comment as specified in the subject.

- No direct functions/routines, (main included).

- Three macros.

☑ Yes ☒ No

Compilation & checking

Run make.

Once compiled (verify the executable is named correctly), run the following command:
`rm Grace_kid.s; ./Grace ; diff Grace_kid.s Grace.s`

If there were any errors (diff output), the correction ends here.

☑ Yes ☒ No

Sully ASM

Sully_start

Examine the source code. Make sure it follows the following restrictions:

- A(n) integer/register initialized to 5.

☑ Yes ☒ No

Compilation & checking

Run make.

Once compiled (verify the executable is named correctly), run the following command:
`mkdir -p tmp; cp Sully tmp/; cd tmp/; ./Sully ; ls -al | grep Sully | wc -l`

The number displayed on the standard output should be 13.
Run the following command:
`diff ../Sully.s Sully_5.s ; diff ../Sully.s Sully_4.s ; diff Sully_5.s Sully_0.s`

The only difference must be the integer/register set initially, which is decremented at each step.

Verify that the program does not use arguments for the main. If it does, select the
CHEAT flag. The correction ends here.

Verify that with an initial integer/register with a value of -1, the program should not do anything.
If there are any errors, the correction ends here.

☑ Yes ☒ No

Dr_Quine bonus

Only grade the bonus section if ALL previous cases are PERFECT.

Bonus

You may only count the bonus if all three quines were recoded in another language
and all pass the test cases specified above.

☑ Yes ☒ No

Ratings

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

✔ Ok

★ Outstanding project

Empty work | Incomplete work | No author file | Invalid compilation | Cheat

Crash | Forbidden function

Conclusion

Leave a comment on this evaluation

Пир объяснил все предельно ясно,

Finish evaluation