

SCALE FOR PROJECT COMPUTORV2 (/PROJECTS/COMPUTORV2)

Introduction

We ask you for the good progress of this evaluation to respect the following rules:

- Be courteous, polite, respectful and constructive in all situations during this exchange. The bond of trust between the community 42 and you depends on it.
- Highlight to the person (or group) noted the possible malfunctions of work done, and take the time to discuss and discuss.
- Accept that there may sometimes be differences of interpretation on the subject's requests or the scope of the features. Stay open-minded about the other's vision (is he or she right or wrong?), And write down the most honestly possible. The pedagogy of 42 only makes sense if the peer-evaluation is done seriously.

Guidelines

- You should only evaluate what is on the student / group's rendering GiT repository .
- Make sure that the GiT repository is the one corresponding to the student or group and the project.
- Meticulously verify that no malicious alias has been used to mislead you and have you evaluate anything other than the content of the official repository.


- Any meaningful script facilitating the evaluation provided by one of the two parties must be rigorously checked by the other party to avoid unpleasant surprises.


- If the correcting student has not yet done this project, it is mandatory for this student to read the subject in full before starting this defense.

- Use the flags available on this scale to signal an empty rendering, non-functional, a standard fault, a cheat case, etc. In this case, the evaluation is completed and the final grade is 0 (or -42 in the special case of cheating). However, except cheating, you are encouraged to continue to exchange around the work done (or not done precisely) to identify the problems that led to this situation and avoid them for the next rendering.

- Check the code carefully to see if any libraries that facilitate the calculation and the parsing have been used.

Attachments

 Subject (<https://cdn.intra.42.fr/pdf/pdf/5012/computor.fr.pdf>)

 Subject (<https://cdn.intra.42.fr/pdf/pdf/5668/computorv2.en.pdf>)

Preliminary part

In this part it is just a matter of verifying that the answer key does not use anything forbidden that can facilitate the realization of the project, whether with a complex type in the language used or otherwise.

preliminaries

Once the repository has been cloned, ask your answer to set up the work environment to execute the rendering. You will also take advantage of this to check that there is code to manage the different types of variables requested:

- Natural integers
- Rational numbers - Complex numbers (with rational coefficients)
- Matrices
- Polynomial equations of degree less than or equal to 2

Check also that the program compiles well and / or runs well.

Throughout this correction, the program will NEVER leave impromptu (Segfault, misinterpretation ...).
If one of these steps is wrong, the project is worth 0 and you can stop the correction.

 Yes

 No.

Usual checks

Ask the answer key to explain how it handles parsing, the different types (complex, matrices). If the latter uses a library that facilitates only one of these last points, the correction stops, the project is worth 0 and you can stop the correction.

 Yes

 No.

Part assignment

In this part we will test all the behaviors related to the assignment of a variable or a function. You are invited to test extensively, the correction only brings you the idea of tests.

Basic error test

Here we will test basic errors such as `x == 2`,
or rollfacs on the keyboard, nonsense like
`x = 23edd23 - + - +`

 Yes

 No.

Semi-advanced error test

There, it will be more vicious like for example `= 2`
or even `3 = 4` or even `x = g` while `g` is not defined!
Test tendentious syntaxes like `f (x = 2, or x = [[4,2]`.

 Yes

 No.

Advanced error test

Try all the wackiest cases that you can imagine, such as `x = - 2`, `f (x) = x * 2` then `t = f (x)` (which is not possible), `i = 2` (knowing it is forbidden to let the user assign the variable `i`). Do not hesitate to test what goes through your head.

☒ Yes

☐ No.

Elementary valid test

On future tests, use this "variableName =?" to know the value assigned to the variable in the context of the program. For example if you enter $x = 2$, you can do $x = ?$ and you are supposed to see 2 at the line in the program interface. In this part it is a question of testing operations of the form $x = 2$, $y = 4i$, $z = [[2, 3]; [3, 5]]$.

☒ Yes

☐ No.

Valid semi-advanced test

There we will test the function assignment and inter-variable. Feel free to play on the spaces, tabs that need to be managed. Test $x = 2$ then $y = x$, then $y = ?$ Also test $x = 2$ then $x = 5$ then $x = ?$, If x is not worth 5 it's 0. You can try to do the same with matrices or imaginary numbers, like $A = [[2, 3]]$, then $B = A$, if $B = ?$ do not display A , it's 0.

☒ Yes

☐ No.

Advanced valid test

For this question, it is assignments mixing many elements that we will test. Start with $x = 2$, $y = x * [[4, 2]]$, $f(z) = z * y$, if $f(z) = ?$ do not display $z * [[8.4]]$, it is 0. Also test $x = 2$ then $f(x) = x * 5$, if $f(x) = ?$ do not display 10 (or something similar like $2 * 5$ for example), it's 0 too. Do not hesitate to test everything you can imagine at the level of an assignment, mixing all types, from matrices to imaginary ect, as long as it has a mathematical meaning.

☒ Yes

☐ No.

Calculating part

In this part we will test all the behaviors related to the calculation as well as the evaluation of a function. You are invited to test extensively, the correction only brings you the idea of tests.

Elementary valid test

Here you are led to test very simple calculations like $2 + 2 = ?$, $3 * 4 = ?$, $X = 2$ then $x + 2 = ?$. In the same style you can try the division by 0 as with $2/0 = ?$, Also test the management of float as with $1.5 + 1 = ?$

☒ Yes

☐ No.

Valid semi-advanced test

Here it will be a little more complex calculations, for example $x = 2 * i$ then $x^2 = ?$, If the result is not $-2i$ it is 0. Also test a matrix multiplication, like $A = \begin{bmatrix} 2 & 3 \\ 3 & 4 \end{bmatrix}$ then $B = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ then $A * B = ?$ You must see the matrix A to be displayed, otherwise it is 0. Also test inputs of the style $f(x) = x + 2$, $p = 4$, $f(p) = ?$ Same, if the result is not 6, it's 0.

✓ Yes

✗ No.

Advanced valid test

Test quite complex calculations like $4 - 3 - (2 * 3)^2 * (2 - 4) + 4 = ?$ for example, or $f(x) = 2 * (x + 3 * (x - 4))$ then $p = 2$, then $f(3) - f(p) + 2 = ?$ and the result is supposed to be 10. You can mix the complexes with the functions like $f(x) = 2 * x * i$ then $f(2) = ?$ (the expected result is $4i$). Same with the dice, feel free to try, use the program as if you were using a calculator.

✓ Yes

✗ No.

bonus

Reminder: If at one time or another, the program does not react correctly (bus error, segfault, etc.), the defense is finished and the note is 0. Remember to use the corresponding flags. This instruction is active from one end to the other of the defense. Bonuses should only be assessed if and only if the mandatory game is PERFECT. By PERFECT, we obviously hear that it is fully realized, it is not possible to put his behavior in default, even in case of error, as vicious as it is, misuse, etc. Concretely, this means that if the mandatory part did not obtain ALL the points during this defense, the bonuses must be entirely IGNORED.

bonus

It's up to you to let yourself be guided by your answers on the implemented bonuses. Bonus scoring is at your discretion.

Rate it from 0 (failed) through 5 (excellent)



ratings

Do not forget to check the flag

✓ Okay

★ Outstanding project

📄 Empty work

📄 Incomplete work

💬 No author file

💻 Invalid compilation

📄 Standard

📄 cheat

💻 Crash

👤 Incomplete group

🚫 Forbidden function

Conclusion

Leave a comment on this evaluation

Preview !!!

General term of use of the website
(<https://signin.intra.42.fr/legal/terms/6>)

Privacy policy
(<https://signin.intra.42.fr/legal/terms/5>)

Legal notices
(<https://signin.intra.42.fr/legal/terms/3>)

Declaration on the use of cookies
(<https://signin.intra.42.fr/legal/terms/2>)

Terms of use for video monitoring
(<https://signin.intra.42.fr/legal/terms/1>)

f
(<https://signin.intra.42.fr/legal/terms/4>)