�51

mgrimald

㊅

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

Scale for Project (https://profile.intra.42.fr/searches)

ComputorV2 (/ projects / computorv2)

(https://signin.intra.42.fr/users/sign_ou

you okay shoulds 1 student in this team

㊳

Git repository

vogsphere@vogsphere.42.fr: intra / 2017 / activities / computorv2 /

aiwanesk 謫 Introduction

We ask for smooth running of the evaluation of the following rules:

- Be courteous, polite, respectful and constructive in all situations in this exchange the bond of trust between the community 42 and you depend on(..

- highlight from the person or group) noted any malfunctions rendering work, and take time to discuss and debate.

- Accept that there may sometimes differences of interpretation on the applications of the subject or scope of features. Stay open minded to the vision of the other (is he or she right or wrong?), and write as honestly as possible. Pedagogy 42 makes sense only if the peer-evaluation is done seriously.

Guidelines

- You should evaluate what is on the GiT deposit of rendering the student (s) or group.

- Take care to ensure that the deposit GiT is the one corresponding to the student (s) or group, and the project.

- Check carefully that no malicious alias was used to confuse you and make you evaluate anything other than the content of the official repository.

- Any script sensible facilitate the assessment provided by one of the two parties must be rigorously checked by the other party to avoid unpleasant surprises.

- If the student (s) corrector / proofreader has not done this project, it is mandatory for this (you) student (e) to read the whole subject before beginning the defense.

- Use the available flags on this scale for a report rendered void, inoperative, a standard of fault, a case of cheating, etc. In this case, the assessment is completed and the final score is 0 (or -42 in the special case of cheating). However, except in cases of cheating, you are encouraged to continue to exchange around the work done (or not done correctly) to identify the problems that led to this situation and to avoid the next rendering.

- Carefully check the code to see if any bookstore facilitating the calculation and parsing have been used.

Attachments

疋 Topic (https://cdn.intra.42.fr/pdf/pdf/1006/computor.fr.pdf)

Sections

preliminarypart

in this part it is just to verify that the corrected uses nothing forbidden can facilitate his realization of the project, be it with a complex type in the language or else .

Preliminaries

Once the cloned deposit, ask your corrected to set up the working environment to execute his report. You will also enjoy in order to verify that there are many code to manage the different types of variables requested are: - 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 fine and / or execute well. Throughout this correction, the program will NEVER leave impromptu (Segfault, misinterpretation ...). If any of these steps is false, the project is 0 and you can stop the correction.

usual checks

Ask thecorrected to explain how he handles the parsing, the different types (complex matrices). If it uses a bookstore facilitating one of these latter points, the correction is stopped, the project is 0 and you can stop the correction.

Part assignment

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

Elementary error test

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

semi-advanced error checking

there, it'll be more vicious eg = 2 or even 3 = 4 or x = g where g is not defined! Test syntaxes tendentious as f (x = 2 or x = [[4.2].

Test erroradvanced

Tryall the most wacky event you can imagine, such as x = --2, f (x) = x * 2 and t = f (x) (which is not possible), i = 2 (knowing that it is forbidden to let the user set the variable i). do not hesitate to test what happens to you head.

elementary valid test

on the next test, use this "variableName =?" to find out 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 2 are supposed to see the line in the program interface in this part it comes to test the form of operations x = 2, y = 4i = z. [[2,3] [3,5]] .

Test valid semi-advanced

Here we will test the function assignment and inter-variable. Do not hesitate to play on spaces, tabs that must be managed. Test x = 2 then y = x and y =?. try as x = 2 and x = 5 then x = if x is not worth 5 is 0. You can try to do the same with matrices or imaginary numbers, like A = [[2.3]], then A = B, if B =? does not display A is 0.

休 ⇒ Yes No

Yes 休 ⇒No

休Yes No ⇒

休 ⇒ Yes No

Yes 休 ⇒No

休Yes No ⇒

㈱ ⇒ Yes No

valid advanced Test

for that matter, it's assignments combining many elements that will be tested. Start with x = 2, y = x * [[4,2]], f (z) = z * y, where f (z) =? does not display z * [[8,4]], it is 0. Also test 2 then x = f (x) = x * 5, if f (x) =? not displayed 10 (or something similar as 2 * 5, for example) is also 0. Really do not hesitate to test everything you can imagine at a meeting, combining all types of matrices imaginary ect, as AC to a mathematical sense.

Yes No ㈱ ⇒

computational Part

In this part we will test all behaviors related to the calculation and the evaluation of a function. You are invited to test extensively, the correction will take you that the idea of testing.

Elementary valid test

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

Yes No ㈱ ⇒

valid semi-advanced test

Here gonna be a little more complex calculations, such as x = 2 * i and x ^ 2 =, if the result is not -2i is 0. Test as a matrix multiplication as a = [[2,3] [3,4]] and B = [[1,0] [0,1]] and a ** B =? You should see the A matrix display, otherwise it is 0. Also test inputs style f (x) = x + 2 = 4 p, f (p) =? Same if the result is not 6 is 0.

㈱ Yes No⇒

advancedvalid test

Test rather complex calculations as 4 - 3 - (2 * 3) ^ 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 them with complex functions such as f (x) = 2 * x * i and f (2) =? (the expected result is 4i). Same with the dice, feel free to try, use the program as if using a calculator.

Yes No㈱ ⇒

Bonus

Reminder: If at one time or another, the program does not respond correctly (bus error, segfault, etc ..), the thesis is completed and the score is 0. Consider using the corresponding flags. This setpoint is active from one end to the other of the defense. Bonuses should only be evaluated if and only if the compulsory part is PERFECT. By PERFECT, it obviously means that it is fully implemented, it is not possible to its default behavior, even on error, as vicious as it is, misuse, etc. In practice this means that if the compulsory part did not get ALL the points during this thesis, the bonuses must be fully IGNORED.

Bonus

up to you to be guided by your corrected on the implemented bonus. The scoring bonus is at your free discretion.

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

3

Conclusion

Leave a comment on this patch

* (required) How to

Finish correcting