Scale for project computorv2 (https://projects.intra.42.fr/projects/computorv2)

You should correct 1 student in this team

Git repository

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

Introduction

We ask you for the proper conduct 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-assessment is done seriously.

Guidelines

- You should evaluate only what is on
the student's or group's rendering GiT repository .

- Make sure that the GiT repository is the one
corresponding to the student or group and the project.

- Meticulously check that no malicious alias has been used
to mislead you and make you evaluate anything other than
the contents 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 cases , 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 parsing have been used.

Attachments

Subject (https://cdn.intra.42.fr/pdf/pdf/1207/computorv2.en.pdf) Subject (https://cdn.intra.42.fr/pdf/pdf/1006/computor.fr.pdf)

sections

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 its rendering.
You will also take advantage of this to check that there is code
to manage the different types of requested variables, namely:
- 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 should 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 rollfaces 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 you can imagine, such as x = - 2, f (x) = x * 2 and 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

Sur les prochains tests, utilisez ceci "nomDeLaVariable = ?" pour savoir la valeur attribuée à la variable dans le contexte du programme. Par exemple si vous entrez x = 2, vous pouvez faire x = ? et vous êtes censés voir 2 à la ligne dans l'interface du programme. Dans cette partie il est question de tester des opérations de la forme x = 2, y = 4i, z = [[2,3];[3,5]].

☐ Yes ☐ No

Test valide semi-avancé

Là on va tester l'assignation de fonction et inter-variable. N'hésitez pas à jouer sur les espaces, tabulations qui doivent être gérés. Testez x = 2 puis y =x, puis y = ?. Testez aussi x = 2 puis x=5 puis x = ?, si x ne vaut pas 5 c'est 0. Vous pouvez essayer de faire pareil avec des matrices ou des nombres imaginaires, comme A = [[2,3]], puis B = A, si B = ? n'affiche pas A, c'est 0.

☐ Yes ☐ No

Test valide avancé

Pour cette question, c'est des assignations mêlant beaucoup d'élements que l'on va tester. Commencez avec x = 2, y = x * [[4,2]], f(z) = z * y, si f(z) = ? n'affiche pas z * [[8,4]], c'est 0. Testez aussi x = 2 puis f(x) = x * 5, si f(x) = ? n'affiche pas 10 (ou quelque chose de similaire comme 2 * 5 par exemple), c'est 0 aussi. N'hésitez vraiment pas à tester tout ce que vous pouvez imaginer au niveau d'une assignation, mêlant tous les types, de matrices à imaginaires ect, tant que ca à un sens mathématique.

☐ Yes ☐ No

Partie calculatoire

Dans cette partie on va tester tous les comportements liés au calcul ainsi qu'à l'évaluation d'une fonction. Vous êtes invités à tester longuement, la correction ne vous amène que l'idee des tests.

Test valide élémentaire

Ici vous êtes amenés à tester des calculs très simples comme 2 + 2 = ?, 3 * 4 = ?, x = 2 puis x + 2 = ?. Dans le même style vous pouvez essayer la division par 0 comme avec 2 / 0 = ?, testez aussi la gestion de float comme avec 1.5 + 1 = ?

☐ Yes ☐ No

Test valide semi-avancé

Ici ca va être des calculs un peu plus complexes, comme par exemple x = 2 * i puis x ^ 2 = ?, si le résultat n'est pas -2i c'est 0. Testez aussi une multiplication matricielle, comme A = [[2,3];[3,4]] puis B = [[1,0];[0,1]] puis A ** B = ? Vous devez voir la matrice A s'afficher, sinon c'est 0. Testez aussi des inputs du style f(x) = x + 2, p = 4, f(p) = ? Pareil, si le résultat n'est pas 6, c'est 0.

☐ Yes ☐ No

Test valide avancé

Testez des calculs assez complexes comme 4 - 3 - (2 * 3) ^ 2 * (2 - 4) + 4 = ? par exemple ou encore f(x) = 2*(x + 3*(x - 4)) puis p = 2, puis f(3) - f(p) + 2= ? et le resultat est censé être 10. Vous pouvez mêler les complexes avec les fonctions comme f(x) = 2 *x * i puis f(2) = ? (le résultat attendu est 4i). Pareil avec les matrices, n'hésitez pas à essayer, utilisez le programme comme si vous utilisiez une calculatrice.

☐ Yes ☐ No

Bonus

Rappel : Si à un moment ou un autre, le programme ne réagit pas correctement (bus error, segfault, etc..), la soutenance est terminée et la note est 0. Pensez à utiliser les flags correspondants. Cette consigne est active d'un bout à l'autre de la soutenance. Les bonus ne doivent être évalués que si et seulement si la partie obligatoire est PARFAITE. Par PARFAITE, on entend bien évidemment qu'elle est entièrement réalisée, qu'il n'est pas possible de mettre son comportement en défaut, même en cas d'erreur, aussi vicieuse soit-elle, de mauvaise utilisation, etc. Concrètement, cela signifie que si la partie obligatoire n'a pas obtenu TOUS les points pendant cette soutenance, les bonus doivent être intégralement IGNORÉS.

Bonus

It's up to you to 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

☐ OK ☐ Outstanding project

☐ Empty work ☐ Incomplete work ☐ No author file W Invalid compilation ☐ Standard ☐ cheat d Crash ☐ Incomplete group

Conclusion

Leave a comment on this evaluation

*** (required) How**

Finish evaluation