



mgrimald



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

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

(https://signin.intra.42.fr/us

Scale for project ComputorV2 (/projects/computorv2)

You should correct 1 student in this team



### Git repository

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

## Introduction

Nous vous demandons pour le bon déroulement de cette évaluation de respecter les règles suivantes :

- Restez courtois, polis, respectueux et constructifs en toutes situations lors de cet échange. Le lien de confiance entre la communauté 42 et vous en dépend.
- Mettez en évidence auprès de la personne (ou du groupe) notée les dysfonctionnements éventuels du travail rendu, et prenez le temps d'en discuter et d'en débattre.
- Acceptez qu'il puisse y avoir parfois des différences d'interprétation sur les demandes du sujet ou l'étendue des fonctionnalités. Restez ouvert d'esprit face à la vision de l'autre (a-t-il ou elle raison ou tort ?), et notez le plus honnêtement possible. La pédagogie de 42 n'a de sens que si la peer-évaluation est faite sérieusement.

## Guidelines

- Vous ne devez évaluer que ce qui se trouve sur le dépôt GiT de rendu de l'étudiant(e) ou du groupe.
- Prenez soin de vérifier que le dépôt GiT est bien celui correspondant à l'étudiant(e) ou au groupe, et au projet.
- Vérifiez méticuleusement qu'aucun alias malicieux n'a été utilisé pour vous induire en erreur et vous faire évaluer autre chose que le contenu du dépôt officiel.
- Tout script sensé faciliter l'évaluation fourni par l'un des deux partis doit être rigoureusement vérifié par l'autre parti pour éviter des mauvaises surprises.
- Si l'étudiant(e) correcteur/correctrice n'a pas encore fait ce projet, il est obligatoire pour cet(te) étudiant(e) de lire le sujet en entier avant de commencer cette soutenance.
- Utilisez les flags disponibles sur ce barème pour signaler un rendu vide, non fonctionnel, une faute de norme, un cas de triche, etc. Dans ce cas, l'évaluation est terminée et la note finale est 0 (ou -42 dans le cas spécial de la triche). Toutefois, hors cas de triche, vous êtes encouragés à continuer d'échanger autour du travail effectué (ou non effectué justement) pour identifier les problèmes ayant entraîné cette situation et les éviter pour le prochain rendu.
- Vérifiez soigneusement le code pour voir si aucune librairie facilitant le calcul ainsi que le parsing n'ont été utilisées.

## Attachments

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

## Sections

### Partie préliminaire

Dans cette partie il s'agit juste de vérifier que le corrigé n'utilise rien d'interdit pouvant lui faciliter la réalisation du projet, que ce soit avec un type complexe dans le langage utilisé ou autre.

#### Préliminaires

Une fois le depot cloné, demandez a votre corrigé de mettre en place l'environnement de travail pour executer son rendu.

Vous en profiterez aussi pour verifier qu'il existe bien du code pour gerer les differents types de variables demandées à savoir:

- Nombres entiers naturels
- Nombres rationnels
- Nombres complexes (à coefficients rationnels)
- Matrices
- Equations polynomiales de degré inférieur ou égal a 2

Vérifiez aussi que le programme compile bien et/ou s'exécute bien.  
Tout au long de cette correction, le programme ne devra JAMAIS quitter de manière impromptue (Segfault, erreur d'interprétation...).  
Si une de ces étapes est fausse, le projet vaut 0 et vous pouvez arrêter la correction.

✓ Yes

✗ No

#### Vérifications usuelles

Demandez au corrigé d'expliquer comment il gère le parsing, les différents types (complexes, matrices). Si ce dernier utilise une librairie facilitant un seul de ces derniers points, la correction s'arrête, le projet vaut 0 et vous pouvez arrêter la correction.

✓ Yes

✗ No

### Partie assignation

Dans cette partie on va tester tous les comportements liés à l'assignation d'une variable ou d'une fonction. Vous êtes invités à tester longuement, la correction ne vous amène que l'idée des tests.

#### Test erreur élémentaire

Ici on va tester des erreurs de base comme par exemple `x == 2`, ou encore des rollfacs sur le clavier, des non-sens comme `x = 23edd23-+-+`

✓ Yes

✗ No

#### Test erreur semi-avancé

Là, ça sera plus vicieux comme par exemple `= 2` ou même `3 = 4` ou encore `x = g` alors que `g` n'est pas défini !  
Testez des syntaxes tendancieuses comme `f(x = 2)`, ou `x = [[4,2]]`.

✓ Yes

✗ No

#### Test erreur avancé

Essayez tous les cas les plus farfelus que vous pouvez imaginer, comme par exemple `x = --2`, `f(x) = x * 2` puis `t = f(x)` (qui n'est pas possible), `i = 2` (sachant qu'il est interdit de laisser l'utilisateur assigner la variable `i`). N'hésitez pas à tester ce qui vous passe par la tête.

✓ Yes

✗ No

#### Test valide élémentaire

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 assignments mêlant beaucoup d'éléments 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 ça a 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'idée 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 ça 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 résultat 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**

À vous de vous laisser guider par votre corrigé sur les bonus implémentés. La notation des bonus est à votre libre discrétion.

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

3

**Conclusion****Leave a comment on this correction****\*(required) Comment**


Finish correction

