



(<https://profile.intrav2.42.fr>)

## (<https://profile.intrav2.42.fr/searches>) Scale for project Get\_Next\_Line (/projects/get\_next\_line)

### Git repository

`vogsphere@vogsphere.42.fr:intra/2015/activities/get_next_line/cly`



## Introduction

Ce barème contient des zones où VOUS devez décider, selon votre propre expérience, de quelle façon doit être contrôlée une fonctionnalité demandée.

Nous vous demandons, pour le bon déroulement de cette notation :

- De rester courtois, poli, respectueux, constructif, lors de cet échange. Le lien de confiance entre la communauté 42 et vous en dépend.
- De bien mettre en évidence auprès de la personne notée (ou du groupe) les dysfonctionnements éventuels.
- D'accepter 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 raison ou tort ?), et notez le plus honnêtement possible.

Bonne soutenance à tous !

## Guidelines

VOUS NE DEVEZ CORRIGER QUE CE QUI SE TROUVE SUR LE DÉPÔT DE RENDU DE L'ÉTUDIANT.

Constatez que l'étudiant ou le groupe à noter est bien dans un répertoire VIDE. Il effectue un "git clone" de son dépôt de rendu.

Vérifiez que les éléments suivants sont bien conformes au sujet :

- il y a bien un rendu
- rien d'autre dans le rendu que ce qui est demandé
- le fichier auteur
- la présence ou non du Makefile avec les règles demandées
- s'il n'y a pas de Makefile, que le programme compile bien
- la Norminette indique bien qu'il n'y a pas de problème de norme

Si un élément n'est pas conforme au sujet, la notation s'arrête là. Vous pouvez toujours débattre du projet, mais le barème n'est pas appliqué.

Pour la suite, quelle que soit la partie, y compris les bonus, si le programme:

- segfault
- buserror
- double free
- tout autre indicateur qui révèle un comportement inadéquat du programme vis à vis du système alors la notation s'arrête.

Si il y a triche, la notation s'arrête. Pour mémoire, un cas de triche correspond à un code qui n'est pas fait ou pas reproductible par son auteur, ou bien l'utilisation d'éléments explicitement interdits ou qui visent à dénaturer le rôle pédagogique du projet. A utiliser avec calme, précaution, et discernement.

Pour le `get_next_line`, pas de variable globale. Sera également considéré comme triche des fonctions spécifiques à `get_next_line` dans la bibliothèque `libft` pour contourner les limitations de la Norme. Prenez bien le temps de parcourir les sources de la bibliothèque `libft` de l'étudiant.


Dans n'importe lequel de ces cas de figure, sélectionnez le bouton correspondant à droite (invalid compilation en cas de Makefile non fonctionnel).

## Ratings

Define the type of error (if there is an error), which ended the correction.

☒ Ok   ☐ Empty work   ☐ Incomplete work   ☐ No author file   ☐ Invalid compilation   ☐  Norme   ☐ Cheat   ☐ Crash

## Attachments

 Subject ([https://cdn.intrav2.42.fr/pdf/pdf/16/get\\_next\\_line.pdf](https://cdn.intrav2.42.fr/pdf/pdf/16/get_next_line.pdf))

 Intra - Elearning - Subnotions (<https://elearning.intrav2.42.fr/notions/62/subnotions>)

## Sections

### Préliminaires

#### Conditions de base

Les conditions suivantes :

- présence des fichiers `get_next_line.c` et `get_next_line.h`
- `get_next_line.h` contient le prototype de la fonction et la macro qui définit le nombre de caractères lus simultanément sur le `filedescriptor`. Nous l'appellerons ici "`BUFF_SIZE`" (mais chacun peut l'appeler comme il le souhaite).

doivent être remplies. Si ce n'est pas le cas, la notation avec ce barème s'arrête. Vous pouvez malgré tout continuer de discuter sur le projet.

Attention à bien séparer les erreurs dans `get_next_line` et celles qui pourraient être présentes dans le main de test (gardez un main de test très simple).

☒ Yes

☐ No

### Tests

## Tests basiques

Choisissez `BUFF_SIZE` a 8, et compilez un programme de test qui lit sur l'entree standard grace a la fonction `get_next_line`.

Effectuez au minimum les tests suivants :

- Lire et retourner une ligne de 8 caractères terminée par un `'\n'` inclus depuis l'entrée standard.
- Lire et retourner deux lignes de 8 caractères terminées par un `'\n'` inclus depuis l'entrée standard.
- Lire et retourner un nombre quelconque de lignes de 8 caractères terminées par un `'\n'` inclus depuis l'entrée standard.

Ajoutez un open de `argv[1]` dans votre main, puis :

- Lire et retourner une ligne de 8 caractères terminée par un `'\n'` inclus depuis un fichier.
- Lire et retourner deux lignes de 8 caractères terminées par un `'\n'` inclus depuis un fichier.
- Lire et retourner un nombre quelconque de lignes de 8 caractères terminées par un `'\n'` inclus depuis un fichier.

✓ Yes

No

## Tests intermédiaires

- Lire et retourner une ligne de 16 caractères terminée par un `'\n'` inclus depuis un fichier.
- Lire et retourner deux lignes de 16 caractères terminées par un `'\n'` inclus depuis un fichier.
- Lire et retourner un nombre quelconque de lignes de 16 caractères terminées par un `'\n'` inclus depuis un fichier.
- Lire et retourner une ligne de 16 caractères terminée par un `'\n'` inclus depuis l'entrée standard.
- Lire et retourner deux lignes de 16 caractères terminées par un `'\n'` inclus depuis l'entrée standard.
- Lire et retourner un nombre quelconque de lignes de 16 caractères terminées par un `'\n'` inclus depuis l'entrée standard.

✓ Yes

No

## Tests avancés

- Lire et retourner une ligne de 4 caractères terminée par un `'\n'` inclus depuis un fichier.
  - Lire et retourner deux lignes de 4 caractères terminées par un `'\n'` inclus depuis un fichier.
  - Lire et retourner un nombre quelconque de lignes de 4 caractères terminées par un `'\n'` inclus depuis un fichier.
  - Lire et retourner une ligne de 4 caractères terminée par un `'\n'` inclus depuis l'entrée standard.
  - Lire et retourner deux lignes de 4 caractères terminées par un `'\n'` inclus depuis l'entrée standard.
  - Lire et retourner un nombre quelconque de lignes de 4 caractères terminées par un `'\n'` inclus depuis l'entrée standard.
  - Lire et retourner une ligne de 4 caractères sans `'\n'` depuis un fichier.
  - Lire et retourner une ligne de 8 caractères sans `'\n'` depuis un fichier.
  - Lire et retourner une ligne de 16 caractères sans `'\n'` depuis un fichier.
- (rappel : la fin de fichier doit se comporter comme une fin de ligne pour votre `get_next_line`).
- Lire et retourner une ligne vide depuis un fichier.
  - Lire et retourner une ligne vide depuis l'entrée standard.

✓ Yes

No

## Gestion d'erreur

## Gestion d'erreur

Effectuez AU MOINS les tests suivants afin de tenter de mettre en défaut la gestion d'erreur :

- Passer un descripteur de fichier arbitraire à la fonction `get_next_line` sur lequel il n'est pas possible de lire, par exemple 42. La fonction doit retourner -1.
- Mettre `BUFF_SIZE` à 1, 32, 9999 puis 10000000. Le dernier cas peut ne pas fonctionner (et ne doit pas compter comme une erreur dans cette soutenance). L'un de vous deux sait-il pourquoi ?

✓ Yes

No

## Bonus

### Bonus multi `filedescriptor`

Ne considérez ce bonus que si vous avez répondu OUI à toutes les questions précédentes.

Effectuez les tests pour contrôler que ce bonus est bien fonctionnel.

✓ Yes

No

### Autres bonus

Y-a-t-il d'autres bonus ? (comme une seule variable statique utilisée)

Faites votre évaluation par rapport au nombre de bonus supplémentaires présents, utiles, et fonctionnels (1 point par bonus).



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

## Conclusion

Leave a comment on this correction

**\* (required) Comment**

Finish correction