



ASTEK



Colle Text Count

Responsable Astek astek_resp@epitech.eu

Abstract:



Table des matières



.1 Consignes

- Le sujet peut changer jusqu'à une heure avant le rendu.
- Vos exercices doivent être à la norme.
- Binaire et exemple non contractuels (le sujet fait foi).
- Votre programme devra compiler avec un Makefile.
- Travaillez en local !
C'est à dire que pour chaque exercice vous devez le compiler sur votre compte linux.
Ceci dans le simple but de ne pas surcharger les serveurs car vous êtes nombreux.
- Le binaire s'appellera `text_count`.
- Vous avez le droit d'utiliser les fonctions `malloc`, `free`, `exit`, `write`, `read`, `open` et `close`.
- Vous n'avez pas le droit à votre `libmy`.
- Vous n'avez pas le droit à internet.
- Il n'y a pas de rendu sur les serveurs :
présentez-vous à la soutenance avec votre machine



.2 PARTIE 1 : "Compte tout"

- `./text_count [STR]`
- Pour cette première partie vous devez faire un programme qui affiche le nombre de chacun des caractères du texte passé en paramètre. Vous n'affichez pas les caractères à 0.
- Exemple :

```
$ ./text_count "abcdef"
a:1
b:1
c:1
d:1
e:1
f:1
$ ./text_count "          "
:8
$
```



.3 PARTIE 2 - "Compte une partie"

- `./text_count [STR] [STR]`
- Pour cette partie, vous devez compter les occurrences de caractères de la première chaîne seulement pour les caractères dans la seconde chaîne.
- Exemple :

```
$ ./text_count "ffaabcdef" "abcdefg"
a:2
b:1
c:1
d:1
e:1
f:3
$ ./text_count "ffaabcdef" "abc"
a:2
b:1
c:1
$ ./text_count "ffaabcdef" "aaa"
a:2
$ ./text_count "zzzzzzzzzzzz" "aaa"
$ ./text_count "zzzzzzzzzzzz" ""
$ ./text_count "zzzzzzzzzzzz"
z:11
```



Vous noterez dans le dernier exemple que si l'on n'envoie pas la seconde chaîne on doit traiter TOUS les caractères.



.4 PARTIE 3 - "Compte une partie et dans l'ordre"

- `./text_count -t [STR] [STR]`
- Avec l'option `-t`, vous devez faire du dénombrement comme en partie 2 et triez par quantité.
- Exemple :

```
$ ./text_count -t "ffaabcdef" "abcdefg"
f:3
a:2
b:1
c:1
d:1
e:1
$ ./text_count -t "ffaabcdefa" "abc"
a:3
b:1
c:1
$
```



.5 PARTIE 4 - "Compte une partie et range-les joliment"

- `./text_count -r [STR] [STR]`
- Pour cette partie, vous devez faire du dénombrement comme la partie 2 et supporter l'option `-r` qui range.
- Exemple :

```
$ ./text_count -r "ffaabcdef" "abcdefg"  
3:f  
2:a  
1:b,c,d,e
```



Vous noterez que les virgules sont présentes uniquement entre 2 éléments.



.6 PARTIE 5

- Ajouter l'option `-reverse` qui affiche à l'envers.
- Ajouter l'option `-fullstat` qui refait des stats sur les denombrements.
- Ajouter l'option `-on_the_fly` les deux chaines qui étaient passées en paramètre sont maintenant lues sur l'entrée standard.
- Toutes ces options DOIVENT :
 - Etre parsées dans n'importe quel ordre
 - Supporter les paramètres `=off` et `=on`.
 - Utiliser le dernier paramètre passe (voir exemple)

- Exemple :

```
$ ./text_count -r -reverse "ffaabcdef" "abcdefg"
1:b,c,d,e
2:a
3:f
=> reverse apres -r
```

```
$ ./text_count -reverse -r "ffaabcdef" "abcdefg"
1:b,c,d,e
2:a
3:f
=> reverse avant -r
```

```
$ ./text_count -r "ffaabcdef" "abcdefg" -reverse
1:b,c,d,e
2:a
3:f
=> reverse apres les parametres
```

```
$ ./text_count -r -reverse=on "ffaabcdef" "abcdefg"
1:b,c,d,e
2:a
3:f
=> -reverse=on a le meme effet que -reverse
```

```
$ ./text_count -r -reverse=off "ffaabcdef" "abcdefg"
3:f
2:a
1:b,c,d,e
=> -reverse=off a le meme effet que de ne pas mettre -reverse
```




```
$ ./text_count -reverse -r -reverse=off "ffaabcdef" "abcdefg"
3:f
2:a
1:b,c,d,e
=> -reverse=off apres avoir mis un -reverse => on annule le -reverse
```

```
$ ./text_count -reverse -reverse -r -reverse=off "ffaabcdef" "abcdefg"
3:f
2:a
1:b,c,d,e
=> pareil meme si il y en a plusieurs, on prend que le dernier.
```

```
$ ./text_count -reverse -reverse -r -reverse=off -reverse "ffaabcdef" "abcdefg"
1:b,c,d,e
2:a
3:f
=> pareil meme si il y en a plusieurs, on prend que le dernier.
```

```
$ ./text_count -r --fullstat "ffaabcdefb" "abcdefghij"
3:f(1)(10%)
2:ab(2)(20%)
1:c,d,e(3)(30%)
=> compte le nombre de lettre pour la quantite: "1:c,d,e(3)" signifie 3 lettres
ont une occurence: la lettre c,d et e.
=> il faut ensuite ajouter le %age de lettre de cette occurence dans les lettres
=> ici 30% des lettres surveilles dans le dernier champs on une occurence de 1 :
3 lettres(cde) sur les 10(abcdefghij)
```