Exercice 1 :

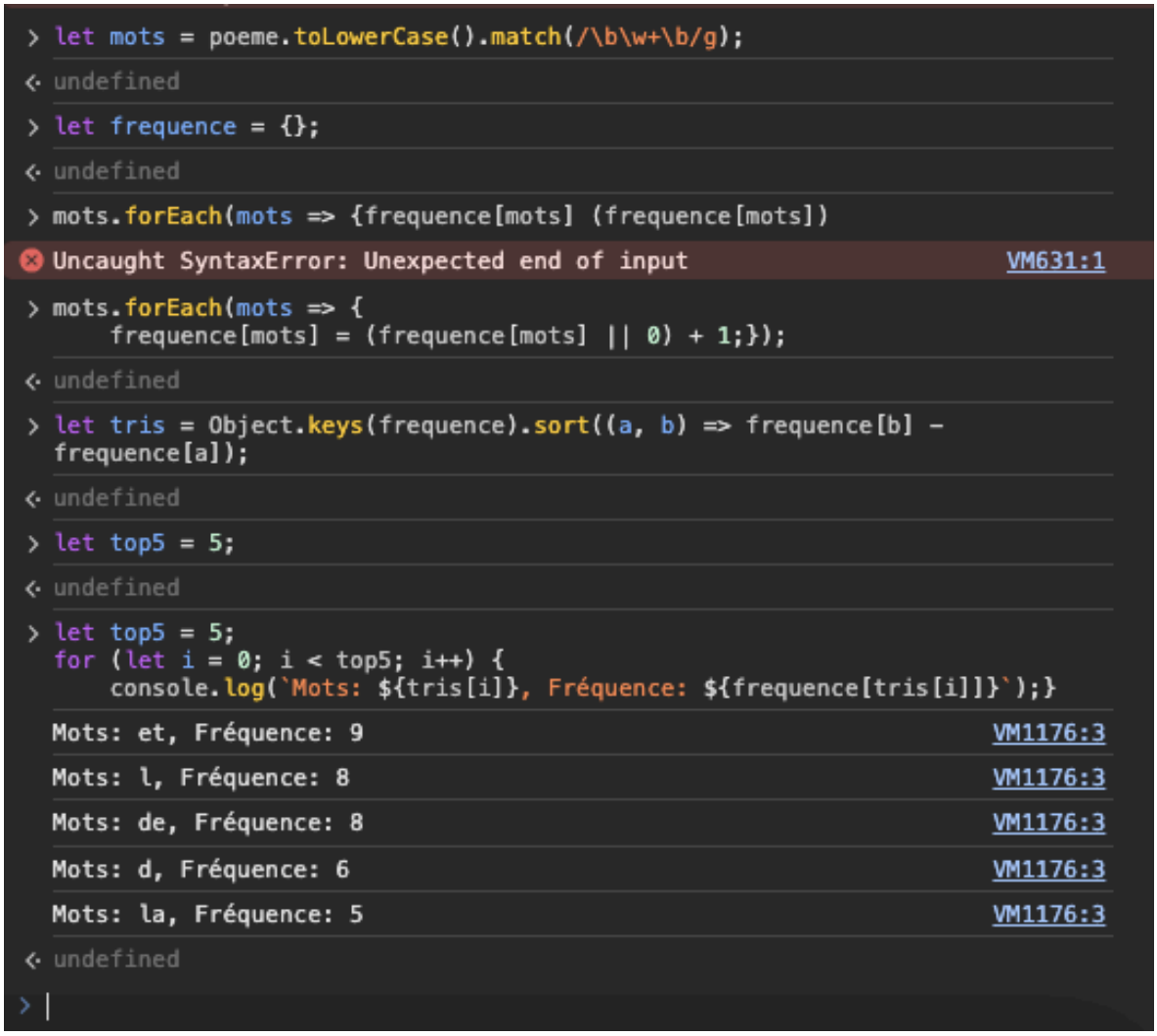
```
> let poeme = `A LA FEMME AIMÉE
  Lorsque tu vins, à pas réfléchis, dans la brume,
  Le ciel mêlait aux ors le cristal et l'airain.
  Ton corps se devinait, ondoiement incertain,
  Plus souple que la vague et plus frais que l'écume.
  Le soir d'été semblait un rêve oriental
  De rose et de santal.

  Je tremblais. De longs lys religieux et blêmes
  Se mouraient dans tes mains, comme des cierges froids.
  Leurs parfums expirants s'échappaient de tes doigts
  En le souffle pâmé des angoisses suprêmes.
  De tes clairs vêtements s'exhalaient tour à tour
  L'agonie et l'amour.

  Je sentis frissonner sur mes lèvres muettes
  La douceur et l'effroi de ton premier baiser.
  Sous tes pas, j'entendis les lyres se briser
  En criant vers le ciel l'ennui fier des poètes
  Parmi des flots de sons languissamment décrus,
  Blonde, tu m'apparus.

  Et l'esprit assoiffé d'éternel, d'impossible,
  D'infini, je voulus moduler largement
  Un hymne de magie et d'émerveillement.
  Mais la strophe monta bégayante et pénible,
  Reflet naïf, écho puéril, vol heurté,
  Vers ta Divinité.`
< undefined
```

La variable « let poème » permet de déclarer mon poème et son titre.

Exercice 2 :

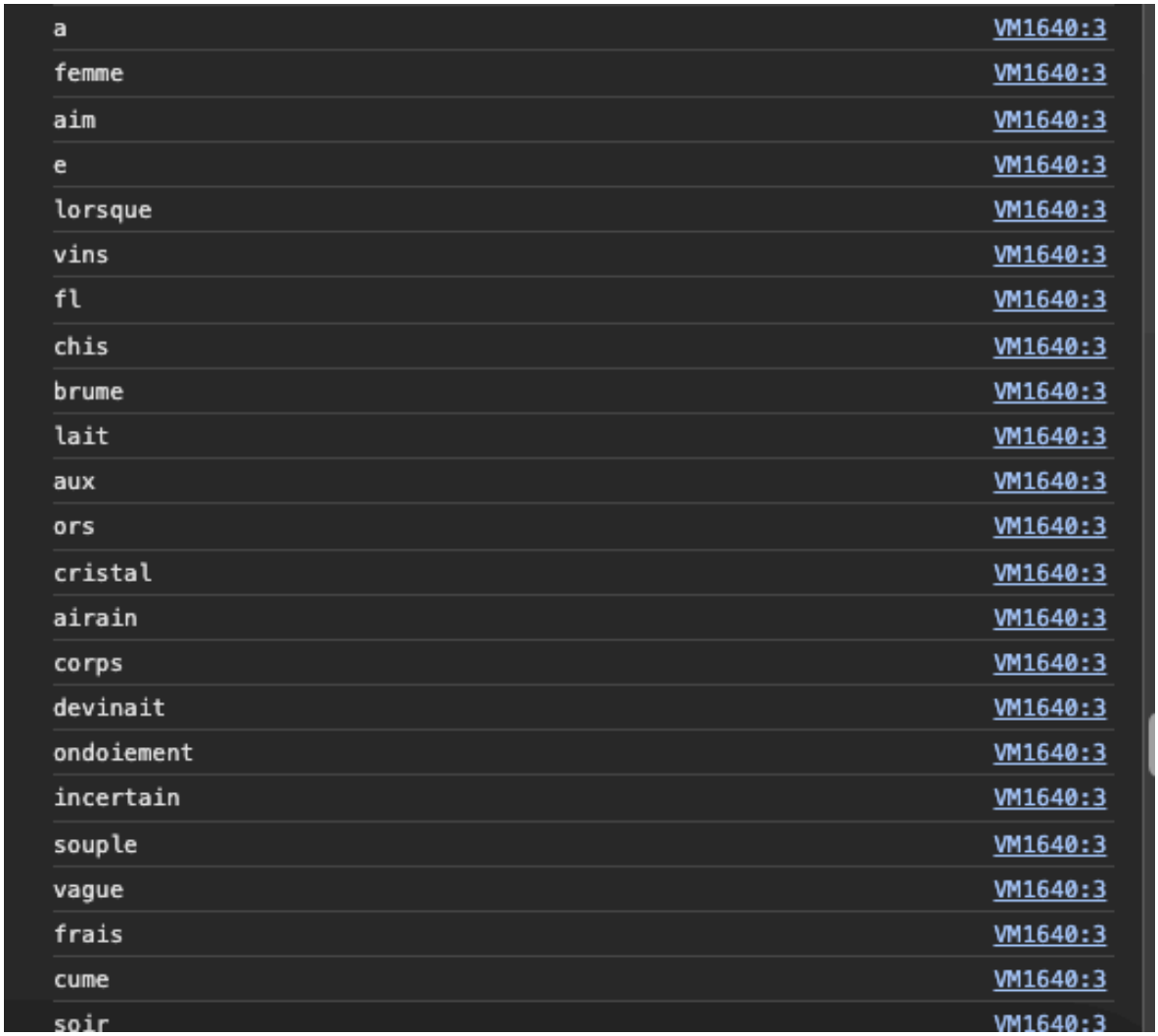
Dans l'exercice 2, j'ai commencé par identifier les 5 mots les plus fréquents dans ce poème :
- Le poème est d'abord présenté comme un tableau et il est défini avec « mots ».
- Ensuite, j'ai initialisé « frequence » pour stocker les futures données comme un ensemble vide.
- « forEach » permet de parcourir les éléments du tableau « mots ».
- Les mots du poème sont classés en fonction de leur fréquence d'apparition dans un certain ordre : du plus fréquent au moins fréquent.
- J'utilise la boucle « for » ensuite pour pouvoir afficher les mots qui sont les plus fréquents pour qu'ils oient stockés dans un tableau.

Les 5 mots les plus fréquents que j'ai obtenu sont « et » (9 fois), « l » (8 fois), « d » (8 fois), « de » (6 fois) et « la » (5 fois).



Après cela, j'essaye de calculer la richesse lexicale du poème en comptant le nombre de mots uniques et le nombre total de mots.
Pour ce dernier, on peut utiliser « mots » pour afficher tous les éléments présents dans le tableau. Sinon, on peut utiliser la ligne de code « let total = mots.length; » pour avoir le nombre total de mots.
Pour le premier, les mots uniques, donc ceux qui n'apparaissent que une seule fois dans le texte du poème, on utilise la boucle « for…in » pour parcourir l'ensemble du tableau. Avec l'utilisation de la condition « if…===1 », on peut vérifier que la fréquence est bien égale à 1. Si c'est le cas, cela signifie que le mot n'apparaît qu'une seule fois dans notre poème.


> let total = mots.length;
undefined
> mots
(196) ['a', 'la', 'femme', 'aim', 'e', 'lorsque', 'tu', 'vins', 'pas', 'r', 'fl', 'chis', 'dans', 'la', 'brume', 'le', 'ciel', 'm', 'lait', 'aux', 'ors', 'le', 'cristal', 'et', 'l', 'airain', 'ton', 'corps', 'se', 'devinait', 'ondoiement', 'incertain', 'plus', 'souple', 'que', 'la', 'vague', 'et', 'plus', 'frais', 'que', 'l', 'cume', 'le', 'soir', 'd', 't', 'semblait', 'un', 'r', 've', 'oriental', 'de', 'rose', 'et', 'de', 'santal', 'je', 'tremblais', 'de', 'longs', 'lys', 'religieux', 'et', 'bl', 'mes', 'se', 'mouraient', 'dans', 'tes', 'mains', 'comme', 'des', 'cierges', 'froids', 'leurs', 'parfums', 'expirants', 's', 'chappaient', 'de', 'tes', 'doigts', 'en', 'le', 'souffle', 'p', 'm', 'des', 'angoisses', 'sup r', 'mes', 'de', 'tes', 'clairs', 'v', 'tements', 's', 'exhalaient', 'tou r', …]


a VM1640:3
femme VM1640:3
aim VM1640:3
e VM1640:3
lorsque VM1640:3
vins VM1640:3
fl VM1640:3
chis VM1640:3
brume VM1640:3
lait VM1640:3
aux VM1640:3
ors VM1640:3
cristal VM1640:3
airain VM1640:3
corps VM1640:3
devinait VM1640:3
ondoiement VM1640:3
incertain VM1640:3
souple VM1640:3
vague VM1640:3
frais VM1640:3
cume VM1640:3
soir VM1640:3

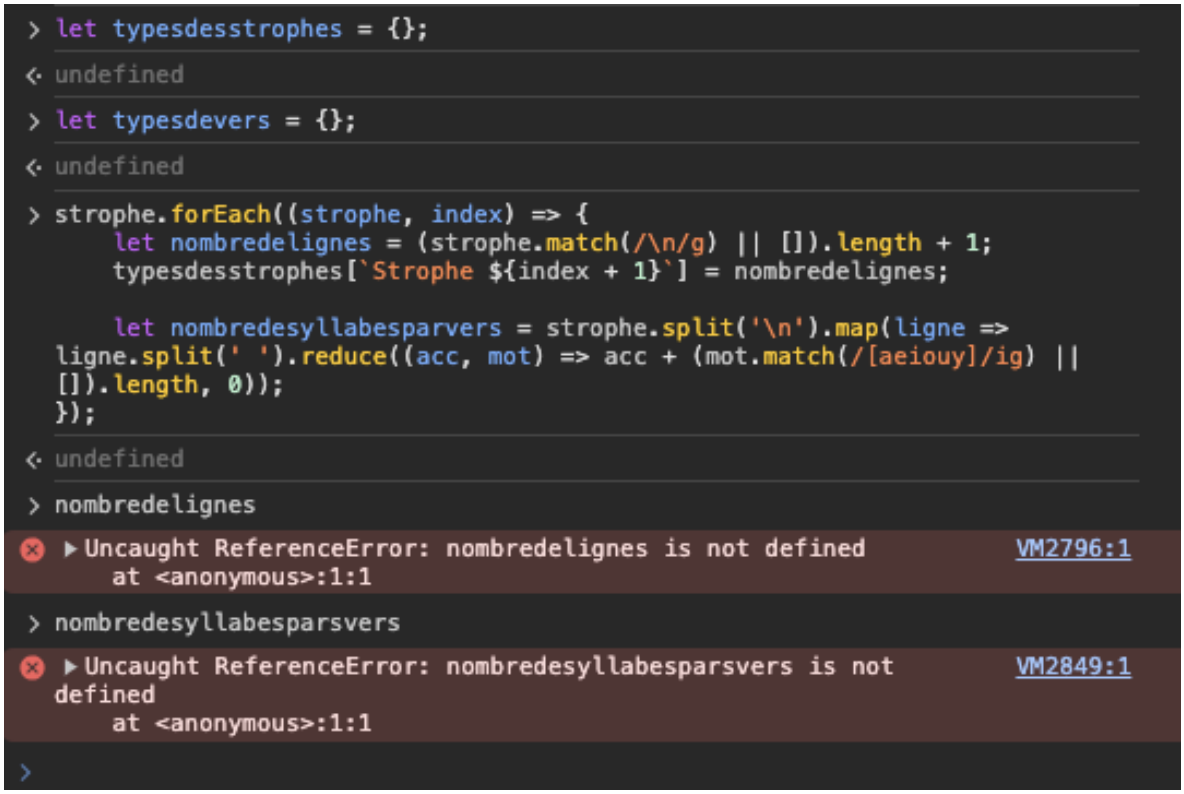
Exercice 3 :

« Let phrase » permet de rechercher une chaîne de caractères précise commençant par une majuscule et terminant par « . », « ! » ou « ? ». « let nombredephrases » permet de compter le nombre de phrases obtenues avec la variable « let phrase ».

Il y a 11 phrases dans le poème.

Exercice 4 :

Ce code nous permet d'identifier les strophes présentes dans ce poème. J'ai commencé par supprimer tous les espaces qui sont en début et en fin de chaque chaîne de caractères. Après ça, j'ai définis la délimitation des strophes comme étant les caractères « \r » et « \n » qu'on obtient après que le poème aie été mis sous la forme d'un tableau. Ensuite, on peut compter les strophes présentes.



Tout d'abord, nous définissons des espaces vides appelés « typesdesstrophes » et « typesdevers ». Ensuite, nous examinons chaque ligne de chaque strophe du poème pour compter le nombre de syllabes.

(strophe.match(/\n/g) ll []).length + 1; → est l'utilisation d'une expression régulière pour compter le nombre de ligne en recherchant les sauts de ligne. Puis on stocke tout cela dans notre ensemble vide « typesdesstrophes ».
Puis, nous subdivisons le texte en lignes puis en mots afin de compter le nombre de syllabes. Cette information est ensuite stockée dans le tableau « nombredesyllabesparvers ».

Enfin, nous parcourons l'ensemble du contenu du tableau « nombredesyllabesparvers » et transférons cette information dans l'ensemble vide « typesdevers » que nous avions créé précédemment.

On peut compter 4 strophes dans le poème.

Mais je n'arrive pas à trouver comment faire pour avoir le nombre de syllabes par vers et le nombre de vers par strophes. Cependant, je sais que chaque strophe est composé elle-même de 6 vers et que chaque vers, excepté le dernier de chaque strophe, est composé de 12 syllabes, ceux sont donc des alexandrins. Le dernier vers de chaque strophe est composé de 6 syllabes, ceux sont donc des hexasyllabes.