TP5

Langage C (LC4)

semaine du 25 février

1 Libérez-vous de vos chaînes !

Le but de cette section est d'implémenter une structure de données pour stocker des chaînes arbitrairement longues. Indice : pensez de manière récursive.

Question 1. Déclarez dans un fichier `chaine.h` une structure `struct str_s` aliasée vers `str_s` (pensez à `typedef`). Définissez la structure dans le fichier `chaine.c` avec comme champs `c` de type `char` et `next` de type `struct str_s *`.

Note Par convention, la valeur de `next` à la dernière case sera `NULL`. Sauf spécifié, les déclarations de chaque fonction seront faites dans `chaine.h` et leurs définitions dans `chaine.c`.

Question 2. Considérons la chaîne "hello". Combien de place faut-il pour la stocker dans un `char *` ? Et pour `str_s` ?

Question 3. Quel intérêt peut-il y avoir à ne pas définir la structure `struct str_s` dans le fichier `chaine.h` ?

1.1 Entrée, sortie

Question 4. Créez une fonction `void str_print(str_s *s)` qui affichera la chaîne sur la sortie standard.

Question 5. Créez une fonction `str_s* str_init(const char* s)` qui transformera une chaîne en une liste de caractères de type `str_s*` et la retournera.

Question 6. Créez une fonction `void str_free(str_s *s)` qui libèrera la mémoire allouée à la liste `s`.

Question 7. Créez une fonction `str_s* str_scan()` qui récupèrera une chaîne saisie sur l'entrée standard jusqu'au retour à la ligne `'\n'`.

Question 8. Dans un fichier `main.c` incluant `chaine.h`, demandez à l'utilisateur de rentrer une chaîne arbitrairement longue, et affichez-la. N'utilisez pas `char*` mais `str_s`.

Note N'oubliez pas de libérer la mémoire avec `str_free` à la fin du programme !

1.2 Les e ets

Question 9. Créez une fonction `str_s* str_concat(str_s* s1, str_s* s2)` qui concatène les chaînes `s1` et `s2` en faisant pointer la dernière case de `s1` vers `s2` et retourne le résultat.

Question 10. Trouvez l'erreur du programme suivant et corrigez-la. Quel est le résultat ?

```c
#include <stdlib.h>
#include "chaine.h"

int main() {

  str_s *s1 = str_init("hello"), *s2 = str_init("world");
  str_s *s3 = str_concat(s1, s2);

  str_print(s1);
  str_print(s2);
  str_print(s3);

  str_free(s1);
  str_free(s2);
  str_free(s3);
  return 0;
}
```

Question 11. Quel est l'affichage de l'instruction suivante `str_print(str_concat(s1, s1))` si `s1 = str_init("hello")` ?

Question 12. Créez une fonction `str_s* str_cpy(str_s* s)` qui duplique la chaîne `s` et retourne l'adresse de la chaîne copiée.

Question 13. Modifiez `str_s* str_concat(str_s *s1, str_s *s2)` pour que la chaîne `s1` reste inchangée après l'appel de la fonction.

1.3 Insertion/suppression

Question 14. Créez une fonction `int str_length(str_s* s)` qui retourne la taille de la chaîne `s`.

Question 15. Créez une fonction `int str_insert(str_s* s, char c, int n)` qui insère le caractère `c` à la position `n`.

Question 16. Créez une fonction `int str_remove(str_s* s, char c)` qui supprime toutes les occurences du caractère `c`.

Question 17. Créez une fonction `str_s* str_tri(str_s* s)` qui trie les caractères par ordre ASCII.