Université de Nice-Sophia Antipolis
Polytech ELEC3

Lundi 5 février 2018
Durée: 2h

CPP - Chaînes de caractères
Travaux Pratiques – Séance n° 9

---

1 CPP - Le préprocesseur

Le préprocesseur s'applique dans une phase qui précède celle de la compilation proprement dite, passe sur tout le texte source du programme et traite toutes les directives qui lui sont destinées. Ces directives sont introduites par un dièse (#). Nous avons déjà vu les directives #**define**, pour définir une constante, et #**include**, pour inclure le contenu d'un fichier.

Pour mettre en évidence le fonctionnement réel, nous allons jouer avec les options `-P` et `-save-temps` de `gcc`.

1) Reprenez un fichier de la séance précédente qui contient un #**define** et un #**include**. Compilez-le avec les deux options précisées ci-dessus. Décrivez les fichiers produits.

2) À quoi sert l'option `-P` ? Quelle information enlève cette option ?

3) Décrivez le comportement exact du #**define** et du #**include**. Quelles informations sont perdues après le passage du préprocesseur ?

4) Écrivez, compilez, exécutez et expliquez le comportement du programme suivant `prog.c` :

```
#include <stdlib.h>
#include <stdio.h>
int main(void) {
  printf("%s %d\n", __FILE__, MAX);
  return EXIT_SUCCESS;
}
```

```
gcc -DMAX=300 -o prog prog.c && ./prog
```

5) Que se passe-t-il si `MAX` est aussi défini dans le programme ? Vérifiez.

2 Chaînes de caractères

En C, il n'existe pas de type chaîne de caractères spécifique. Les chaînes de caractères sont des tableaux de caractères, mais le dernier caractère doit être le délimiteur de fin de chaîne *'\0'*. Ainsi, **sizeof** *"abcdefghi"* est égal à 10 et non pas 9.

Une constante chaîne de caractères peut être définie par une suite de caractères délimitée par des guillemets, ou à l'aide d'une initialisation classique de tableau. Le caractère *'\0'* est automatiquement ajouté à la fin d'une constante littérale. Le caractère \<saut de ligne> permet d'écrire la constante sur plusieurs lignes.


```
char str1 [10] = {'a', 'b', 'c', '\0'};
char str2 [10] = "abc";
"Une chaîne de caractères"
"" /* une chaîne vide */
" \
  ?" /* une chaîne sur plusieurs lignes */
```

Contrairement à d'autres langages, il n'y a aucun opérateur spécifique pour les chaînes de caractères en C. La bibliothèque C contient de nombreuses fonctions de manipulation de chaînes de caractères. Les prototypes des fonctions sont définis dans `string.h`. Faites un `man string.h` pour avoir une idée précise. Il est important de noter que ces fonctions ne gèrent pas l'allocation de la mémoire pour les caractères de chaînes (sauf `strdup`).

6) Écrivez et testez le programme suivant :

```
#include <stdlib.h>
#include <stdio.h>
#include <string.h>
int main(void) {
  char str1 [10] = {'c', 'o', 't', '\0'};
  char str2 [10] = "abri";

  printf("(%s) : (%s)\n", str1, str2);
  printf("(%c) : (%c)\n", str1[1], str2[2]);
  printf("(%lu) : (%lu)\n", sizeof str1, sizeof "abc");
  printf("(%zd) : (%zd)\n", strlen(str1), strlen(str2));
  printf("(%s)\n", strcat(str2,str1));
  return EXIT_SUCCESS;
}
```

7) Récrivez les fonctions `strlen` et `strchr`.

8) Récrivez les fonctions `strcpy` et `strcat`.

9) Écrivez une fonction `compare` qui prend deux chaînes de caractères, s1, et s2, en paramètre et les compare selon l'ordre lexicographique. La fonction renverra :

- une valeur négative, si s1 est inférieure à s2 ;
- 0, si s1 est égale à s2 ;
- une valeur positive, si s1 est supérieure à s2.

10) Comment récrire les fonctions précédentes afin qu'elles puissent faire des vérifications sur la taille des chaînes (par exemple, pour `strcpy`, que le premier paramètre ait assez de place pour accueillir tous les caractères du deuxième) ?

11) Sans utiliser de fonctions de la bibliothèque C, écrivez en C la procédure `trim` qui prend en paramètre deux chaînes de caractères `s1` et `s2`. `trim` construit la chaîne `s2` à partir de `s1` dans laquelle tous les caractères espaces (*i.e.* *' '*) de début et de fin ont été supprimés. Si la chaîne `s1` est *" bonne année 2017 "*, la chaîne résultat `s2` sera égale à *"bonne année 2017"*. Attention à bien penser à tous les cas possibles.