

Quelques notions de C avancées

Ensimag 1A Apprentissage, Logiciel de Base

2011

Pour vous faire gagner du temps, des squelettes de programmes C sont fournis dans le répertoire `c/3-c-avance/` de votre archive.

1 Arguments du programme (1-arguments.c)

Question 1 *Ecrire une fonction `main` qui parcourt les arguments de la ligne de commande et les concatène dans un tableau de caractères qui est affiché à la fin. On fera un tableau de taille fixe, en définissant une constante `TAILLE`, par exemple égale à 16.*

```
char *strcpy(char *dst, const char *src);  
char *strcat(char *s, const char *append);  
size_t strlen(const char *s);
```

2 Arithmétique de pointeurs (2-arithmetique.c)

Question 2 *Ecrire deux fonctions permettant de détecter automatiquement si une machine fonctionne en big-endian ou en little-endian.*

Question 3 *Ecrire une fonction prenant en argument un tableau d'entiers terminé par l'entier 0, et inversant le codage mémoire des entiers de ce tableau (i.e. passage de little-endian à big-endian). On fournit une fonction `main` et une fonction permettant d'afficher le contenu du tableau en hexadécimal.*

Question 4 *Que fait la fonction suivante ?*

```
void f(char *a, char *b) {  
    while(*a++ = *b++);  
}
```

3 Masques et décalages (3-masques-bits.c)

Question 5 *Ecrire une fonction affichant une valeur codée sur 8 bits en binaire. Ecrire une fonction échangeant les bits d'un octet (i.e. bit 7 ↔ bit 0, bit 6 ↔ bit 1, etc.). Quelle est la différence entre l'opérateur `!` et l'opérateur `~` ?*

4 Pointeur de fonctions (4-qsort.c)

Question 6 *Écrire une fonction qui fait un tri de tableau d'entiers, en utilisant la fonction `qsort` de la bibliothèque standard (`man qsort` devrait vous en dire plus).*

5 Pile, tas, variables globales

Trouver les erreurs et proposez une correction dans les morceaux de code suivants :

Question 7

```
int *f() {
    int x = 1;
    return &x;
}
```

Question 8

```
int *creer_tableau() {
    int x[] = {1, 2, 3, 4, 42};
    return x;
}
```

Question 9

```
char *chaine = "une_chaine";

void majuscule(char *chaine) {
    if (chaine[0] >= 'a' and chaine[0] <= 'z') {
        chaine[0] += 'A' - 'a';
    }
}

int main() {
    majuscule(chaine);
    /* ... */
}
```

6 Encore des bugs vicieux !

Le répertoire `c/3-c-avance/bugs/` de l'archive Git contient 9 petits fichiers C et un Makefile pour les compiler rapidement. Regardez les sources de ces programmes, et exécutez le résultats. Les programmes ne font pas ce qu'on aurait attendu d'eux ... essayez de comprendre leurs comportements.