Pratique du C :
Introduction aux pointeurs

Licence Informatique — Université Lille
Pour toutes remarques : Alexandre.Sedoglavic@univ-lille1.fr

Semestre 4 — 2016-2017

Exercice 1 — Passage de paramètres.

Écrivez une fonction qui échange les valeurs de deux variables entières.

Exercice 2 — Inversion destructive d'une chaîne de caractères.

Donnez la définition d'une fonction de prototype :

```
void inverse(char tab[]) ;
```

qui inverse la chaîne de caractères codée par le tableau `tab` dans ce même tableau.

Par exemple, si le tableau `char str[]="foo";" est pass\'ee en param\‘etre \verb`inverse(str);+ alors ce tableau contiendra la chaîne `"oof"`.

Exercice 3 — Une implantation de strcmp.

Donnez la définition d'une fonction C de prototype :

```
int strcmp(char tab1[], char tab2[]);
```

qui retourne :

- 0 si `tab1` et `tab2` code la même chaîne de caractères ;
- -1 si `tab1` code une chaîne de caractères plus petite que celle codée par `tab2` pour l'ordre lexicographique ;
- 1 si `tab1` code une chaîne de caractères plus grande que celle codée par `tab2` pour l'ordre lexicographique.

Exercice 4 — Tableaux à plusieurs dimensions.

Soit la déclaration d'un tableau

```
int b[3][5];
```

En considérant que l'allocation du tableau se fait linéairement en mémoire (les 3 ≪ tranches ≫ de `b` sont allouées à des adresses contiguës), donnez l'état du tableau `b` après l'exécution du code C suivant :

```
int b[3][5];
int *a = *b, i;
for (i=0 ; i<15 ; *a++ = i++)
    ;
**b = 15;          **(b+1) = 16;          *(b[0]+1) = 17;
*(*b+8) = 18;      *(b[1]+2) = 19;        *(*(b+1)+5) = 20;
*(b[2]+3) = 21;    *(*(b+2)+2) = 22;
```

Exercice 5 — Affichage des paramètres du shell.

Donnez la définition d'une fonction de prototype :

```
int main(int argc, char **argv) ;
```

qui affiche les paramètres du shell passés à l'exécutable correspondant.

Indication : vous utiliserez la fonction

```
#include <stdio.h>
int putchar(int c);
```

pour l'affichage sur la sortie standard.

Exercice 6 — Noms temporaires, allocations automatique, statique ou dynamique.

Donnez la définition d'une fonction qui retourne à chaque invocation une chaîne de caractères différente, par exemple en vue de l'utiliser comme un nom de fichier temporaire.

Exercice 7 — Une fonction de comparaison.

En utilisant l'exercice 1 de la feuille de TD 4, donnez la définition d'une fonction de prototype :

```
int CompareDateNaissance (const void *, const void *) ;
```

qui retourne 0 (resp. -1 ou 1) si la personne représentée par la structure pointée par le premier pointeur passé en paramètre a le même age (resp. est plus jeune ou plus vielle) que la personne représentée par la structure pointée par le second pointeur passé en paramètre.

Exercice 8 — Recherche par dichotomie.

Donnez la définition de la fonction de prototype :

```
#include <stdlib.h>
void * bsearch (const void *key, const void *base, unsigned int nel,
unsigned int width, int (*compar) (const void *, const void *));
```

qui retourne l'adresse de la cellule *correspondant* à l'objet `key`, dans un tableau de `nmemb` cellules, commençant à l'adresse `base`. La taille des cellules du tableau est de `size` octets.

On suppose que le contenu de la table doit est trié en ordre croissant par rapport à la fonction de comparaison référencée par `compar`. La fonction pointée doit retourner respectivement -1, 0, ou 1 si l'objet pointée par le premier pointeur est inférieur, égal, ou supérieur à l'object pointée par le second pour l'ordre implanté par cette fonction.

La fonction `bsearch` renvoie NULL si aucune cellule ne correspond. Si plusieurs cellules de la table correspondente à la clef, celui qui est renvoyé n'est pas spécifié.