

Introduction à l'informatique

Exercices — Série 9

Exercice 1

Déterminer ce qui sera affiché lors de l'exécution des trois fragments de code C suivants :

```
— int  x[2] = { 0, 0 };
   int  *y;

   y = x;
   (++y)[0] = 1;
   printf("%d %d\n", x[0], x[1]);
```

```
— int a, b, *c;

   a = 1;
   b = 2;
   c = &a;
   a += *c;
   c = &b;
   *c-- = 3;
   printf("%d %d\n", a, b);
```

```
— struct
{
    int val, *adr;
} x, *y;

x.adr = &x.val;
y = &x;
x.val = 41;
y -> val = 42;
printf("%d\n", *(y -> adr));
```

Exercice 2 (*Examen de janvier 2019*)

Quelle est la valeur affichée lors de l'exécution du fragment de code suivant ? (Justifier votre réponse.)

```
int a[] = { 1, 0 };
int *b[2];
int **c;

b[*a] = a + 1;
c = b;

printf("%d\n", **++c);
```

Exercice 3 (*Examen de juin 2019*)

Quelle est la valeur affichée lors de l'exécution du fragment de code suivant ? (Justifier votre réponse.)

```
int t[2] = { 1, 2 };
int *a, **b;

a = t + *t;
b = &a;

printf("%d\n", *--*b);
```

Exercice 4 (*Examen de septembre 2019*)

Quelle est la valeur affichée lors de l'exécution du fragment de code suivant ? (Justifier votre réponse.)

```
int t[2] = { -2, 0 };
int *a, b;

b = *t ? ++*t : --*t;
a = &b;

printf("%d\n", a[*(t + 1)]);
```

Exercice 5

Soit une fonction `int rand(int min, int max)` qui retourne un nombre entier aléatoire compris dans l'intervalle $[\text{min}, \text{max}]$. On demande d'écrire une fonction qui alloue un tableau d'entiers de taille donnée et qui remplit celui-ci avec des entiers aléatoires compris dans un intervalle donné.

La fonction demandée est déclarée comme suit :

```
int rand_tab(unsigned int n, int min, int max, int **dest);
```

où

- `n` est la taille du tableau,
- `min` et `max` sont les bornes inférieure et supérieure des nombres aléatoires à insérer dans le tableau,
- `dest` la destination du tableau généré.

La fonction retourne une valeur différente de zéro si le tableau a pu être généré et zéro si ce n'est pas le cas (en particulier, si `min > max` ou `dest == NULL`).

Exercice 6

Soit une fonction `int f(int *x)` donnée. Cette fonction est capable de produire un nombre limité et a priori inconnu d'entiers. Lorsque `f` est appelée, si un entier est produit, il est stocké dans l'espace pointé par `x` et `f` retourne une valeur différente de zéro. Sinon, `f` retourne zéro et aucun appel suivant ne produira d'entier.

On demande d'écrire un programme qui remplit, au moyen de `f`, un tableau d'entiers `t` de taille `n` donnée ($n \geq 0$) à partir de l'indice zéro et tant que `f` produit un nombre. À la fin de l'exécution du programme, la variable `nb` contiendra le nombre d'entiers que `f` a produit (avec $nb \leq n$).

Exercice 7 (*Examen de janvier 2019*)

1. Écrire un fragment de code C définissant les deux types structurés suivants :
 - un type `point` représentant les coordonnées cartésiennes (x, y) d'un point dans le plan, où x et y sont des valeurs réelles.
 - un type `segment` représentant un segment de droite caractérisé par son origine et sa destination (qui sont des points).
2. Écrire une fonction C prenant en arguments quatre réels x_1, y_1, x_2 et y_2 définissant les extrémités (x_1, y_1) et (x_2, y_2) d'un segment de droite, et retournant un pointeur vers une représentation de ce segment, nouvellement allouée.
3. Écrire une fonction C permettant de libérer une représentation d'un segment créée par la fonction obtenue au point (b).

Exercice 8 (*Examen de juin 2019*)

1. Écrire un fragment de code C définissant un type structuré capable de retenir le nom et le prénom d'une personne, tous deux représentés sur au plus 20 caractères.
2. Écrire une fonction C prenant en argument des pointeurs vers deux instances du type obtenu à la question précédente, et retournant une valeur booléenne qui indique si les deux personnes concernées possèdent le même nom et le même prénom. La comparaison doit tenir compte des caractères non alphabétiques, mais pas de la casse (majuscule ou minuscule) des lettres ; par exemple, les noms “le petit” et “Le Petit” sont considérés égaux, mais différents de “Lepetit”. Il est permis de définir des fonctions auxiliaires de votre choix, mais pas de faire appel à des fonctions issues d'une bibliothèque.

Exercice 9 (*Examen de septembre 2019*)

1. Écrire un fragment de code C définissant un type structuré capable de représenter un intervalle $[a, b]$ de nombres entiers, avec $a \in \mathbb{Z} \cup \{-\infty\}$ et $b \in \mathbb{Z} \cup \{+\infty\}$ tels que $a \leq b$.
2. Écrire une fonction C prenant en argument des pointeurs vers deux instances du type obtenu à la question précédente, et retournant une valeur booléenne qui indique si le premier intervalle est entièrement inclus dans le second. (Par exemple l'intervalle $[1, 4]$ est inclus dans $[0, +\infty]$, mais pas dans $[-2, 3]$.) Il est permis de définir des fonctions auxiliaires de votre choix.