

Tous documents interdits à l'exception d'une page A4 recto/verso, manuscrite de votre main. La correction tiendra compte de la qualité de l'argumentaire et de la présentation. Votre copie n'est pas un brouillon.

Exercice 1 : ROT13 (chaînes de caractères et fichiers – 4pts)

Le ROT13 (rotate by 13 places) est un algorithme très simple de chiffrement de texte. Il s'agit d'un décalage de 13 caractères de chaque lettre du texte à chiffrer. Puisqu'il y a 26 lettres et que le décalage est de 13, il suffit d'appliquer deux fois de suite le chiffrement pour obtenir le texte en clair : la même procédure est utilisée pour le chiffrement et le déchiffrement. Pour cela on doit considérer l'alphabet comme circulaire, c'est-à-dire que $Z+1=A$. Les accentuées sont inchangées. (D'après Wikipédia).

On souhaite écrire un programme appliquant ce chiffrement à une chaîne de caractères donnée.

Caractère non-chiffré	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
Caractère chiffré	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M

(1.1) Écrivez un programme complet prenant une chaîne de caractères à analyser en argument de la ligne de commande, de la manière suivante (2pts).

```
$ ./rot13 "Bonjour les consolidés. Bon courage pour ce partiel."  
Obawbhe yrf pbafbyvqéf. Oba pbhentr cbhe pr cnegvry.  
$ ./rot13 './rot13 "Bonjour les consolidés. Bon courage pour ce partiel."'  
Bonjour les consolidés. Bon courage pour ce partiel.
```

(1.2) Modifiez votre programme afin qu'il lise le texte à analyser depuis un fichier dont le nom est passé en argument de la ligne de commandes (2pt).

Exercice 2 : Gestion de BDthèque (programmation structurée – 6pts)

Afin de ne plus égarer les bandes dessinées que je prête autour de moi, je vous demande de m'aider à réaliser un petit programme de gestion de ma collection. Chaque ouvrage est défini par son genre (manga, SF, medfan, historique, qui sont les seuls genres de ma collection), son titre et sa date d'achat. Chaque ouvrage peut de plus être disponible ou bien prêté, auquel cas je dois enregistrer le nom de la personne à qui j'ai prêté mon livre, et la date.

- (2.1) Écrivez une structure de données `bd_t` contenant toutes les informations relatives à une BD donnée. Vous suivrez l'approche plutôt orientée objet vue en cours. (1pt)
- (2.2) Écrivez la fonction `bd_new()`, prenant en arguments les informations initiales, et retournant une référence à un «objet», c'est-à-dire un pointeur sur une nouvelle structure `bd_t` (2pt).
- (2.3) Écrivez la fonction `bd_free()` prenant une référence à une BD pour libérer la mémoire occupée (1pt).
- (2.4) Écrivez une fonction `bd_est_dispo()` retournant si oui ou non la BD en paramètre est chez moi.
- (2.5) Écrivez une fonction `bd_pret(bd_t BD, char *qui, char *quand)` qui s'assure que la BD en paramètre est marquée comme prêtée à la personne indiquée (1pt).

NOM :

PRÉNOM :

Exercice 3 : QCM (10pts)

Répondez sur cette feuille. Une seule case juste par question. Les réponses fausses seront pénalisées.

— Début du programme —

```

1 #include <stdlib.h>
2 #include <stdio.h>
3 void f1 (int a, int *b){
4     a = *b;
5 }
6 void f2 (int *b, int c){
7     *b = c;
8 }
9 void f3 (int *a, int c){
10    f4 ( &a, c );
11 }
12 void f4 (int **b, int a){
13     **b = a;
14 }

```

— Fin du programme —

```

15 int main () {
16     int x = 5, y = 7, z = 9;
17     f1 ( x, &y );
18     printf ( "x = %d, y = %d, z = %d\n", x, y, z );
19     f2 ( &x, y );
20     printf ( "x = %d, y = %d, z = %d\n", x, y, z );
21     f3 ( &y, z );
22     printf ( "x = %d, y = %d, z = %d\n", x, y, z );
23 }

```

(3.1) Qu'affiche la ligne 18 ? x = y = z =

(3.2) Qu'affiche la ligne 20 ? x = y = z =

(3.3) Qu'affiche la ligne 22 ? x = y = z =

(3.4) Quel est le type de chacune des variables dans cet extrait de programme ? (4pts)

```

typedef struct {
    char *marque;
    int nb_places;
} voiture_t,
*voiture;

```

```

voiture_t a,b[12];
voiture c,d[12];

```

variable	entier	flottant	structure	pointeur	tableau	(invalide)
a	<input type="checkbox"/>	<input type="checkbox"/>	X	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
a.nb_places	X	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
a->nb_places	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	X
b	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	X	<input type="checkbox"/>
b.nb_places	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	X
b->nb_places	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	X
b[0].nb_places	X	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
b[0]->nb_places	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	X
c	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	X	<input type="checkbox"/>	<input type="checkbox"/>
c.nb_places	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	X
c->nb_places	X	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
d	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	X	<input type="checkbox"/>
d.nb_places	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	X
d->nb_places	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	X
d[0].nb_places	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	X
d[0]->nb_places	X	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

(3.5) Si dans un fichier *makefile* on trouve la ligne suivante : `toto: tutu` cela veut dire (1pt) :
☒ il faut reconstruire toto quand tutu change ☐ il faut reconstruire tutu quand toto change

(3.6) Que ne devrait-on PAS trouver dans les fichiers d'entêtes (.h) d'un programme bien écrit (1pt) :
☒ Des définitions de fonctions ☐ Des prototypes de fonctions ☐ Des définitions de types

(3.7) Que fait le code ci-contre ? (1pt)

- ☐ Elle affecte à p la valeur -5 ☐ Elle affecte à p la valeur -1
☐ L'adresse de p vaut -3 ☐ L'adresse de p vaut -1
☒ Elle affecte à troisième case entière de p la valeur -3

```

int* p=malloc(3*sizeof(int));
*(p + 2) = -3;

```