

# Eléments de programmation 2 – 1I002 – CORRIGÉ

Examen du 9 mai 2016 (2 heures)

Seule la fiche mémo est autorisée

*Les calculatrices, baladeurs et autres appareils électroniques sont interdits. Les téléphones mobiles doivent être éteints et rangés dans les sacs. Le barème sur 60 points n'a qu'une valeur indicative. La dernière question de cet examen est une question bonus. Vous pouvez donc avoir une bonne note sans la traiter ou bien vous pouvez la traiter pour compenser d'éventuelles erreurs faites dans vos réponses aux autres questions. Avant de commencer, vérifiez que votre copie contient **12 pages** avec 17 questions.*

## 1 Compilation (7 pts)

### Question 1 (4 points)

Pour tester son compilateur, une informaticienne a écrit le programme `test.c` suivant :

```
#define A 1664
#define B 17

int main() {
    int a;
    int b = B;
    float c = 42.25;
    printf("%d_%d_%d\n", a, b, c);
    return 0;
}
```

Lorsqu'elle essaie de le compiler avec la commande `gcc -Wall -Werror -o test test.c` elle voit que le compilateur émet les messages suivants :

```
test.c: In function 'main':
test.c:8:3: error: implicit declaration of function 'printf'
[-Werror=implicit-function-declaration]
    printf("%d %d %d\n", a, b, c);
    ^
test.c:8:3: error: incompatible implicit declaration of built-in
function 'printf' [-Werror]
test.c:8:3: error: format '%d' expects argument of type 'int',
but argument 4 has type 'double' [-Werror=format=]
test.c:8:3: error: 'a' is used uninitialized in this function
[-Werror=uninitialized]
ccl: all warnings being treated as errors
```

A-t-elle obtenu un fichier exécutable ? Si vous répondez « oui, elle obtient un exécutable », expliquez pourquoi le compilateur arrive à produire cet exécutable. Si vous répondez « non, elle n'obtient pas d'exécutable », donnez les raisons qui empêchent le compilateur de le produire.

**Solution:** Non, l'informaticienne n'a pas obtenu de fichier exécutable. Le programme contient de petites erreurs qui provoquent des avertissements. Comme l'option `-Werror` a été spécifiée, tous les avertissements sont traités comme des erreurs, d'où l'absence d'exécutable.

### Question 2 (3 points)

Corrigez tous les problèmes dans le programme `test.c` de la question précédente. Recopiez complètement le programme. Votre programme doit garder le comportement voulu ; en particulier il doit afficher `1664 17 42.250000`.

**Solution:**

```
#include <stdio.h>

#define A 1664
#define B 17

int main() {
    int a = A;
    int b = B;
    float c = 42.25;
    printf("%d_%d_%f\n", a, b, c);
    return 0;
}
```

## 2 Pointeurs, tableaux et structures - Questions de cours (9 pts)

### Question 3 (2 points)

Cochez les bonnes réponses dans le tableau suivant :

Un pointeur permet de	vrai	faux
– modifier l'adresse d'une variable	<input type="checkbox"/>	<input type="checkbox"/>
– accéder à une variable déclarée dans une fonction appelante	<input type="checkbox"/>	<input type="checkbox"/>
– convertir d'un type à un autre, par exemple d'un entier à un réel	<input type="checkbox"/>	<input type="checkbox"/>
– éviter les recopies de structures de données larges	<input type="checkbox"/>	<input type="checkbox"/>

**Solution:**

Un pointeur permet de	vrai	faux
– modifier l'adresse d'une variable	<input type="checkbox"/>	<input checked="" type="checkbox"/>
– accéder à une variable d'une fonction appelante	<input checked="" type="checkbox"/>	<input type="checkbox"/>
– convertir d'un type à un autre, par exemple d'un entier à un réel	<input type="checkbox"/>	<input checked="" type="checkbox"/>
– éviter les recopies de structures de données larges	<input checked="" type="checkbox"/>	<input type="checkbox"/>

### Question 4 (4 points)

Lesquelles des manipulations de variables suivantes sont correctes ? Si elles ne le sont pas, comment peut-on les corriger ?

1. `int a; int *pa = &a; *pa = 42;`
2. `char b; char *pb = &b; pb = 'L';`
3. `int c; float *pc = &c;`
4. `float d[42]; float *pd = d + 17; *pd = 112.25;`

Toute réponse « non, incorrect » sans proposition de correction sera considérée comme fausse.

**Solution:**

1. `int a; int *pa = &a; *pa = 42;` : correct
2. `char b; char *pb = &b; pb = 'L';` : incorrect, l'affectation `pb = 'L';` ne respecte pas les types ; à gauche, on a un pointeur vers caractère, à droite un caractère. On ne peut pas créer de pointeur vers caractère à partir d'un caractère, resp. entier. Correction : `char b; char *pb = &b; *pb = 'L';`
3. `int c; float *pc = &c;` : incorrect, le pointeur `&c` est de type `int *` alors qu'il faut un pointeur `float *` pour initialiser `pc`. Correction : `float c; float *pc = &c;` ou `int c; int *pc = &c;`
4. `float d[42]; float *pd = d + 17; *pd = 112.25;` : correct, `d` est un tableau de 42 valeurs en virgule flottante, donc aussi un pointeur vers un flottant. L'initialisation de `pd` par arithmétique de pointeurs est donc correcte. On obtient même un pointeur valide parce qu'on se met à la 18-ième case d'un tableau qui en a 42.

**Question 5** (3 points)

Donnez la déclaration d'une structure `vetement` qui décrit un vêtement et qui contient les champs suivants :

- un champ `description`, qui contient une description textuelle du vêtement, sous forme d'une chaîne de caractères d'au plus 32 caractères (en laissant la place pour le marqueur de fin),
- un champ `taille`, qui indique la taille du vêtement sous la forme d'un réel,
- un booléen `femme`, qui indique qu'il s'agit d'un vêtement pour femmes.

**Solution:**

```
struct vetement {  
    char description[33];  
    float taille;  
    int femme;  
};
```

### 3 Récursivité (9 pts)

**Question 6** (4 points)

Soit  $(u_n)_n$  la suite des entiers  $u_n$  définie par la récurrence

$$\begin{aligned}u_0 &= 42 \\ u_1 &= 17 \\ u_n &= 2u_{n-1} - 3u_{n-2}.\end{aligned}$$

Écrivez une fonction **réursive** `suite_u`, qui étant donné un entier  $n$  positif ou nul en argument, calcule et renvoie  $u_n$ .

**Solution:**

```
int suite_u(int n) {
    if (n == 0) return 42;
    if (n == 1) return 17;
    return 2 * suite_u(n-1) - 3 * suite_u(n-2);
}
```

#### Question 7 (5 points)

La fonction exponentielle  $e^x$  vérifie

$$e^x = \left(e^{\frac{x}{2}}\right)^2.$$

Pour une valeur  $x$  petite en valeur absolue, disons telle que  $-0.01 \leq x \leq 0.01$ , on peut approcher cette fonction  $e^x$  par le polynôme  $p(x) = 1 + x + 0.5x^2$ .

On peut donc écrire une fonction **réursive** `exponentielle` qui prend un réel  $x$  en argument et qui calcule et renvoie  $e^x$  (aussi un réel) de la façon suivante :

- si  $x$  est petit en valeur absolue, c'est-à-dire entre  $-0.01$  et  $0.01$ , la fonction renvoie  $1 + x + 0.5x^2$ ,
- sinon, elle calcule  $E = e^{\frac{x}{2}}$  d'abord et en déduit  $e^x = E^2$ .

Écrivez cette fonction **réursive** `exponentielle`.

**Solution:**

```
float exponentielle(float x) {
    float E;
    if ((-0.01 <= x) && (x <= 0.01)) {
        return 1.0 + x + 0.5 * x * x;
    }
    E = exponentielle(x/2);
    return E * E;
}
```

## 4 Boucles et chaînes de caractères (12 pts)

#### Question 8 (2 points)

Écrivez une fonction `longueur` qui prend en entrée un tableau de caractères rempli avec une chaîne de caractères et qui renvoie la longueur (en caractères) de cette chaîne.

Par exemple pour la chaîne de caractères "Salut!", votre fonction doit renvoyer 6.

Votre fonction ne doit faire aucun affichage. Elle ne doit prendre qu'un seul paramètre, le tableau avec la chaîne de caractères. La fonction ne doit pas être réursive.

Votre fonction ne doit pas se servir de fonctions définies dans une bibliothèque du système.

**Solution:**

```
int longueur(char chaine[]) {
    int i;
    i = 0;
    while (chaine[i] != '\0') {
        i++;
    }
    return i;
}
```

**Question 9** (3 points)

Écrivez une fonction `majuscules` qui prend en entrée un tableau de caractères rempli avec une chaîne de caractères et qui modifie ce tableau pour remplacer tous les caractères en minuscule (sans accent) par les majuscules correspondantes.

Votre fonction ne doit faire aucun affichage. Elle ne doit prendre qu'un seul paramètre, le tableau avec la chaîne de caractères. **Elle doit laisser les caractères qui ne sont pas des minuscules tels quels ; elle ne doit pas les modifier.** Votre fonction ne doit pas se servir de la fonction `longueur` définie à la question précédente. Elle ne doit pas être récursive.

**Solution:**

```
void majuscules(char chaine[]) {
    int i;
    i=0;
    while (chaine[i] != '\0') {
        if ((chaine[i] >= 'a') && (chaine[i] <= 'z')) {
            chaine[i] = 'A' + (chaine[i] - 'a');
        }
        i++;
    }
}
```

**Question 10** (2 points)

Écrivez un programme qui déclare une chaîne de caractères et l'initialise avec un texte contenant à la fois des lettres minuscules, des lettres majuscules et de la ponctuation, l'affiche, puis applique la fonction `majuscules` programmée à la question précédente pour transformer les minuscules en majuscules et affiche la chaîne obtenue.

Par exemple, votre programme affichera

I kissed a girl and I liked it, the taste of her cherry chapstick.

I KISSED A GIRL AND I LIKED IT, THE TASTE OF HER CHERRY CHAPSTICK.

Si vous n'avez pas su répondre à la question précédente, donnez d'abord le prototype de la fonction `majuscules`, puis servez-vous-en comme si vous l'aviez définie à la question précédente.

Vous devez donner un programme complet, avec toutes les inclusions de fichiers en-tête de bibliothèque, la fonction `main` etc. En revanche, vous ne devez pas recopier la fonction `majuscules`.

**Solution:**

```
#include <stdio.h>
```

```
int main() {
    char chaine[] = "I_kissed_a_girl_and_I_liked_it,_the_taste_of_her_
        cherry_chapstick.";

    printf("%s\n", chaine);
    majuscules(chaine);
    printf("%s\n", chaine);

    return 0;
}
```

**Question 11** (5 points)

Écrivez une fonction `rotation` qui prend en entrée un tableau de caractères rempli avec une chaîne de caractères et qui modifie ce tableau pour effectuer une rotation d'un caractère vers la gauche sur la chaîne :

- le premier caractère de la chaîne devient son dernier et
- les autres caractères sont tous avancés d'une position.

Par exemple, la fonction `rotation` transforme la chaîne de caractères "abcd" en "bcda".

Votre fonction peut se servir de la fonction `longueur` définie à la question 8. Si vous n'avez pas su répondre à la question 8, donnez juste le prototype de cette fonction et supposez-la définie.

La fonction ne doit pas se servir d'autre tableau de caractères pour effectuer une copie intermédiaire.

**Vous pouvez utiliser une variable auxiliaire de type `char`.**

Étant donné une chaîne de caractères vide (""), la fonction `rotation` ne doit rien faire.

Votre fonction ne doit faire aucun affichage. Elle ne doit prendre qu'un seul paramètre, le tableau avec la chaîne de caractères. **La fonction `rotation` ne doit pas être récursive.** La fonction doit maintenir le marqueur de fin de la chaîne de caractères inchangé, à la position où il se trouve initialement.

**Solution:**

```
void rotation(char chaine[]) {
    int longueur_chaine;
    int i;
    char premier_caractere;
    longueur_chaine = longueur(chaine);
    if (longueur_chaine != 0) {
        premier_caractere = chaine[0];
        for (i=1; i<longueur_chaine; i++) {
            chaine[i-1] = chaine[i];
        }
        chaine[longueur_chaine-1] = premier_caractere;
    }
}
```

## 5 Fonctions et pointeurs (10 pts)

### Question 12 (5 points)

On considère le programme suivant :

```
#include <stdio.h>

float a;
int b;

void f(float x) {
    a = a + x;
    printf("%d\n", b);
}

float g(char d) {
    int a = 9;
    a = a + (d - 'A');
    b = b + 1;
    printf("%d\n", a + b);
    return a * 13.5;
}

void h() {
    a = 17.0;
}

int main() {
    float a = 42.0;
    b = 1664;
    h();
    f(1.0);
    a = g('B');
    printf("%f_%d\n", a, b);
    return 0;
}
```

Donnez un programme qui n'utilise pas de variables globales et qui ait le même comportement que le programme ci-dessus.

**Solution:**

```
#include <stdio.h>

void f(float x, float *a, int b) {
    *a = *a + x;
    printf("%d\n", b);
}
```

```
}

float g(char d, int *b) {
    int a = 9;
    a = a + (d - 'A');
    *b = *b + 1;
    printf("%d\n", a + *b);
    return a * 13.5;
}

void h(float *a) {
    *a = 17.0;
}

int main() {
    float a_globale;
    int b;
    float a = 42.0;
    b = 1664;
    h(&a_globale);
    f(1.0, &a_globale, b);
    a = g('B', &b);
    printf("%f_%d\n", a, b);
    return 0;
}
```

**Question 13** (3 points)

Écrivez une fonction `imin_imax` qui prend en entrée un tableau de nombres en virgule flottante ainsi que la taille de ce tableau et qui rend à la fonction appelante à la fois l'indice du minimum et l'indice du maximum des valeurs dans le tableau. Pour un tableau vide, c'est-à-dire de taille nulle, la fonction rend -1 comme indices de minimum et de maximum.

Pour le tableau de taille 7 déclaré comme

```
float tab[] = { -1.0, -17.0, 42.0, 1664.0, -1789.0, 19.0, -22.25 };
```

l'indice du minimum (-1789.0) est 4 et l'indice du maximum (1664.0) est 3.

**Solution:**

```
void imin_imax(int *imin, int *imax, float tab[], int taille) {
    int i;

    if (taille < 1) {
        *imin = -1;
        *imax = -1;
    } else {
        *imin = 0;
        *imax = 0;
```



```
for (i=1;i<taille;i++) {  
    if (tab[i] < tab[*imin]) {  
        *imin = i;  
    }  
    if (tab[i] > tab[*imax]) {  
        *imax = i;  
    }  
}  
}
```

**Question 14** (2 points)

Donnez une fonction `affiche_min_max` qui prend en entrée un tableau de nombres en virgule flottante ainsi que la taille de ce tableau et qui affiche les indices et les valeurs du minimum et du maximum de ce tableau. Vous pouvez supposer que le tableau est de taille non-nulle.

**Votre fonction `affiche_min_max` doit forcément être basée sur la fonction `imin_imax`, programmée à la question précédente.** Si vous n'avez pas su répondre à cette question, donnez juste le prototype de la fonction `imin_imax` et supposez-la définie.

**Solution:**

```
void affiche_min_max(float tab[], int taille) {  
    int imin, imax;  
  
    imin_imax(&imin, &imax, tab, taille);  
    printf("La_valeur_minimale_se_trouve_a_l'indice_%d_et_vaut_%f\n",  
        imin, tab[imin]);  
    printf("La_valeur_maximale_se_trouve_a_l'indice_%d_et_vaut_%f\n",  
        imax, tab[imax]);  
}
```

## 6 Problème : Affichage défilant (13 pts)

Dans cette section, on désire écrire un programme qui effectue plusieurs rotations sur une chaîne de caractères pour faire un affichage défilant.

Le programme final affichera donc par exemple

```
NAITRE ET MOURIR POUR LA PAIX  
ET MOURIR POUR LA PAIX NAITRE  
MOURIR POUR LA PAIX NAITRE ET  
POUR LA PAIX NAITRE ET MOURIR  
LA PAIX NAITRE ET MOURIR POUR  
PAIX NAITRE ET MOURIR POUR LA
```

Pour bien structurer le programme, on va d'abord écrire deux fonctions auxiliaires. Le programme complet va être développé à la question 17.

**Question 15** (7 points)

Pour décaler la chaîne d'un nombre de caractères tel que le deuxième mot de la chaîne devienne son premier, il faut déterminer l'indice de la première lettre du deuxième mot de la chaîne.

On suppose que les mots sont séparés par des espaces ('\_'). Il n'y a donc pas de ponctuation ni de blancs formés par des tabulations ou des retours à la ligne.

Par exemple, dans la chaîne "NAITRE\_ET\_MOURIR\_POUR\_LA\_PAIX\_", le deuxième mot est "ET" et se trouve à l'indice 7 dans le tableau contenant cette chaîne.

Le deuxième mot d'une chaîne de caractères est typiquement à l'indice où se trouve le premier caractère qui n'est pas une espace et qui se trouve **après le premier mot**. Ce premier mot se termine à la première espace qui le suit.

Attention : le premier mot d'une chaîne de caractères ne commence pas forcément à l'indice 0 ! La chaîne peut commencer par des espaces. Par exemple, dans la chaîne "\_\_\_\_VIVE\_\_\_\_LA\_\_\_\_FRANCE\_", le premier mot commence à l'indice 4 et se termine à l'indice 7. Le deuxième mot ne commence qu'à l'indice 12.

Écrivez une fonction `indiceDeuxiemeMot` qui prend en entrée un tableau de caractères rempli avec une chaîne de caractères et qui renvoie l'indice du deuxième mot de la chaîne de caractères, quelle que soit la disposition des espaces autour du premier et deuxième mot de la chaîne. Si la chaîne ne contient pas de deuxième mot (chaîne vide, chaîne ne contenant que d'espaces, chaîne ne contenant qu'un seul mot etc.), la fonction doit retourner l'indice du marqueur de fin de la chaîne.

Votre fonction ne doit faire aucun affichage. Elle ne doit prendre qu'un seul paramètre, le tableau avec la chaîne de caractères. Elle ne doit pas être récursive.

**Solution:**

```
int indiceDeuxiemeMot(char chaine[]) {
    int recherche_debut_premier_mot = 1;
    int recherche_espace_apres_premier_mot = 0;
    int trouve = 0;
    int i = 0;

    while ((chaine[i] != '\0') && (!trouve)) {
        if (recherche_debut_premier_mot) {
            /* On cherche le debut du premier mot, apres les possibles
               espaces au debut de la chaine .
            */
            if (chaine[i] != '_') {
                /* On a trouve un autre caractere qu'une espace, on passe a
                   la
                   recherche de l'espace apres le premier mot.
                */
                recherche_debut_premier_mot = 0;
                recherche_espace_apres_premier_mot = 1;
            }
        } else {
            if (recherche_espace_apres_premier_mot) {
                /* On cherche la premiere espace apres le premier mot. */
                if (chaine[i] == '_') {
```

```
        /* On a trouve une premiere espace. */
        recherche_espace_apres_premier_mot = 0;
    }
} else {
    /* On cherche le debut du deuxieme mot. */
    if (chaine[i] != ' ') {
        /* On a trouve un autre caractere qu'une espace, donc on a
           trouve le debut du deuxieme mot.
        */
        trouve = 1;
    }
}
}
/* Il ne faut avancer que quand on n'a pas trouve le deuxieme
   mot
*/
if (!trouve) i++;
}
return i;
}
```

**Question 16** (3 points)

Écrivez une fonction `rotationNCaracteres` qui prend en entrée un tableau de caractères rempli avec une chaîne de caractères et un entier  $n$ , supposé positif ou nul. Elle modifie ce tableau pour effectuer une rotation de  $n$  caractères vers la gauche sur la chaîne :

- les  $n$  premiers caractères de la chaîne deviennent ses  $n$  derniers et
- les autres caractères sont tous avancés de  $n$  positions.

La fonction `rotationNCaracteres` appelée par exemple avec  $n = 7$  transforme la chaîne "NAITRE\_ET\_MOURIR\_POUR\_LA\_PAIX\_" en "ET\_MOURIR\_POUR\_LA\_PAIX\_NAITRE\_".

Votre fonction **doit** réutiliser la fonction `rotation` définie à la question 11 (en l'appelant). Si vous n'avez pas su définir de fonction à la question 11, donnez-en juste le prototype et supposez-la définie.

Votre fonction ne doit faire aucun affichage. Elle ne doit prendre que deux paramètres, le tableau avec la chaîne de caractères et l'entier  $n$ . **La fonction ne doit pas être récursive.** La fonction doit maintenir le marqueur de fin de la chaîne de caractères inchangé, à la position où il se trouve initialement.

**Solution:**

```
void rotationNCaracteres(char chaine[], int n) {
    int i;
    for (i=0; i<n; i++) {
        rotation(chaine);
    }
}
```

**Question 17** (3 points)

Écrire un programme complet (avec inclusion des fichiers en-têtes des bibliothèques, `main` etc.) qui se

sert des fonctions `longueur` (question 8), `indiceDeuxiemeMot` (question 15) et `rotationNCaracteres` (question 16) pour réaliser l’affichage suivant :

```
NAITRE ET MOURIR POUR LA PAIX
ET MOURIR POUR LA PAIX NAITRE
MOURIR POUR LA PAIX NAITRE ET
POUR LA PAIX NAITRE ET MOURIR
LA PAIX NAITRE ET MOURIR POUR
PAIX NAITRE ET MOURIR POUR LA
```

Votre programme doit partir d’une seule chaîne de caractères déclarée dans `main` sur laquelle opèrent les fonctions `longueur`, `indiceDeuxiemeMot` et `rotationNCaracteres`.

Votre programme doit être indépendant de la chaîne de caractères donnée. On peut supposer cette chaîne non-vide et composée d’au moins deux mots, séparés par des espaces. **En l’occurrence, vous ne devez pas déterminer de longueurs de chaînes ou sous-chaînes et/ou de nombres de tours de boucle à la main.**

On remarquera que l’affichage défilant peut s’arrêter quand le nombre total de caractères soumis à la rotation atteint ou dépasse la longueur de la chaîne.

**Solution:**

```
int main() {
    char chaine[] = "NAITRE_ET_MOURIR_POUR_LA_PAIX_";
    int longueur_chaine;
    int rotations_totales = 0;
    int k;

    longueur_chaine = longueur(chaine);

    do {
        printf("%s\n", chaine);
        k = indiceDeuxiemeMot(chaine);
        rotationNCaracteres(chaine, k);
        rotations_totales = rotations_totales + k;
    } while (rotations_totales < longueur_chaine);

    return 0;
}
```