

Perfectionnement à la programmation en C

Examen – Licence 2 Informatique 2022 – 2023

mercredi 17 mai 2023

Ce devoir est constitué de 16 questions. Voici quelques conseils :

- les réponses aux questions sont **indépendantes** à l'exception des questions 5 à 8.
- respecter les **conventions** étudiées. Une fonction qui « marche » peut être incorrecte ;
- lorsqu'il est demandé de définir une fonction, seuls les paramètres principaux sont précisés. Il est ainsi possible d'**ajouter d'autres paramètres** ;
- les seules fonctions externes, types et macro-définitions qu'il est autorisé d'utiliser sont ceux apportés par les **en-têtes** `stdio.h`, `stdlib.h` et `assert.h` ;
- il est autorisé de déclarer des **types** et de définir des **fonctions intermédiaires** si besoin est.

[Q1] Lister les éventuels effets de bords et donner la valeur de l'expression sous-jacente à chacune des instructions du morceau de programme suivant :

```
1 int a, b, *p;
2 a = b = 2;
3 b = a = b + 1;
4 a = a >= 3 ? 1 : a;

5 p = (int *) malloc(sizeof(int) * 64);
6 p[(a + b) << 1] = 32;
7 printf("%d\n", p[1 << b]);
8 free(p);
```

[Q2] En supposant que `y` est un identificateur de variable de type `float*`, donner le type de l'expression `*(&y + 1) - 1`. Justifier la réponse.

[Q3] En supposant que `x` est un identificateur de variable de type `int**` et `y` un identificateur de variable de type `int`, écrire une instruction dans laquelle la sous-expression `y` apparaît en tant que valeur droite et la sous-expression `*(x + 1)` en tant que valeur gauche.

[Q4] En supposant que `f` est un identificateur de fonction à type de retour `char*` et de signature `(int, char*)`, dire si l'expression `*f(f(16, NULL)[2], "1000") - '8'` est correctement typée. Si c'est le cas, donner son type. Justifier les réponses.

[Q5] Soit un projet constitué de quatre modules `A`, `B`, `C`, `D` et d'un module principal `Main`. Il figure dans ce projet les inclusions suivantes :

- `Main.c` inclut `A.h` et `B.h`
- `B.c` inclut `A.h`
- `C.c` inclut `B.h`
- `A.h` inclut `B.h` et `C.h`
- `C.h` inclut `B.h`

Tracer le graphe d'inclusions (étendues) du projet.

[Q6] Expliquer si le projet contient une ou plusieurs incohérences

[Q7] Écrire un Makefile simple pour compiler le projet.

[Q8] Après une compilation (qu'on supposera bien déroulée), on modifie le fichier **A.h**, donner la suite d'instructions de compilation suffisantes pour re-compiler le programme.

[Q9] Définir un type **QuatreCouleurs** permettant de représenter l'état d'un stylo quatre couleurs. Pour chacune des quatre couleurs (noir, rouge, vert, bleu), on souhaite représenter la quantité d'encre restante (de 0 pour vide, à 100 pour plein), et on doit également représenter si une pointe est sortie ainsi que sa couleur.

[Q10] Écrire une fonction **transformer_titre** paramétrée par une chaîne de caractère **chaîne** qui modifie **chaîne** afin que chaque mot commence par une majuscule suivi de minuscules. Un mot est une suite de lettres de l'alphabet délimitée par n'importe quel autre caractère (chiffre, ponctuation, espace...) Par exemple, la chaîne de caractères "**ceCi_n'eSt_PAS_rai5onN4ble**" devient "**Ceci_N'Est_Pas_Rai5onn4Ble**".

[Q11] Écrire une fonction **melange** paramétrée par un tableau de chaînes de caractères et renvoie une nouvelle chaîne de caractères construite en prenant successivement les premiers caractères de chaque chaîne, puis les deuxièmes caractères et ainsi de suite jusqu'à ce qu'il n'y ait plus de lettre. Par exemple, le résultat sur le tableau de chaînes de caractères {"**aeik**", "**bf**", "**cg**", "", "**dhj**"} est la chaîne "**abcdefghijk**".

[Q12] Écrire une fonction **compte_lettre_a** à gestion d'erreur paramétrée par un nom de fichier **nom** et qui renvoie le nombre d'occurrences de la lettre '**a**' dans le fichier.

[Q13] Écrire une fonction **combine_tabs** paramétrée par deux tableaux de même taille d'entiers de type **int** et un pointeur sur une fonction **f** à type de retour **int** et à deux paramètres de type **int**. La fonction renvoie un nouveau tableau de la même taille que ses arguments et dont l'élément d'indice **i** est le résultat de l'application de **f** aux deux éléments d'indices **i** des tableaux en paramètre.

[Q14] Écrire une fonction **equiv_rotation** paramétrée par deux nombres de 64 bits (non signés) et renvoie **1** si les bits du premier nombre sont les mêmes que ceux du second à rotation près et renvoie **0** sinon. Par exemple sur 8 bits, les nombres **01110010** et **01001110** sont les mêmes par une rotation de 3 bits vers la droite : **01110010** -> **01001110**

[Q15] Écrire une fonction **plus_long_suffixe** paramétrée par deux nombres de 64 bits (non signés) et qui renvoie la taille du plus long suffixe commun dans l'écriture binaire des deux nombres. Par exemple, les nombres **0...0100110** et **0...0101110** ont comme plus grand suffixe commun les 3 bits **110**.

[Q16] Donner en justifiant les erreurs présentes dans cet extrait de code de la fonction **main** :

```
1 if (argc == 4 && argv[1] == "compare") {
2     int different = (strlen(argv[2]) != strlen(argv[3]));
3     for (int i = 0; i < strlen(argv[2]) && i < strlen(argv[3]); i++) {
4         if (argv[2][i] != argv[3][i])
5             different = 1;
6         break;
7     }
8 }
9 if (different) printf("%s et %s sont différents\n", argv[2], argv[3]);
```