

Nom :
Prénom :
Groupe :

## Eléments de programmation 2 – 1i002

Partiel du 10 mars 2017 (1h30)

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 40 points n'a qu'une valeur indicative. Avant de commencer, vérifiez que votre copie contient **12 pages** avec 12 questions.*

### 1 Cours (8 pts)

#### Question 1 (3 points)

Pour chacune des déclarations suivantes, dire si elle est juste (*i.e.* si elle ne provoque, ni erreur, ni avertissement à la compilation) ou non. Si non, expliquer pourquoi elle est syntaxiquement fausse ou provoque un avertissement à la compilation et proposer une solution pour la corriger. Une réponse sans proposition de correction sera considérée comme fausse.

```
1  int x = 2;
2  char chaine[] = "Hello";
3  float z = 2,3;
4  int t, int v;
5  float tab[3] = {1.0, 2.0};
6  #define N 3;
```

Réponse :

**Question 2** (3 points)

Donner le nombre d'itérations minimum qu'il est possible de faire en C avec un `for`, un `while` et un `do-while`. Donner un exemple à chaque fois pour illustrer votre réponse.

**Réponse :**

**Question 3** (2 points)

Décrire de manière précise le résultat de l'exécution de ce programme suivant en justifiant votre réponse.

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

```
int main () {  
    int i;  
    int f[6]={1, 2};  
    for (i=2; i<=6; i++) {  
        f[i] = f[i-1]+f[i-2];  
        printf("%d_", f[i]);  
    }  
  
    return 0;  
}
```

Réponse :

## 2 Exercices (17 pts)

### Question 4 (3 points)

Une fraction  $A/B$  est définie par 2 entiers  $A$  et  $B$ . Nous considérerons ici que  $A$  et  $B$  sont deux entiers non nuls.

Écrire un programme qui, à partir de deux fractions  $A/B$  et  $C/D$ , calcule la multiplication de ces deux fractions et affiche à l'écran le résultat de ce calcul. Les valeurs de  $A$ ,  $B$ ,  $C$  et  $D$  sont définies par des directives `#define`. Par exemple, pour  $A = 2$ ,  $B = 5$ ,  $C = 3$  et  $D = 4$ , le programme doit afficher : “Le produit de  $2/5$  par  $3/4$  est  $6/20$ ”. Le code doit bien sûr être indépendant des valeurs de  $A$ ,  $B$ ,  $C$  et  $D$ .

Réponse :

### Question 5 (3 points)

Donald sait planter les choux. Il plante trois choux dans une première rangée. Il augmente de deux choux d'une rangée à l'autre. Écrire un programme C qui affiche le nombre total de choux plantés par

Donald s'il fait en tout 49 rangées. Vous utiliserez des directives **#define** pour définir les constantes du programme.

**Réponse :**

**Question 6** (4 points)

Écrire un programme qui calcule la fonction  $w(n) = \frac{1*3*\dots*(2n+1)}{2*4*\dots*(2n)}$  avec  $n \geq 1$ . Par exemple  $w(3) = 2.1875$  ( $= \frac{1*3*5*7}{2*4*6} = \frac{105}{48}$ ). Le programme demandera à l'utilisateur d'entrer la valeur de  $n$  au clavier en vérifiant qu'il entre bien une valeur strictement positive (*i.e.* saisie avec vérification, donc on lui demande de saisir une valeur jusqu'à ce que celle-ci soit valide), puis affichera à l'écran les valeurs de  $n$  et de la fonction calculée.

NOM :

Prénom :

Groupe :

---

**Réponse :**

**Question 7** (3 points)

Écrire un programme qui remplit un tableau de  $N$  entiers en tirant aléatoirement le contenu de chaque case parmi les valeurs 2, 3 et 5.  $N$  sera défini par un **#define**. Le programme affichera ensuite le nombre de 2, de 3 et de 5 contenus dans ce tableau.

**Réponse :**

**Question 8** (4 points)

On considère des chaînes de caractères formées des deux caractères parenthèses ouvrante et fermante : ( et ). On veut écrire un programme qui détermine si une chaîne de caractères est bien parenthésée comme : ( ( ) ( ) ) ou ( ( ( ) ) ) ou encore ( ) ( ) ( ( ( ) ) ) ou mal parenthésée comme : ( ) ) ou ) ( ou encore ( ( ( ) ) ) ). Dans le cas d'une chaîne de caractères mal parenthésée, il y a au moins une parenthèse ouvrante qui ne correspond à aucune parenthèse fermante ou une parenthèse fermante qui ne correspond à aucune parenthèse ouvrante.

Pour déterminer cela, on parcourt la chaîne de caractères de gauche à droite et on compte les parenthèses. On ajoute +1 chaque fois que l'on trouve une parenthèse ouvrante et -1 chaque fois que l'on trouve une parenthèse fermante. Le programme doit s'arrêter dès que le compteur devient négatif ou lorsque l'on atteint la fin de la chaîne de caractères.

Votre programme écrira à la fin la chaîne de caractères et si cette chaîne de caractères est bien ou mal parenthésée. Les caractères autres que les parenthèses seront ignorés par votre programme.

Réponse :

### 3 Manipulation de boucles (7 pts)

#### Question 9 (3 points)

Soient deux tableaux de nombres de type **float** de tailles respectives  $N$  et  $M$  avec  $N$  et  $M \geq 1$ . On veut calculer la somme de l'ensemble des valeurs obtenues en multipliant chaque élément du premier tableau par chaque élément du deuxième tableau.

Écrire le programme qui affiche ce total. Vous utiliserez une itération de type **for** et vous initialiserez les deux tableaux avec les valeurs de votre choix.

**Réponse :**

**Question 10** (4 points)

On fait une simulation de jeu de combat entre deux joueurs. Chacun des deux joueurs dispose d'un nombre de points de vie initial  $PV_{INIT}$ . À chaque tour de jeu, chacun des joueurs inflige à l'autre un nombre de points de dégâts tiré aléatoirement par le lancer d'un dé à 6 faces : ces points de dégâts sont retirés des points de vie de son adversaire.

Le combat se termine lorsqu'au moins un des deux joueurs a un nombre de points de vie inférieur ou égal à zéro (il peut y avoir match nul).

Écrire un programme qui effectue un combat où, à chaque tour, on affiche le numéro du tour, le nombre de points de vie des deux joueurs au début du tour et le nombre de points de dégâts infligés par chaque joueur à son adversaire. À la fin du dernier tour, le programme affiche le joueur gagnant et le nombre de tours joués.



NOM :

Prénom :

Groupe :

---

**Réponse :**

## 4 Problème (8 pts)

### Question 11 (4 points)

Un carré magique d'ordre  $n$  est composé de  $n^2$  entiers strictement positifs rangés dans un tableau de  $n \times n$  cases. Les valeurs sont placées de sorte que les sommes des nombres sur chaque ligne, dans

NOM :

Prénom :

Groupe :

---

chaque colonne et sur chaque diagonale soient toutes égales. La valeur unique de ces sommes est appelée constante magique.

Écrire un programme qui teste si un tableau est un carré magique. Si oui, le programme affiche la constante magique. Sinon, dès qu'il détecte que le tableau n'est pas un carré magique, il doit se terminer en affichant un message.

NOM :

Prénom :

Groupe :

---

**Réponse :**

**Question 12** (4 points)

On souhaite écrire un programme qui fait l’affichage graphique des valeurs contenues dans un carré (magique ou non) en utilisant la bibliothèque CINI. On se limitera à des carrés contenant des entiers inférieurs à 100. On affectera à l’affichage de chaque valeur un rectangle de taille 30 pixels de large par 25 pixels de haut.

Pour afficher le contenu d’une case, on utilisera la fonction `CINI_draw_string` qui affiche une chaîne de caractères. Pour convertir un entier en chaîne de caractères, on utilisera la fonction `sprintf` de la bibliothèque `stdio.h` qui copie dans un tableau de  $k + 1$  caractères déclaré préalablement la chaîne correspondant à un nombre d’au plus  $k$  chiffres. Par exemple :

```
char str[3];  
int n = 99;  
sprintf(str, "%d", n);
```

copie la chaîne "99" dans `str`.

La taille de la fenêtre d’affichage correspondra à la taille du carré plus une bordure de `MARGE` pixels autour.

Il n’est pas nécessaire de dessiner le quadrillage mais juste les valeurs du carré.

**Réponse :**