

IF2A – Examen médian

Durée : 1h30

Seule une feuille A4 manuscrite nominative recto-verso est autorisée comme document.

Tous les autres documents ou tout autre dispositif électronique sont interdits.

Barème donné à titre indicatif (± 1)

Remarque : vous pouvez rédiger les algorithmes demandés en pseudo-code ou en langage C. Le prototype complet des sous-programmes doit être fourni. Il n'est pas nécessaire de préciser les « includes » pour les exercices en C.

Exercice 1 – Méthode de Héron (7 points)

La méthode de Héron est une méthode permettant de calculer la racine carrée d'un nombre strictement positif de façon approximée.

Soit la suite suivante définie par récurrence :

$$\forall n \in \mathbb{N}, x_{n+1} = \frac{x_n + \frac{a}{x_n}}{2}$$

Pour tout $a \in \mathbb{R}, a > 0$ et $x_0 \in \mathbb{R}, x_0 > 0$, cette suite converge vers \sqrt{a} (racine positive), ou écrit autrement $\lim_{n \rightarrow \infty} x_n = \sqrt{a}$.

Écrire le sous-programme qui, étant donné a réel strictement positif, x_0 réel strictement positif, et un seuil e réel strictement positif, calcule et retourne la valeur de \sqrt{a} en utilisant la méthode de Héron. On s'arrêtera d'itérer dès lors qu'on sera assez proche de la valeur recherchée, c'est-à-dire si $|x_n \times x_n - a| < e$. Si les paramètres effectifs de la fonction sont invalides ($a \leq 0$, $x_0 \leq 0$ ou $e \leq 0$), la fonction retournera la valeur -1.

Naturellement, il est interdit pour cet exercice d'appeler la fonction `sqrt()` du langage C ou toute autre fonction issue de la bibliothèque `math.h`. Vous pouvez définir autant de sous-programmes que nécessaire.

Exercice 2 – Pesée de pièces (7 points)



Nous disposons de 12 pièces, numérotées de 0 à 11. Toutes les pièces font le même poids, à l'exception d'une seule légèrement plus lourde que les autres et que nous souhaitons trouver.

Pour nous aider nous disposons de la fonction suivante :

```
int pesee(int N1, int P1[], int N2, int P2[])
```

Cette fonction de pesée, équivalente à une balance, considère deux tableaux d'entiers contenant des indices de pièces, et retourne -1 si les pièces représentées dans le tableau P1 de taille N1 sont plus lourdes que les pièces du tableau P2 de taille N2, 1 si les pièces du tableau P2 sont plus lourdes que celles du tableau P1, et 0 si les pièces des deux tableaux font le même poids.

Écrire un algorithme permettant de déterminer le numéro de la pièce plus lourde que les autres, en appelant la fonction `pesee()` au maximum 3 fois au cours du déroulement de l'algorithme

(l'algorithme peut mentionner plus de 3 fois la fonction *pesee()*, mais aucun chemin d'exécution ne doit mener à appeler cette fonction plus de 3 fois).

Exercice 3 – Fonctions et paramètres (6 points)

Soit le code C suivant :

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

int f1(int a, int b) {
    a = a + 1;
    b = b * 2;
    printf("%d %d\n", a, b);
    return b;
}

void f2(int b, int a, int n) {
    do {
        f1(a, b);
        n = n + 1;
    } while (n < 3);

    printf("%d %d %d\n", a, b, n);
}

int main() {
    int a = 6;
    int b = 3;
    int n = 8;

    a = f1(a, b);
    f2(a, b, n);

    printf("%d %d %d", a, b, n);

    return EXIT_SUCCESS;
}
```

Ecrire ce qui sera affiché à l'écran (le plus précisément possible, en intégrant espaces et retours à la ligne).