

Programmation Impérative

28/2/2024

Nom _____
Prénom _____
numéro _____
mail _____

Les questions sont à traiter dans l'ordre qui vous plaira mais vous n'aurez peut-être pas le temps de tout faire...

Ce sujet doit être remis avec votre copie, une version imprimable sera donnée sur la page du cours pour que vous puissiez retravailler les questions.

Quand vous simulez le fonctionnement de l'ordinateur lors de l'exécution d'une fonction, vous devez donner l'adresse et les contenus successifs de la mémoire pour chacune des variables utilisées. Pour les fonctions récursives tous les appels doivent être spécifiés.

A) Simples

1. Soit la fonction `nope`, cf. Table 1, donnez toutes les étapes de l'exécution de cette fonction.
2. Soit la fonction suivante qui prend deux entiers naturels en argument, écrivez une fonction en C qui calcule ses valeurs.

$$f(0, a) = a + 1$$

$$f(a, 0) = f(a - 1, 1)$$

$$f(a, b) = f(a - 1, f(a, b - 1))$$

3. Soit la fonction `g`, donnez une forme récursive de cette fonction.

$$g(0) = 0$$

```
void hope (float * a, float * b) {
    while (* a != *b) {
        if (*a > *b) {
            *a -= *b;
        }
        else {
            *b -= *a;
        }
    }
}

void nope () {
    float x, y;
    x = 120.0;
    y = 45.0;
    hope (&x, &y);
    printf(" x == %f\n", x);
    x = 125.0;
    y = 21.0;
    hope (&x, &y);
    printf(" x == %f\n", x);
}
```

TABLE 1 – Fonctions `hope` et `nope`

```
int lav (int a, int b) {
    if ( b == 0)
        return -1;
    if (a < b)
        return 0;
    return lav (a-b, b) + 1;
}
```

TABLE 2 – Fonction `lav`

$$g(1) = 1$$

$$g(n) = g(n - 1) + g(n - 2) + 2$$

4. Donnez une forme itérative de cette fonction `g`.
5. Écrivez une fonction itérative qui renvoie le reste de la division de ses deux arguments.
6. Vous trouverez en table 2 une fonction `lav`, donnez sa forme récursive terminale.

B) Bac

Vous trouverez dans la Table 3 une structure de vecteur qui correspond à des coefficients, des notes sur 20 et des notes effectives (note sur vingt \times coefficient).

Dans la suite du sujet, nous noterons **note sur vingt**, la note initiale et **note**, la note une fois que le coefficient a été appliqué.

Écrivez les fonctions suivantes avec cette structure.

1. Écrivez une fonction qui affiche tous les champs d'un vecteur de type **vecnotes_t**.
2. Écrivez une fonction qui, pour tout le vecteur, remplit le dernier champ de la structure (la **note** finale).
3. Écrivez une fonction qui renvoie la somme des coefficients du vecteur.
4. Écrivez une fonction qui renvoie la moyenne pondérée (avec coefficients, comme au Bac) des notes du vecteur.
5. Écrivez une fonction qui renvoie la meilleure **note sur vingt**.
6. Écrivez une fonction qui range le vecteur dans l'ordre des **notes sur vingt** décroissantes.

C) Tableaux

Soit la structure de tableau, **table 4**.

1. Donnez une fonction qui fait la somme de toutes les valeurs de ce tableau.

```
typedef struct notes {
    int coef;
    float notesurvingt;
    float note;
} notes_t;

typedef struct vectnotes {
    int nbn;
    notes_t vec[100];
} vecnotes_t;
```

TABLE 3 – Structure de vecteur

```
typedef struct table_alt {
    int imax;
    int jmax;
    float t[100][100];
} table_t;
```

TABLE 4 – Structure de tableau de nombres

2. Donnez une fonction qui fait la somme de toutes les valeurs de ce tableau inférieures à un x donné.
3. Donnez une fonction qui renvoie la plus grande des valeurs de ce tableau.