

Partiel Informatique CIR I

6 janvier 2021

Les parties I et II doivent être rendues sur des copies séparées.

Les documents et équipements électroniques ne sont pas autorisés.

Partie I – 10 points, dont 1 de présentation.

Pour chacune de ces fonctions, proposez une implémentation en langage C.

- 1) Ecrire une fonction booléenne **est_premier(int n)** qui renvoie **vrai** quand la valeur entière passée en argument est un nombre premier, et **faux** dans le cas contraire.

Un nombre est dit premier quand il admet pour uniques diviseurs distincts 1 et lui-même.

Optimiser la boucle de recherche d'un diviseur. La fonction racine carrée est `sqrt()`.

- 2) Ecrire une fonction qui recherche le prochain nombre premier à partir du nombre passé en argument de la fonction.

Par exemple :

Après l'exécution de **prems = premier_suivant(7)** , **prems** vaut 7

Après l'exécution de **prems = premier_suivant(8)** , **prems** vaut 11

- 3) Ecrire une fonction qui affiche la décomposition d'un nombre entier non nul en un produit de facteurs premiers.

Par exemple : après l'exécution de `decomposition(10164);`

la console affiche :

10164 = 2 x 2 x 3 x 7 x 11 x 11

Partie 2 – 10 points, dont 1 de présentation.

1) Expliquez ce que fait la fonction compte dans le cas général.

```
#include <stdbool.h>

bool compte(int tab[], int tailleTab) {
    int compte1 = 0;
    int compte2 = 0;
    for (int i = 0; i < tailleTab; i++) {
        if (tab[i] % 2 == 0) {
            compte1++;
        }
        else {
            compte2++;
        }
    }
    return (compte1 > compte2);
}
```

Soit le tableau défini ci-dessous :

```
int carac [10] = {1, 4, 62, 43, 92, 62, 73, 24, 7, 85};
```

Pendant l'appel de fonction correspondant à la ligne suivante :

```
compte(carac, 8) ;
```

- 2) Quelles seront les valeurs de compte1 et de compte2 avant de sortir de la fonction ?
- 3) Quels seront le type et la valeur retournés par cette fonction ?
- 4) La variable compte2 n'est pas nécessaire ici, comment pourrait-on modifier le code de la fonction compte pour se passer de cette deuxième variable locale ?
- 5) Écrivez une fonction qui prend trois tableaux d'entiers en paramètre t1, t2, t3 et un entier représentant la taille utilisée du tableau t1. Cette fonction doit remplir t2 avec tous les nombres pairs contenus dans t1 et remplir t3 avec tous les nombres impairs contenus dans t1. L'ordre d'apparition des chiffres doit être conservé.
Vous pouvez vous inspirer de la fonction compte initiale.

Exemple :

- entrée : t1 contient [1,3,5,6,8,9] et taille vaut 6
- En sortie :
 - t1 est inchangé
 - t2 contient [6,8]
 - t3 contient [1,3,5,9]

Soit la fonction `modif` déclarée ci-dessous (il n'est pas nécessaire de connaître son fonctionnement):

```
void modif(int* b);
```

- 6) Écrivez une fonction qui prend un tableau d'entiers ainsi qu'un entier représentant sa taille utilisée et appelle la fonction `modif` sur toutes les cases du tableau qui contiennent une valeur négative.

Questions indépendantes

- 7) Rassemblez les expressions suivantes par groupes d'équivalence.
(Mettez ensemble les expressions qui ont le même sens. Il peut y avoir des groupes contenant un seul élément.)

- a) `!x`
- b) `x > 10`
- c) `!(x == 0)`
- d) `x != 0`
- e) `x >= 0`
- f) `-x <= 0`
- g) `x-10 < -10`
- h) `!(x+10 < 10)`

- 8) Qu'affiche le programme suivant :

```
int main() {
    int x = 10;
    int* adx = &x;
    int y = x;

    y++;
    *adx += 2;
    printf("%d,%d,%d", x, *adx, y);
}
```