

Examen de Langage C

Durée : 1h30

Aucun document autorisé

Mobiles interdits Toutes les fonctions que vous écrirez doivent être clairement commentées avec des affirmations significatives (antécédents, conséquents, invariants). Vous prendrez soin de définir les bons paramètres et les bons types des données manipulées. Pensez à définir des fonctions auxiliaires si cela est nécessaire.

Question 1 : Maximum

- 1. Écrivez la fonction `max3` qui renvoie le maximum de 3 entiers passés en paramètres. Cette fonction ne devra faire que 2 comparaisons.

```
/*
 * Antécédent : a, b et c trois entiers quelconques
 * Rôle : renvoie le maximum de a, b et c
 */
int max3(const int a, const int b, const int c) {
    int max = a > b ? a : b;
    // max = maximum(a,b)
    return c > max ? c : max;
}
```

Ci-dessous, une seconde version structurée qui est préférable :

```
/*
 * Antécédent : a et b deux entiers quelconques
 * Rôle : renvoie le maximum de a et b
 */
int max2(const int a, const int b) {
    return a > b ? a : b;
}
/*
 * Antécédent : a, b et c trois entiers quelconques
 * Rôle : renvoie le maximum de a, b et c
 */
int max3(const int a, const int b, const int c) {
    return max2(max2(a,b), c);
}
```

Question 2 : Nombres parfaits

- 2. Écrivez la fonction `sommeDesDiviseurs` qui renvoie la somme des diviseurs stricts d'un nombre entier positif (> 0) passé en paramètre. Par exemple, la somme des diviseurs stricts de 78 est égale à 90.

```
/*
 * Antécédent : n > 0
 * Rôle : renvoie la somme des diviseurs stricts de n
 */
int sommeDesDiviseurs(const int n) {
    int s = 0;
    // inutile de tester les diviseurs > à n/2 !
    for (int i = n/2 ; i > 0 ; i--)
        if ((n%i) == 0)
            // i est un diviseur
            s += i;
    //
    return s;
}
```

- 3. Un nombre est dit *parfait* s'il est égal à la somme de ses diviseurs stricts. Par exemple, $28 = 1 + 2 + 4 + 7 + 14$. Écrivez la fonction booléenne `estParfait` qui teste si son paramètre (> 0) est un nombre parfait ou non.

```
/*
 * antécédent : n > 0
 * rôle : renvoie 1 si n est un nombre parfait, 0 sinon
 */
int estParfait(const int n) {
    return sommeDesDiviseurs(n) == n;
}
```

- 4. Écrivez la fonction `main` qui lit sur l'entrée standard un entier n (> 0) et qui écrit sur la sortie standard tous les nombres parfaits sur l'intervalle $[1; n]$.

```
int main(void) {
    int n;
    scanf("%d", &n);
    assert(n > 0);
    for (int i = 1; i <= n; i++)
        if (estParfait(i))
            printf("%d\n", i);
    //
    return EXIT_SUCCESS;
}
```

Question 3 : Nombres binaires

- 5. Écrivez la fonction `lireBinaire` qui lit sur l'entrée standard une suite (non vide) de caractères, des '0' et des '1', à partir de laquelle elle calcule et renvoie sa conversion sous forme d'un entier non signé (**unsigned int**). Par exemple, si on a sur l'entrée standard la suite de caractères '1' '1' '1' '0' '1', la fonction `lireBinaire` renverra l'entier non signé 29.

```
/*
 * Antécédent : une suite de 0/1 est disponible sur l'E/S
 * Rôle : renvoie la conversion en entier non signé de la suite de 0/1
 */
unsigned int lireBinaire(void) {
    char c;
    int onContinue=1, n=0;
    do {
        // lire le prochain caractère sur l'E/S
        scanf("%c", &c);
        if (c=='0' || c=='1') {
            // vérifier si pas d'overflow (i.e. n <= (UINT_MAX-(c-'0'))/2)
            assert(n<=(UINT_MAX-(c-'0'))/2);
            // ajouter le chiffre à n
            n = (n<<1) + c-'0';
        }
        else
            // on a fini de lire la suite de chiffres binaires
            onContinue=0;
    } while (onContinue);
    //
    return n;
}
```

Question 4 : Prêt bancaire

Une banque accorde à ses clients un prêt si la somme des intérêts acquis sur un capital placé à un taux intérêt annuel donné est supérieur à un certain seuil.

- 6. Écrivez la fonction `nombreAnnees` qui renvoie le nombre d'années (**int**) qu'un capital placé à un taux d'intérêt annuel donné doit rester sur le compte bancaire pour que les intérêts cumulés dépassent le seuil et que le détenteur du compte puisse obtenir son prêt. Par exemple, si le capital initial est 5000.0 euros avec un taux d'intérêt annuel de 1.5% et que le seuil est 1000.0 euros, il faudra attendre 11 ans!

```
/*
 * antécédent : capital initial placé>0,
 *               taux d'intérêt>0,
 *               seuil>0 à partir duquel on accorde le prêt
 * rôle : renvoie le nombre d'années pour que les intérêts atteignent
 *         le seuil
 */
```

```
*/
int nbAnnees(double capital, double taux, const double seuil) {
    int annees = 0;
    double interetsCumules=0.0;
    taux = taux/100;

    do {
        // intérêtsCumulés < seuil;
        double interetsAnnuels = capital*taux;
        capital+=interetsAnnuels;
        interetsCumules+=interetsAnnuels;
        annees++;
        // capital = capital initial + intérêts cumulés sur années
    }
    while (interetsCumules<seuil);
    // intérêts cumulés>=seuil sur années
    return annees;
}
```