

Examen de Info C

Durée : 1h

Aucun document autorisé

Mobiles interdits

Nom :

Prénom :

- 1. Avec la notation de pointeur, écrivez en C la fonction `somme` qui prend en paramètre une chaîne de caractères qui représente un entier naturel et qui renvoie la somme de ses chiffres. Si la chaîne de caractères ne contient pas que des chiffres, la fonction renverra `-1`. Vous pouvez utiliser la fonction `isdigit`, à l'exclusion de toute autre.

```
/*
 * Antécédent :
 * Rôle : renvoie la somme des chiffres contenus dans la
 * chaîne s. Renvoie -1 si s ne représente pas un naturel
 */
int somme(const char *s) {
    if (*s=='\0') return -1;
    // s n'est pas vide et contient au moins un caractère
    int som = 0;
    do {
        if (!isdigit(*s))
            return -1;
        // le caractère courant est un chiffre => on le somme
        som += *s - '0';
    } while (++s);
    // on a parcouru tous les chiffres de la chaîne s
    return som;
}
```

- 2. Écrivez en C la procédure `compter` qui compte le nombre d'entiers pairs, `pa`, et impairs `imp` contenus dans un tableau `t` de `n` entiers. Le tableau et le nombre d'entiers qu'il contient sont des paramètres « données », et les entiers `pa`, et `imp` sont des paramètres « résultats ».

```
/*
 * Antécédent : t un tableau entiers
 * n nombre d'entiers contenus dans t
 * Conséquent : pa nombre d'entiers pairs de t
 * imp nombre d'entiers impairs de t
 */
void compter(const int *t, const int n, int *pa, int *imp) {
    *pa = 0;
    *imp = 0;
    for (int i=0; i<n; i++)
        if ((t[i]&1)==0)
            (*pa)++;
        else
            (*imp)++;
}
```

- 3. Écrivez en C la fonction `main` qui déclare un tableau contenant les entiers `-10, 8, 9, 6, 12`, qui appelle la procédure `compter` et qui écrit sur la sortie standard le nombre d'entiers pairs et impairs.

```
int main(void) {
    int pairs, impairs;
    int t[] = {-10, 8, 9, 6, 12};
    compter(t, 5, &pairs, &impairs);
    printf("%d %d\n", pairs, impairs);

    return EXIT_SUCCESS;
}
```

- 4. Écrivez en C, un programme qui prend ses données à partir des paramètres programmes. Le programme pourra être utilisé de deux façons différentes :
— soit avec une suite non vide d'entiers naturels. Le programme affichera leur somme;
— soit avec l'unique option `-h`. Le programme affichera un message expliquant ce que fait le programme.
Voici un exemple d'utilisation du programme que nous appellerons `somme` :

```
$ somme 4 20 9 10
```

```
43
```

```
$ somme -h
```

Ce programme affiche la somme de ses paramètres

Vous ferez les contrôles nécessaires de vérification de la validité de l'utilisation du programme. Vous utiliserez exclusivement une notation de pointeur. Sans l'écrire, vous pourrez utiliser la fonction `estUnEntier` vue en TD, ainsi que la fonction `atoi`.

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

int main(int argc, char *argv[]) {
    if (argc==1) {
        fprintf(stderr, "Usage : \tsomme -h\n\tsomme int...\n");
        exit(EXIT_SUCCESS);
    }
    // il existe au moins un paramètre
    if (argc==2 && strcmp(++argv, "-h")==0)
        printf("Ce programme affiche...\n");
    else {
        int somme=0;
        while (argc-->1) {
            if (!estUnEntier(++argv)) {
                fprintf(stderr, "Entier attendu\n");
                exit(EXIT_SUCCESS);
            }
            // on a bien un entier => on l'ajoute
            somme+=atoi(*argv);
        }
        // afficher le résultat
        printf("%d\n", somme);
    }
    return EXIT_SUCCESS;
}
```

