

Algorithmique et Programmation

TP3 *Manipulation de grands entiers*

FICHIERS À RENDRE SUR MARKUS :

- main.c
- utilitaires.c et utilitaires.h
- operations.c et operations.h
- rapport_TP3.pdf

1 Introduction

L'objectif du TP est de pouvoir effectuer des calculs sur des nombres entiers comportant un nombre de chiffres plus important que les types entiers standard. Chaque entier sera représenté par l'ensemble de ses chiffres (en base 10), le nombre de chiffres étant limité par la constante `MAXCHIFFRES`. On définit les types suivants :

types
 t_Chiffres : vecteur d'entiers
 t_EntierLong : enregistrement
 booléen : négatif
 t_Chiffres : chiffres
fin_enregistrement

Pour un entier x , **négatif** vaut *vrai* si le nombre est négatif. Les valeurs du tableau **chiffres** sont les valeurs des $(a_i) \in \{0..9\}$ (rangés de gauche à droite) tels que

$$|x| = \sum_{i=0}^{MAXCHIFFRES-1} a_i 10^i$$

Ainsi, le nombre 357635735 sera représenté dans le tableau **chiffres** de la façon suivante :

indice	0	1	2	3	4	5	6	7	8	9	...	19
chiffre	5	3	7	5	3	6	7	5	3	0	...	0

2 Fichiers à utiliser (disponibles sur le serveur pédagogique)

2.1 Types

Le fichier `entierlong.h` contient la définition du type structuré `t_EntierLong`. Vous devez inclure (`#include`) ce fichier dans les fichiers sources (`.c`) dans lesquels vous voulez utiliser le type `t_EntierLong`. On rappelle que les tableaux en C commencent à l'indice 0.

2.2 Fonctions d’affichage et saisie d’un entier long

Spécification

Fonction `ecrireEntierLong(n)`

Paramètre : `t_EntierLong n`

Résultat : aucun // l’entier long `n` sera affiché à l’écran

Spécification

Fonction `n ← lireEntierLong()`

Paramètre : aucun // un entier long sera saisi au clavier (avec un nombre de chiffres inférieur ou égal à `MAXCHIFFRES`)

Résultat : `t_EntierLong n` // l’entier long saisi au clavier sera rangé dans `n`

Vous disposez sur le serveur pédagogique des fichiers suivants (que vous ne devez pas modifier) :

- Le fichier `lit_ecrit.c` contient la définition de ces deux fonctions.
- Le fichier `lit_ecrit.h` contient la déclaration de ces deux fonctions. Vous devez inclure ce fichier (`#include`) dans les fichiers sources dans lesquels vous voulez utiliser ces fonctions.

3 Travail à réaliser (lisez jusqu’au bout)

Concevez et programmez en C les fonctions suivantes :

- Dans un fichier `utilitaires.c` :
 - conversion d’un entier standard en entier long ;
 - test d’égalité de deux entiers longs ;
 - comparaison en valeur absolue : pour deux entiers longs $n1$ et $n2$, a-t-on $|n1| \leq |n2|$?
- Dans un fichier `utilitaires.h` : les déclarations des fonctions contenues dans `utilitaires.c`
- Dans un fichier `operations.c` :
 - addition $n1 + n2$ de deux entiers longs $n1$ et $n2$ **de même signe** ;
 - soustraction $n1 - n2$ de deux entiers longs $n1$ et $n2$ de même signe et tels que $|n1| \geq |n2|$;
 - addition et soustraction de deux entiers longs de signes quelconques en faisant appel aux fonctions précédentes ;
 - *Pour les plus rapides* : division euclidienne de deux entiers longs positifs.
- Dans un fichier `operations.h` : les déclarations des fonctions contenues dans `operations.c`

Simultanément, dans un fichier `main.c`, écrivez une fonction principale que vous pourrez modifier progressivement pour tester au fur et à mesure les différentes fonctions programmées.

4 Compte-rendu

Votre rapport doit contenir les éléments suivants :

- Pour chaque fonction :
 - la spécification ;
 - une description brève du principe mis en œuvre ;
 - des jeux d’essais simples illustrant son bon fonctionnement ;
- Les algorithmes des fonctions ”addition de même signe” et ”addition quelconque” ;
- Les jeux d’essais correspondant au calcul de la suite de Fibonacci :

$$u_0 = 0, u_1 = 1, \forall n \geq 2, u_n = u_{n-1} + u_{n-2}.$$

Pour plusieurs valeurs de n , dont une très élevée de l’ordre de 20 000, calculer les n premiers termes de la suite, afficher les 3 derniers.