Licence d'informatique
Module de C/Unix

Projet C

`bigint.a` — Une librairie de grands entiers

Jean-Luc LEVAIRE

Décembre 1999

Une librairie de grands entiers

Le but de cette première partie est de mettre au point une librairie travaillant sur de très grands entiers (plusieurs milliers de chiffres décimaux) non signés. Le principe est de simplement stocker un *bigint* (très grand entier) dans un tableau d'entiers de type *unsigned int*. La taille des bigints utilisés étant dynamique (à priori non connue à la compilation), ces tableaux seront alloués dynamiquement, et donc les tailles des bigints devront aussi être stockées . D'où le type *bigint* suivant :

```
typedef struct bigint {
  unsigned int size;   /* taille */
  unsigned int *value; /* valeur */
} bigint;
```

Un problème inhérent à ce type de librairie est le passage de la représentation en binaire des nombres à leur représentation en décimal. En effet, tous les chiffres binaires interviennent dans la valeur d'un des chiffres décimaux. Il est donc préférable de travailler en une base puissance de 10, et on choisira la plus grande puissance de 10 pouvant être contenue dans un élément du tableau (soit 10^9 pour un *unsigned int*). Le stockage se fera du poids faible (indice 0) au poids fort (dernier élément du tableau).

Exercice 1
Écrire les fonctions arithmétiques suivantes :

```
/* Operators */
extern int cmp(bigint a, bigint b);          /* Comparaison   */
extern bigint add(bigint a, bigint b);       /* Addition      */
extern bigint sub(bigint a, bigint b);       /* Soustrac-
tion   (a >= b)*/
extern bigint product(bigint a, bigint b);  /* Produit       */
                                             / * Division     */
extern void intdiv(bigint a, bigint b, bigint *quotient, bi-
gint *modulo);
extern bigint pow2n(unsigned int n);         /* 2 puissance n */

/* Input / Output */
extern void printbigint(bigint n);    /* Affichage sortie stan-
dard     */
extern char *biginttostr(bigint n);   /* Conversion bi-
gint vers chaine */
extern bigint *strtobigint(char *s); /* Conversion chaine vers bi-
gint */
                                      /* Conversion bi-
gint vers chaine */
```

```
                                                /* Deci-
males entre first et last */
extern char *biginttosubstr(bigint n, int first, int last);
```

`cmp` compare a et b, et retourne la position du premier *unsigned int* où a et b diffèrent. Le résultat est positif si $a > b$, négatif si $a < b$ et nul si $a == b$. Tous les calculs se feront élément de tableau par élément de tableau, et devront ainsi être faits en base 10^9. La taille des résultats est calculable à un élément près, et on supprimera celui-ci par réallocation si il est nul.

Application 1 : Nombres premiers de Mersenne

Les nombres de Mersenne M_n sont des nombres de la forme $2^n - 1$ où n est un nombre premier. On sait[1] de plus qu'un nombre de la forme $2^p - 1$ est premier s'il divise $S(p-1)$ avec $S(1) = 4$ et $S(k+1) = S(k)^2 - 2$ pour $k > 1$. Un nombre premier de Mersenne[2] est donc un nombre de Mersenne vérifiant le test de Lucas-Lehmer. Écrire un programme *mersenne n* qui calcule la valeur du nombre de Mersenne M_n et teste si il est premier. Notez qu'il suffit de calculer $S(k+1)\%M_n$ pendant le test de Lucas-Lehmer et de tester si le résultat final $S(n-1)\%M_n$ est nul.

Pour réaliser des tests, voici un tableau (tiré de 2) présentant quelques nombres premiers de Mersenne :

Nombre	Nombre de chiffres	Année	Auteur ou Ordinateur
M_{17}	6	1588	Cataldi
M_{19}	6	1588	Cataldi
M_{31}	10	1772	Euler
M_{127}	39	1876	Lucas
M_{521}	157	1952	SWAC
M_{607}	183	1952	SWAC
M_{4253}	1281	1961	IBM 7090
M_{21701}	6533	1978	Cyber 174
M_{44497}	13395	1979	Cray 1
M_{756839}	227832	1992	Cray 2
$M_{8972593}$	2098960	1999	Pentium II

Le prochain record à battre est un nombre premier de Mersenne à plus de 10 millions de chiffres décimaux : 100000 dollars à la clé. Bonne chance !

Application 2 : Calculette sur de grands entiers

Il s'agit de réaliser une calculette possédant une interface graphique sous X-Window, et travaillant sur de grands entiers grâce à votre librairie. Je vous ai écrit l'interface graphique représentant un clavier de calculette, avec une fenêtre contenant le résultat courant. Cette interface contient la fonction `main`, qui est une boucle infinie gérant les événements souris et clavier (entre autres). Vous devez essentiellement intervenir au niveau de la fonction `HandleTouch` qui est appelée lors d'un clic souris sur une touche du clavier. De même pour les fonctions `print`, `clear` et `quit`, appelées lors d'un clic sur le bouton correspondant. Vous trouverez tout ceci dans `~levaire/TPC/sample/projet9900/`. De plus amples explications se trouveront dans le fichier `README`.

[1] Test de Lucas-Lehmer.

[2] voir `http://www.mersenne.org` et *Pour la Science*, Octobre 1999, pp105–109.