

G4011105 – Programación I

Descripción

O obxectivo deste exercicio é implementar unha biblioteca en C para o traballo con números enteiros grandes.

A representación de números enteiros nun ordenador está limitada polo número de bits da representación. Non obstante hai certas operacións matemáticas de interese que precisan manexar enteiros cun tamaño maior. Nese caso é preciso recorrer a outro tipo de representacións, e definir tamén novas operacións aritméticas para traballar con eles.

Un enteiro grande pode representarse usando un vector de díxitos en base B, usando unha estrutura coma a seguinte:

```
typedef struct {
    int sign;
    int tam;
    int *val;
} bignum;
```

onde **signo** é o signo do inteiro, e **tam** é o tamanho do vector dinámico **val**.

Pídese implementar as seguintes funcións para o caso de $B=10$:

```
bignum str2bignum(char *str) //converte unha cadea a bignum.
```

Hai que decidir se o primeiro elemento do vector se corresponde coa cifra máis significativa ou coa menos significativa, e telo en conta no resto das operacións

```
bignum add(bignum a, bignum b) //devolve a soma de dous enteros grandes.
```

A ideia é percorrer os dois vectores sumando elemento a elemento tendo conta do acarreo:

- 1) Comezar pola posición da cifra menos significativa.
- 2) Ir sumando o acarreo previo e as cifras elemento a elemento ata o outro extremo dos vectores, gardando na mesma posición do vector resultante o valor suma % 10 e tomando como acarreo o valor suma/10.

```
bignum sub(bignum a, bignum b) //resta dous enteiros grandes.
```

- 1) Comezar pola posición da cifra menos significativa.
- 2) Súmaselle ao subtraendo o acarreo previo, despois faise a resta dos elementos correspondentes ata o outro extremo dos vectores, gardando na mesma posición do vector resultante. Se a diferenza é negativa súmaselle 10 e ponse o acarreo a 1, senón o acarreo é cero.

```
bignum mult(bignum a, bignum b) //devolve o produto de dos enteros grandes
```

23958233

✕

5830

00000000 (= 23958233 ✕ 0)
71874699 (= 23958233 ✕ 30)
191665864 (= 23958233 ✕ 800)
+ 119791165 (= 23958233 ✕ 5000)

139676498390 (= 139676498390)

[illegible]

```
bignum fact(bignum n) //devolve n!.
```

```
bignum div(bignum a,bignum b,bignum n)  //devolve a * b módulo n.
```

En todas as operacións hai que ter en conta o signo dos enteiros, e o efecto que teñen no resultado. No traballo con números grandes pódense empregar os algoritmos básicos aprendidos na escola, aínda que existen outros máis eficientes. Para o caso da división o algoritmo descríbese aquí:

[1] https://es.wikipedia.org/wiki/Divisi%C3%B3n_larga

INCLUIR AS SEGUINTE PARTES NA ENTREGA

1. Tedes que usar unha estrutura para representar un enteiro grande, e memoria dinámica para crear cada vector segundo o tamaño que sexa necesario.
2. Hai que crear un arquivo .h coas definición e un arquivo .c coa implementación da biblioteca.
3. Hai que escribir un programa para probar os algoritmo da biblioteca. Podedes comprobar o seu funcionamento con números pequenos.
4. Cada función debe especificarse debidamente coa precondition e a poscondición.
5. Noutro arquivo hai que indicar os diferentes casos de proba e as saídas que vai ofrecendo o programa.
6. O programa debe compilar nunha máquina Linux usando o compilador gcc.