

BÁO CÁO ĐỒ ÁN

MÔN HỌC: KIẾN TRÚC MÁY TÍNH VÀ HỢP NGỮ

ĐỒ ÁN 1: BIỂU DIỄN VÀ TÍNH TOÁN SỐ NGUYÊN LỚN

Thành viên

- ❖ Nguyễn Trần Trung – 18120625
- ❖ Võ Trọng Gia Vinh – 18120652
- ❖ Lê Thành Việt – 18120648

Công việc của mỗi thành viên

- ❖ Nguyễn Trần Trung
 - Xử lý chương trình chính
 - Tester
 - Fix bug
 - Viết báo cáo
- ❖ Võ Trọng Gia Vinh
 - Xây dựng class QInt
 - Xử lý chuyển cơ số
 - Phép cộng, trừ
 - Phép AND, OR, XOR, NOT
 - Dịch bit
- ❖ Lê Thành Việt
 - Phép nhân, chia
 - Xoay bit

Môi trường lập trình

MS Visual Studio – C++

Ý tưởng

Biểu diễn QInt bằng mảng 4 integer (4 x 32 bits): arrBit[4]

arrBit[0] arrBit[1] arrBit[2] arrBit[3]
[127-----96] [95-----64] [63-----32] [31-----0]

Phạm vi biểu diễn của QInt: -2^{127} đến $2^{127} - 1$

Chuyển đổi cơ số: chuyển tất cả về hệ nhị phân, rồi từ nhị phân chuyển sang hệ khác

Hexa sang binary: chuyển từng ký tự trong chuỗi hexa sang chuỗi 4 bit

Decimal sang binary: chia 2 lấy dư

Binary sang Hexa: gom nhóm 4 bit mã hóa ngược lại ký tự trong Hexa

Binary sang Decimal:

1. Nếu là số âm thì chuyển về giá trị tuyệt đối của binary.
2. $D = 0$;
3. Đọc dãy bit từ MSB đến LSB: $D = 2D$ nếu bit = 0, $D = 2D + 1$ nếu bit = 1
4. Trả về D là biểu diễn nhị phân (nếu là số âm thì thêm dấu -)

Phép cộng:

Lần lượt cộng từng phần tử của dãy số nguyên với nhau, kiểm tra nếu có bit nhớ phát sinh thì cộng tiếp bit nhớ vào phần tử sau đó.

Các trường hợp gây tràn số: $A + B = C$

1. A, B không âm; C âm
2. A, B âm; C không âm

Phép trừ:

Lần lượt trừ từng phần tử của dãy số nguyên với nhau, kiểm tra nếu có bit nhớ phát sinh thì trừ tiếp bit nhớ vào phần tử sau đó.

Các trường hợp gây tràn số: $A + B = C$

1. A không âm; B, C âm
2. A âm; B, C không âm

Phép nhân:

Thuật toán Booth's Multiplication: $M \times Q$

Khởi tạo: $A = 0$; $k = n$; $Q_1 = 0$ (thêm 1 bit = 0 vào cuối Q)

Lặp khi $k > 0$

{

 Nếu 2 bit cuối của Q_0Q_1

 {

 = 10 thì $A - M \rightarrow A$

 = 01 thì $A + M \rightarrow A$

 = 00, 11 thì A không thay đổi

 }

Shift arithmetic right [A, Q, Q_1]

$k = k - 1$

 }

Kết quả: [A, Q]

Các trường hợp kết quả không bị tràn:

1. $A = 0$, Q không âm
2. $A = -1$, Q âm

Phép chia:

Thuật toán Restoring Division Algorithm với 2 số dương: Q / M ($Q > M$)

Khởi tạo: A = n bit 0 nếu $Q > 0$; A = n bit 1 nếu $Q < 0$; $k = n$

Lặp khi $k > 0$

{

 Shift left (SHL) [A, Q]

$A - M \rightarrow A$

 # Nếu $A < 0$: $Q_0 = 0$ và $A + M \rightarrow A$

 # Ngược lại: $Q_0 = 1$

$k = k - 1$

 }

Kết quả: Q là thương, A là số dư

1. Chuyển 2 số về không âm, lưu dấu của kết quả
2. Thực hiện chia theo Restoring Division Algorithm
3. Đổi dấu kết quả nếu cần

Trường hợp gây tràn số: $\text{MINQINT} / -1 = -\text{MINQINT} > \text{MAXQINT}$

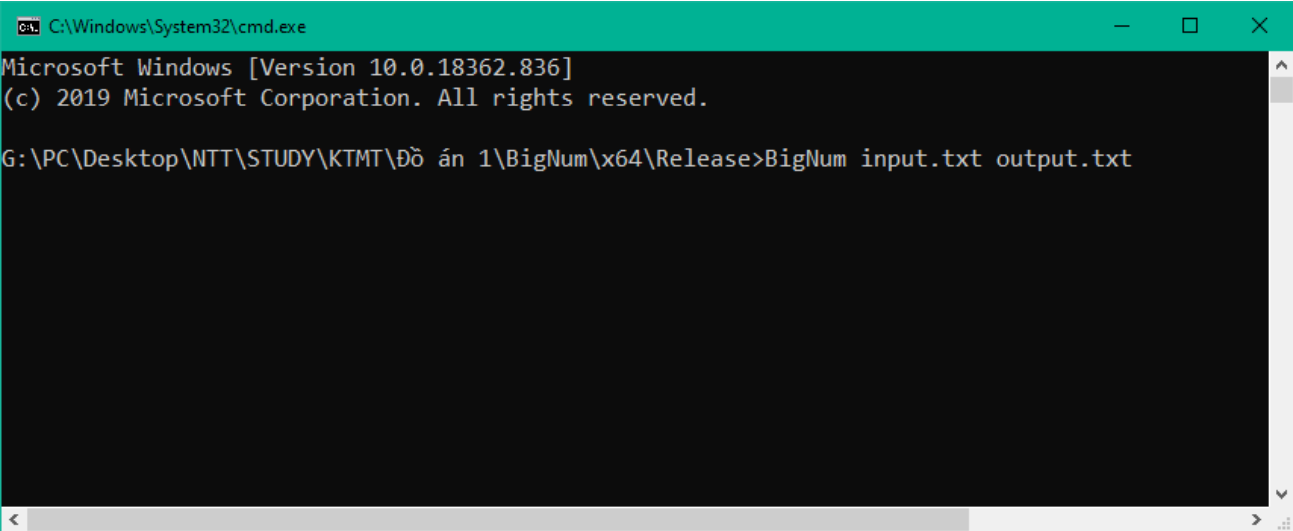
Toán tử AND, OR, XOR, NOT, dịch, xoay: thực hiện trên từng phần tử của mảng arrBit

Chức năng

Thực hiện tính toán các dòng trong file input, trả về kết quả trong file output.

Chạy chương trình bằng cửa sổ dòng lệnh:

BigNum <input file> <output file>



```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.18362.836]
(c) 2019 Microsoft Corporation. All rights reserved.

G:\PC\Desktop\NTT\STUDY\KMTT\Đồ án 1\BigNum\x64\Release>BigNum input.txt output.txt
```

Cú pháp các phép tính:

Chuyển cơ số:

<source base> <destination base> <number>

Cộng, trừ, nhân, chia, bitwise (2 ngôi), bit shift:

<base> <operand 1> <operator> <operand 2>

Rol, ror, not:

<base> <operator> <number>

Input example:

```
input.txt - Notepad
File Edit Format View Help
10 2 2960098589628
10 -2608479417 + -22338170235325255170379
16 94D30AD28A5DB57123A7571546F - 5A
10 -4647 * 4302371896916613314517792614108589
10 36446868780624476630340083329890 / 3191122320557494
16 539 & 83C0B8536
16 539 | 83C0B8536
16 539 ^ 83C0B8536
10 ~ 488108642434
10 rol 1898003864
10 ror -57
16 0127F4D6EC0FFA4601 << 41
16 0127F4D6EC0FFA4601 >> 41
|
```

Output example:

```
output.txt - Notepad
File Edit Format View Help
101011000100110011100111111111011101111100
-22338170235327863649796
94D30AD28A5DB57123A75715415
-19993122204971502072564182277762613083
11421332408924128
530
83C0B853F
83C0B800F
-488108642435
3796007728
-29
24FE9ADD81FF48C020000000000
93FA6B
```

Mức độ hoàn thành

100%

Tài liệu tham khảo

1. Slide KTMT&HN chương 2
2. <https://www.geeksforgeeks.org/overflow-in-arithmetic-addition-in-binary-number-system/>