

**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH**  
**TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN**  
**KHOA CÔNG NGHỆ THÔNG TIN**



**ĐỒ ÁN 1**

**KIẾN TRÚC MÁY TÍNH VÀ HỢP NGỮ - 18\_4**

**ĐỀ TÀI**

**BIỂU DIỄN VÀ TÍNH TOÁN SỐ NGUYÊN LỚN**

**Học kỳ 2 / 2019-2020**

Thành Phố Hồ Chí Minh – Năm 2020

## GIỚI THIỆU NHÓM

Nhóm thành lập ra phục vụ cho việc làm đồ án/bài tập môn **Kiến trúc máy tính và hợp ngữ** cũng như cập nhật thông tin làm việc và bài làm của nhóm.

- ✚ Mã lớp: **18\_4**
- ✚ Số thành viên: **2**
- ✚ Giảng viên hướng dẫn: Thầy **Lê Quốc Hòa**

Danh sách thành viên nhóm:

STT	MSSV	Họ và tên	Email	SĐT	Vai trò
1	18120397	Nguyễn Đặng Hồng Huy	honghuyqtak2018@gmail.com	0905263087	Trưởng nhóm
2	18120408	Trần Ngọc Lan Khanh	lankhanh1482@gmail.com	0949813287	

## GIỚI THIỆU ĐỒ ÁN

Kiểu dữ liệu số nguyên lớn có dấu gọi là QInt có độ lớn 16 byte gồm một số chức năng:

- ✚ Chuyển đổi số QInt từ hệ thập phân sang hệ nhị phân (dạng bù 2) và ngược lại
- ✚ Chuyển đổi số QInt từ hệ nhị phân (dạng bù 2) sang hệ thập lục phân và ngược lại
- ✚ Chuyển đổi số QInt từ hệ thập lục phân sang thập phân và ngược lại
- ✚ Các operator =, operator +, operator -, operator \*, operator / trên các hệ cơ số
- ✚ Các toán tử AND "&", OR "|", XOR "^", NOT "~"
- ✚ Các toán tử: dịch trái "<<", dịch phải ">>" số học
- ✚ Các phép xoay trái "rol", xoay phải "ror" mỗi lần xoay chỉ xử lý cho đúng 1 bit, không cần xử lý cho trường hợp tổng quát xoay k bit

## MỤC LỤC

<b>A. BÁO CÁO NHÓM .....</b>	<b>4</b>
1. Nguyên tắc hoạt động nhóm.....	4
2. Kế hoạch làm việc .....	4
3. Phân công công việc .....	4
<b>B. BÁO CÁO ĐỒ ÁN.....</b>	<b>6</b>
1. Môi trường lập trình .....	6
2. Ý tưởng thiết kế và thực hiện .....	6
3. Phạm vi biểu diễn .....	8
4. Mức độ hoàn thành .....	9
5. Kiểm tra .....	9
6. Tài liệu tham khảo .....	11

## A. BÁO CÁO NHÓM

### 1. Nguyên tắc hoạt động nhóm

- Tất cả vì mục tiêu chung của nhóm;
- Cố gắng lắng nghe các thành viên nhiều hơn;
- Không ngại bộc lộ quan điểm cá nhân;
- Trưởng nhóm phải là người có khả năng lãnh đạo;
- Tất cả các thành viên đều phải có tinh thần trách nhiệm;
- Luôn làm việc trên tinh thần gắn kết;
- Tôn trọng và hỗ trợ lẫn nhau.

### 2. Kế hoạch làm việc

STT	Thời gian	Nội dung	Ghi chú
1	24/5/2020	<ul style="list-style-type: none"> <li>• Lập nhóm</li> <li>• Họp lần 1: Thống nhất cách làm việc Chia đều các mục tìm hiểu, tìm thuật toán phù hợp</li> <li>• Deadline công việc: 23h 28/5/2020</li> </ul>	
2	29/5/2020	<ul style="list-style-type: none"> <li>• Họp nhóm lần 2: Trao đổi tiến độ làm việc, tổng hợp code Thống nhất cách đặt tên hàm Mỗi cá nhân tiếp tục hoàn thành phần việc của bản thân</li> <li>• Deadline công việc: 23h 4/6/2020</li> </ul>	
3	5/6/2020	<ul style="list-style-type: none"> <li>• Họp nhóm lần 3: Báo cáo tiến độ công việc, tổng hợp code Đọc vào các test case, tìm bug, thêm comment Bắt đầu viết báo cáo</li> <li>• Deadline công việc: 23h 11/6/2020</li> </ul>	
4	9/6/2020	<ul style="list-style-type: none"> <li>• Họp nhóm lần 4: Thông qua ý kiến cả nhóm về Đồ án Đánh giá các thành viên</li> <li>• Đánh giá tổng quát Đồ án</li> </ul>	
5	10/6/2020	<ul style="list-style-type: none"> <li>• Nộp bài lên Moodle</li> </ul>	

### 3. Phân công công việc

STT	MSSV	Họ và tên	Nhiệm vụ	Ghi chú
1	18120397	Nguyễn Đăng Hồng Huy	<ul style="list-style-type: none"> <li>• Tìm hiểu số nguyên 16 bytes, tổng hợp vào file pdf (phục vụ cho viết báo cáo).</li> <li>• Chuyển đổi số nguyên sang dạng nhị phân và ghi xuống bộ nhớ (<b>QInt</b>). (code)</li> <li>• Tổng hợp code sau khi hoàn thành.</li> </ul>	Công việc 1
			<ul style="list-style-type: none"> <li>• Toán tử so sánh trên bit (code)</li> <li>• Chuyển đổi Qint qua lại các hệ 2, 10, 16</li> <li>• Xử lý operator</li> <li>• Tham số dòng lệnh, đối số dòng lệnh</li> </ul>	Công việc 2
			<ul style="list-style-type: none"> <li>• Overload toán tử trên bit cho <b>QInt</b>. (code)</li> <li>• Viết báo cáo đồ án</li> </ul>	Công việc 3
2	18120408	Trần Ngọc Lan Khanh	<ul style="list-style-type: none"> <li>• Tìm hiểu số nguyên 16 bytes, tổng hợp vào file pdf (phục vụ cho viết báo cáo).</li> <li>• Chuyển đổi số nguyên sang dạng nhị phân và ghi xuống bộ nhớ (<b>QInt</b>). (code)</li> </ul>	Công việc 1
			<ul style="list-style-type: none"> <li>• Xử lý file</li> <li>• Overload các toán tử của <b>QInt</b>. (code)</li> <li>• Xử lý operator trên các hệ cơ số</li> </ul>	Công việc 2
			<ul style="list-style-type: none"> <li>• Overload các toán tử trên bit của <b>QInt</b>. (code)</li> <li>• Viết báo cáo đồ án</li> </ul>	Công việc 3

## B. BÁO CÁO ĐỒ ÁN

### 1. Môi trường lập trình

- Ngôn ngữ lập trình: C++
- Môi trường lập trình: Visual Studio 2019

### 2. Ý tưởng thiết kế và thực hiện

Lưu QInt bằng unsigned char 16 phần tử (16 bytes = 128 bits)

_arrBits[16]	[0]	[1]	...	[14]	[15]
128 bits	XXXX XXXX	XXXX XXXX	...	XXXX XXXX	XXXX XXXX
	↓ Bit trái nhất (bit dấu)				
X = {0,1}					

- Chuyển đổi số QInt từ **hệ thập phân** sang **hệ nhị phân**:

Kiểm tra phần tử đầu tiên trong chuỗi Qint có phải là dấu trừ không? Nếu phải, xóa dấu trừ đi. Sau đó lần lượt lấy phần tử ở cuối chuỗi chuyển từ kiểu char sang kiểu số nguyên, sau đó mod cho 2, được kết quả '0' hoặc '1' và cộng vào chuỗi kết quả. Bước tiếp theo, lấy Qint chia 2, nếu Qint != 0, ta tiếp tục vòng lặp. Kết thúc vòng lặp, đảo chuỗi kết quả để được kết quả cuối cùng. Nếu kết quả âm, ta cần thực hiện thêm bước lấy số bù 2 để ra kết quả cuối cùng.

- Chuyển đổi số QInt từ **hệ thập lục phân** sang **hệ nhị phân**:

Kiểm tra từng phần tử của chuỗi Hexa là kí tự gì trong chuỗi HEXA\_CHAR "0123456789ABCDEF". Sau đó chuyển phần tử đó thành chuỗi Bin (8 bits), cộng dồn vào resBin.

- Chuyển đổi số QInt từ **hệ nhị phân** sang **hệ thập phân**:

Khởi tạo res = "1", bắt đầu từ vị trí bit đầu tiên sau bit 1 từ trái sang, lần lượt tính  $res = res * 2 + carry$  (carry = 0 nếu bit đọc được là 0 và carry = 1 nếu bit đọc được là 1) cho đến cuối chuỗi Bin.

- Phép cộng:

Khởi tạo một biến nhớ bằng 0, bắt đầu từ vị trí bit thứ 127, lấy bit tại vị trí thứ i của số QInt A cộng với bit tại vị trí thứ i của số QInt B và cộng với biến nhớ.

- Nếu kết quả cộng là 0: khả năng A = 0, B = 0, mind = 0 thì set bit thứ i trong chuỗi kết quả là 0.

- Nếu kết quả cộng là **1**: khả năng  $A + B = 1$ ,  $mind = 0$  hoặc là  $A = 0, B = 0, mind = 1$  thì set bit thứ  $i$  trong chuỗi kết quả là 1.
- Nếu kết quả cộng là **2**: khả năng  $A + B = 0$ ,  $mind = 1$  hoặc là  $A = 1, B = 1, mind = 0$  thì set bit thứ  $I$  trong chuỗi kết quả là 0.
- Nếu kết quả cộng là **3**: khả năng  $A = 1, B = 1, mind = 1$  thì set bit thứ  $I$  trong chuỗi kết quả là 1.

- Phép nhân: sử dụng thuật toán Booth

Khởi tạo các số Qint  $A = 0, Q = B, Q\_0 = 0$

Lặp khi số bit nhỏ hơn 128, mỗi lần lặp lấy LSB của  $Q$

$LSB\_Q = 0$  và  $Q\_0 = 1, A = A + *this$  (M trong thuật toán Booth)

$LSB\_Q = 1$  và  $Q\_0 = 0, A = A - *this$

Dịch phải  $Q\_0, Q, A$ .

Kết thúc vòng lặp, trả kết quả là  $Q$ .

- Phép chia:

Khởi tạo biến số dư mod kiểu Qint bằng 0, div là số bị chia,  $M = \text{divisor}$  là số chia

Nếu div âm, lấy dạng bù 2 để chuyển thành dương, tương tự với divisor.

Duyệt qua tất cả các bit

- Dịch trái 1bit cho mod, lấy bit trái nhất của div gán cho bit phải nhất của mod, sau đó dịch trái 1bit cho div
- Kiểm tra  $mod - M$ , nếu dương gán  $mod = mod - M$ , đồng thời set bit phải nhất của div là 1
- Nếu âm, tiếp tục vòng lặp

Nếu số bị chia và số chia trái dấu, đổi dấu thương.

- Các toán tử AND “&”, OR “|”, XOR “^”, NOT “~”

Sử dụng vòng lặp, dùng toán tử cho từng cặp phần tử tương ứng của hai số Qint A và Qint B.

- Operator<<

Bắt đầu từ phần tử đầu tiên, lần lượt duyệt qua các phần tử của chuỗi `_arrBit[16]` và tiến hành dịch trái từng phần tử, mặc định sẽ set bit 0 vào vị trí bit LSB của phần tử. Tuy nhiên, nếu MSB của phần tử kế tiếp là 1, ta set bit 1 thay vì bit 0 như trường hợp mặc định.

- Operator >>

Bắt đầu từ phần tử đầu tiên, lần lượt duyệt qua các phần tử của chuỗi `_arrBit[16]` và tiến hành dịch phải từng phần tử, mặc định sẽ set bit 0 vào vị trí bit MSB của phần tử. Tuy nhiên, nếu LSB của phần tử trước đó là 1, ta set bit 1 thay vì bit 0 như trường hợp mặc định.

- Operator ==

So sánh từng byte của hai số Qint A và Qint B, trả về false nếu có cặp byte khác giá trị.

- Phép xoay trái “rol”

Đầu tiên, lưu bit trái nhất, sau đó bắt đầu từ byte đầu tiên, lần lượt duyệt qua các phần tử của chuỗi `_arrBit[16]` và tiến hành dịch tất cả các bit sang trái. Nếu bit trái nhất của byte thứ  $i + 1$  là 1, sau khi xoay trái, bit phải nhất của byte thứ  $i$  được set là 1. Đối với byte cuối cùng, set bit phải nhất bằng bit được lưu ban đầu.

- Phép xoay phải “ror”

Đầu tiên, lưu bit phải nhất, sau đó bắt đầu từ byte cuối cùng, lần lượt duyệt ngược qua các phần tử của chuỗi `_arrBit[16]` và tiến hành dịch tất cả các bit sang phải. Nếu bit phải nhất của byte thứ  $i - 1$  là 1, sau khi xoay phải, bit trái nhất của byte thứ  $i$  được set là 1. Khi duyệt đến byte đầu tiên, set bit trái nhất bằng bit được lưu ban đầu.

### 3. Phạm vi biểu diễn

Hệ	Giá trị	Biểu diễn
Nhị phân	max	0111 1111 1111 1111 1111 1111 1111 1111 1111 1111 1111 1111 1111 1111 1111 1111 1111 1111 1111 1111
	min	1000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
Thập phân	max	170141183460469231731687303715884105727
	min	-170141183460469231731687303715884105728
Thập lục phân	max	7FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
	min	80000000000000000000000000000000




#### 4. Mức độ hoàn thành


Yêu cầu		Mức độ hoàn thành (%)	Ghi chú
QInt	Chuyển đổi số QInt từ hệ thập phân sang hệ nhị phân (dạng bù 2) và ngược lại	100	
	Chuyển đổi số QInt từ hệ nhị phân (dạng bù 2) sang hệ thập lục phân và ngược lại	100	
	Chuyển đổi số QInt từ hệ thập lục phân sang thập phân và ngược lại	100	
	Các operator =, operator +, operator -, operator *, operator / trên các hệ cơ số	100	
	Các toán tử AND “&”, OR “ ”, XOR “^”, NOT “~”	100	
	Các toán tử: dịch trái “<<”, dịch phải “>>” số học	100	
	Các phép xoay trái “rol”, xoay phải “ror” mỗi lần xoay chỉ xử lý cho đúng 1 bit	100	
Tổng thể đồ án		100	

#### 5. Kiểm tra


Scrip\_test\_case\_KTMT\_HN\_QInt\_Window\exe\_sv

Name	Date modified	Type	Size
 18120397_18120408	04/06/2020 10:42 PM	Application	50 KB

## Scrip\_test\_case\_KTMT\_HN\_QInt\_Window\output\_sv

Name	Date modified	Type	Size
 18120397_18120408_output	04/06/2020 10:53 PM	Text Document	5 KB

## Scrip\_test\_case\_KTMT\_HN\_QInt\_Window\grade\_sv

Name	Date modified	Type	Size
 18120397_18120408_grade	04/06/2020 10:53 PM	Text Document	1 KB

## 18120397\_18120408\_output.txt

18120397\_18120408\_output - Notepad

File Edit Format View Help

```

-19993122204971502072564182277762613083
FC5380D4780BBFBAE920DE3900
-94993321373287261627843808865078
100000000101111110110010001100110001101111011010001000111101100
1882290153772433758370147818410
11111111111110110100001000001001010011110000100111101100100100011101101001100100
-114812731097961891092328483067729915140
-438609857555182398685
134821250756587495699407289043889
11111111111011100011000010010111100000110101111110010110110100101001000100001101000000
3796007728
-29
80000000000000000000000000000000E3E
1110010100000011010001110100001011110100000000111110011100
4D87BB0
80000000000000000000617B9592B8A37B6
10111100000101110000011100010011
-102041623119479023047863498877
-488108642435

```

Ln 91, Col 17    100%    Windows (CRLF)    UTF-8

18120397\_18120404\_grade.txt



```
1
1
1
1
1
1
1
1
1
1
1
1
1
1
1
Sum: 100
```

## 6. Tài liệu tham khảo

- Slide bài giảng *Kiến trúc máy tính và hợp ngữ*, Bộ môn MMT-VT, Khoa CNTT, ĐH Khoa học Tự nhiên, ĐHQG HCM