



---

# KIẾN TRÚC MÁY TÍNH VÀ HỢP NGỮ

---

Báo cáo đồ án 1

## BIỂU DIỄN VÀ TÍNH TOÁN SỐ NGUYÊN LỚN

06 THÁNG TƯ 2019

# BÁO CÁO ĐỒ ÁN 1

## BIỂU DIỄN VÀ TÍNH TOÁN SỐ NGUYÊN LỚN

### 1. Thành Viên Nhóm:

STT	Họ Tên	MSSV	Nhiệm Vụ
1	Trần Vũ Công	1712306	Chuyển đổi số QInt từ hệ thập phân sang hệ nhị phân (dạng bù 2), thập lục phân và ngược lại. Các toán tử: dịch trái "<", dịch phải ">" số học. Các phép xoay trái "rol", xoay phải "ror".
2	Nguyễn Văn Diện	1712342	Chuyển đổi số QInt từ hệ nhị phân (dạng bù 2) sang hệ thập lục phân, thập phân và ngược lại. Các operator=, operator+, operator-, operator*, operator/ trên các hệ cơ số
3	Tạ Tiến Thành Đạt	1712333	Chuyển đổi số QInt từ hệ thập lục phân sang hệ nhị phân (dạng bù 2), thập phân và ngược lại. Các toán tử AND "&", OR " ", XOR "^", NOT "~"

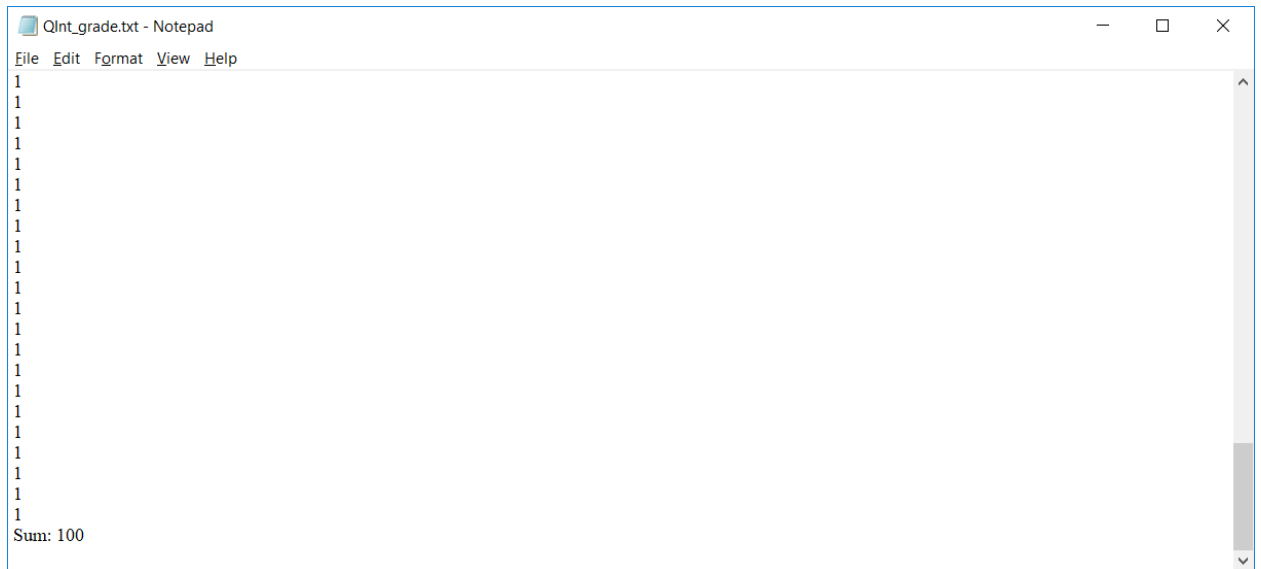
2. Môi trường lập trình: Microsoft Visual Studio 2013 bằng C ++ .

3. Ý tưởng thiết kế:

- Sử dụng 2 biến kiểu long long left và right để biểu diễn số Qint, trong đó biến left chứa 64 bit đầu tiên (từ 0 -> 63), biến right chứa 64 bit còn lại (từ 64 -> 127).
- Xây dựng những hàm tính toán của Qint: chuyển đổi qua lại giữa các dạng cơ số, các toán tử + - \* / =, các phép xử lí bit AND OR XOR NOT << >> ROR ROL.

- + **Chuyển đổi từ 10 -> 2(dạng bù 2):** Chuyển từ chuỗi hệ cơ số 10 sang kiểu QInt bằng cách chia dần số thập phân cho 2 cho đến khi kết quả bằng 0, mỗi lần chia lấy số dư phép chia gán cho bit theo thứ tự. Nếu là số âm thì thực bỏ dấu âm rồi chia như bình thường, sau đó chuyển sang dạng bù 2. Rồi từ QInt chuyển sang dạng bù 2 bằng hàm `Com_2()`, sau đó chuyển sang chuỗi nhị phân.
- + **Chuyển đổi từ 2(dạng bù 2) -> 10:** Từ chuỗi nhị phân chuyển sang kiểu QInt rồi từ QInt chuyển sang chuỗi nhị phân lại rồi từ chuỗi nhị phân chuyển sang hệ 10. Vì tất cả các tham số đầu vào đều chuyển sang kiểu QInt nên có sự lặp lại ở chỗ này.
- + **Chuyển đổi từ 16 -> 2(dạng bù 2):** Với mỗi kí tự trong số hệ 16, chuyển thành 4 bit tương ứng trong hệ nhị phân rồi gán vào giá trị của số. Lặp lại cho đến hết. Sau đó đảo tất cả các bit và + 1 (cũng chuyển về kiểu QInt).
- + **Chuyển đổi từ 2 -> 16:** xét mỗi 4 bit trong số hệ 2 rồi chuyển thành mã hexa tương ứng. Lặp lại cho đến hết (cũng chuyển về kiểu QInt).
- + **Phép cộng:** Chuyển hết các tham số đầu vào về kiểu dữ liệu QInt rồi dùng bất, tất bit trên kiểu dữ liệu QInt để tính như bình thường.
- + **Phép trừ:** Tương tự như phép cộng, coi  $a - b$  như  $a + (-b)$ . Đối với  $(-b)$  dùng hàm `Com_2()` (chuyển về dạng bù 2) để thực hiện.
- + **Phép nhân:** Sử dụng thuật toán nhân cải tiến Booth's Multiplication Algorithm.
- + **Phép chia:** Sử dụng thuật toán Restoring Division Algorithm để thực hiện phép chia. Ở đây chỉ thực hiện trên hai số không âm rồi dùng biến đếm để đếm số đầu đầu vào rồi từ đó gán cho kết quả.
- + **Toán tử AND &:** thực hiện trên nhị phân, AND các bit của 2 số với nhau.
- + **Toán tử OR |:** thực hiện trên nhị phân, OR các bit của 2 số với nhau.
- + **Toán tử XOR ^:** thực hiện trên nhị phân, XOR các bit của 2 số với nhau.
- + **Toán tử NOT ~:** thực hiện trên nhị phân, NOT các bit của số đó gán vào kết quả.
- + **Phép toán dịch trái <, dịch phải >:** dịch lần lượt từng bit sang bên phải 1 bit, lặp lại n lần (với n là yêu cầu của đề bài).
- + **Phép toán ROL:** lưu lại giá trị của bit thứ 127, thực hiện tương tự phép dịch trái 1 bit sau đó gán giá trị của bit thứ 127 đã được lưu cho bit thứ 0.
- + **Phép toán ROR:** lưu lại giá trị của bit thứ 0, thực hiện tương tự phép dịch phải 1 bit sau đó gán giá trị của bit thứ 0 đã được lưu cho bit thứ 127. - Phạm vi biểu diễn:  $-2^{127} \rightarrow 2^{127} - 1$ .

#### 4. Chạy kiểm thử:



```
QInt_grade.txt - Notepad
File Edit Format View Help
1
1
1
1
1
1
1
1
1
1
1
1
1
1
1
1
1
1
1
1
1
Sum: 100
```

Đạt 100/100

5. Đánh giá:

Tốt

6. Các nguồn tài liệu tham khảo:

<https://github.com/hoangvietnguyen99/KTMT-HN-Project01/tree/master/Project01>

[https://github.com/beohoang98/DoAn\\_BigInt?fbclid=IwAR29TfHq396TwNVUMQNQjo054E6t3Pi\\_htE\\_y4m9\\_nNAFqc4YoQHRu4QjZs](https://github.com/beohoang98/DoAn_BigInt?fbclid=IwAR29TfHq396TwNVUMQNQjo054E6t3Pi_htE_y4m9_nNAFqc4YoQHRu4QjZs)