ĐẠI HỌC KHOA HỌC TỰ NHIÊN
ĐHQG TP. HCM

KHOA CÔNG NGHỆ THÔNG TIN

LỚP 14CTT2.1

BÁO CÁO ĐỒ ÁN 1

MÔN KIẾN TRÚC MÁY TÍNH VÀ HỢP NGỮ

BIỂU DIỄN VÀ TÍNH TOÁN SỐ NGUYÊN LỚN

GIẢNG VIÊN GIẢNG DẠY
Ths. Vũ Minh Trí
Ths. Trương Phước Hưng

CÁC THÀNH VIÊN

Nguyễn Thị Thu Hà	**1412140**
Võ Phương Hòa	**1412192**
Đoàn Thị Phương Huyền	**1412197**

Tp. HCM, tháng 10/2015

MỤC LỤC

1. Thông tin các thành viên trong nhóm

Stt	Họ và tên	MSSV	Email
1	Nguyễn Thị Thu Hà	1412140	thuhapa@gmail.com
2	Võ Phương Hòa	1412192	phuonghoa.nt000@gmail.com
3	Đoàn Thị Phương Huyền	1412197	dtphuyen2506@gmail.com

2. Môi trường lập trình

- Ngôn ngữ lập trình: C++.

- Môi trường lập trình: Microsoft Visual Studio 2010

3. Ý tưởng thiết kế và thực hiện đồ án

- Sử dụng mảng arrBits gồm 2 phần tử kiểu __int64 để biểu diễn số nguyên lớn 128 bit, phần tử thứ nhất biểu diễn từ bit 64 tới bit 127 và phần tử thứ 2 biểu diễn từ bit 0 tới bit 63.

- Xử lí và tính toán trên mảng arrBits gồm 2 phân tử __int64

__int64	**127**	**126**	**125**	**…**	**67**	**66**	**65**	**64**	**63**	**62**	**61**	**…**	**3**	**2**	**1**	**0**
mảng hai phần tử	**arrBits[0]**								**arrBits[1]**							
vị trí bit	**63**	**62**	**61**	**…**	**3**	**2**	**1**	**0**	**63**	**62**	**61**	**…**	**3**	**2**	**1**	**0**

- Phạm vi biểu diễn các kiểu dữ liệu đã thiết kế

Hệ	Giá trị nhỏ nhất	Giá trị lớn nhất
Thập phân	-2^{127}	$2^{127}-1$
Nhị phân	1000…..0000000000 (127 chữ số 0)	01111……111111 (127 chữ số 1)
Thập lục phân	0x80……..00 (15 chữ số 0)	0x7F……..FF (15 chữ F)

Chi tiết về quá trình thực hiện đồ án

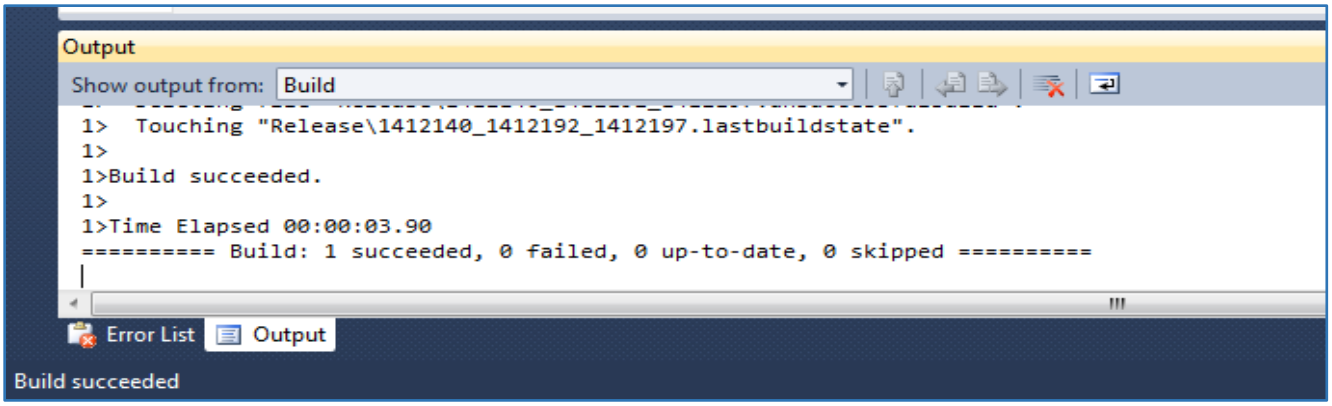
Hàm	Kiểu trả về	Chức năng	Ý tưởng thực hiện
findTwo_Comp() const	QInt	Tìm bù 2 của một số âm (mở rộng: tìm số đối của một số nguyên).	Tìm bit 1 đầu tiên tính từ phải sang, từ vị trí kế tiếp bit 1 vừa tìm được (chiều từ phải sang), đảo tất cả các bit.
setBit(int, int) const	QInt	Đặt bit cho số kiểu QInt.	Dùng biến mặt nạ (mask) (khởi gán bằng 1) để xác định vị trí đặt bit (bằng cách dịch trái mask n lần, n bằng vị trí đặt bit) rồi xor số kiểu QInt với mask, kết quả gán cho số kiểu QInt.
div2(string&)	void	Chia chuỗi số thập phân cho 2.	Tương tự như chia một số thập phân cho 2, từ trái sang phải, chia lần lượt từng ký tự số cho 2.

convertDecToQInt (string) const	QInt	Chuyển chuỗi số thập phân sang kiểu QInt.	Kiểm tra ký tự đầu chuỗi để xác định số âm hay dương, xóa ký tự đầu nếu chuỗi số âm. Vòng lặp: lấy ký tự số cuối chuỗi chia cho 2, bit là số dư, đặt bit vào số q (kiểu QInt), sau đó chia chuỗi số cho 2. Thực hiện vòng lặp cho đến khi chuỗi rỗng. Gọi hàm bù 2 nếu chuỗi số ban đầu âm.
convertBinToQInt (string) const	QInt	Chuyển chuỗi số nhị phân sang kiểu QInt.	Vòng lặp: từ cuối chuỗi về đầu chuỗi, lấy bit từ chuỗi, đặt bit vào số q (kiểu QInt).
convertHexToQInt (string) const	QInt	Chuyển chuỗi số thập lục phân sang kiểu QInt.	Vòng lặp: từ cuối chuỗi về đầu chuỗi, lấy kí tự số hoặc chữ từ chuỗi, chuyển về dạng số thập phân tương ứng, biểu diễn số đó bằng 4 bit. Từ phải sang trái, lấy từng bit của số đó, đặt bit vào q (kiểu QInt).
operator = (const QInt&)	QInt&	Định nghĩa toán tử gán "=".	Trong mảng arrBits (2 phần tử), gán từng phần tử này cho phần tử kia với chỉ số tương ứng.
operator + (const QInt&) const	const QInt	Định nghĩa toán tử cộng "+".	Cộng từng phần tử có chỉ số tương ứng. Kiểm tra các trường hợp tràn bit, nếu tràn bít thì cộng 1 vào phần tử thứ nhất của biến tổng.
operator - (const QInt&) const	const QInt	Định nghĩa toán tử trừ "-".	Tìm số đối của số hạng thứ 2. Cộng số hạng thứ nhất với số đối vừa tìm được. Sử dụng phép cộng và hàm tìm bù 2.
operator * (const QInt&) const	const QInt	Định nghĩa toán tử nhân "*".	Khởi tạo: temp = 0, check = 0, mul = thừa số 2. Vòng lặp (128 lần): Nếu 2 bit cuối của mul và check lần lượt là + 1 và 0: temp = temp - thừa số 1 + 0 và 1: temp = temp + thừa số 1 Dịch phải 1 bit [temp, mul, check] Kết quả: mul là tích cần tìm.
operator / (const QInt&) const	const QInt	Định nghĩa toán tử chia "/".	Khởi tạo: temp = 0, quo = số bị chia, m = số chia. Xét dấu của số bị chia và số chia để xét dấu của thương, Vòng lặp (128 lần): Dịch trái 1 bit [temp ,quo]. temp = temp – m. Nếu temp âm, gán bit đầu của quo bằng 0, temp = temp + m. Nếu temp không âm, gán bit đầu của quo bằng 1. Kết quả: quo là thương cần tìm.
operator & (const QInt&) const	const QInt	Định nghĩa toán tử và (and) "&".	Thực hiện phép và (&) từng phần tử của mảng biểu diễn QInt này với từng phần tử của mảng biểu diễn QInt kia với chỉ số tương ứng.

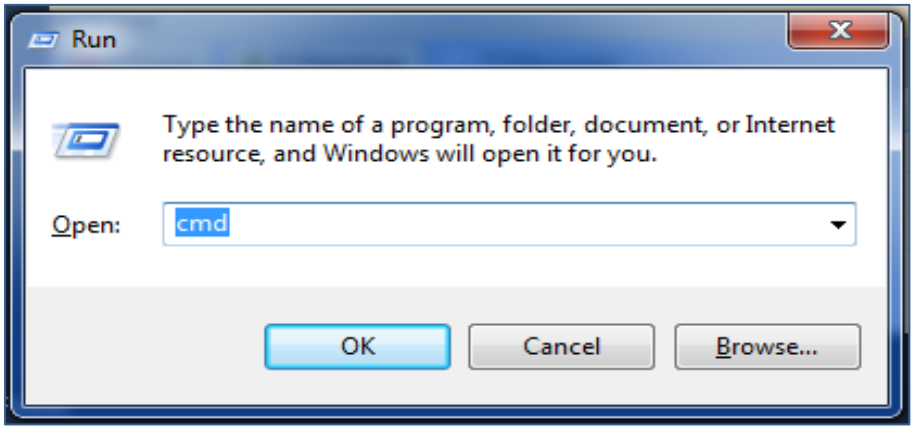
operator | (const QInt&) const	const QInt	Định nghĩa toán từ hoặc (or) "|".	Ý tưởng như phép và (&).
operator ^ (const QInt&) const	const QInt	Định nghĩa toán tử xor "^".	Ý tưởng như phép và (&).
operator ~ () const	const QInt	Định nghĩa toán tử not "~".	Ý tưởng như phép và (&).
operator << (int) const	const QInt	Định nghĩa toán tử dịch trái "<<".	Vòng lặp n lần (n là số lần dịch trái): Dịch trái phần tử thứ nhất 1 bit. Gán bit đầu của phần tử thứ nhất bằng bit dấu của phần tử thứ hai. Dịch trái phần tử thứ hai 1 bit.
operator >> (int) const	const QInt	Định nghĩa toán tử dịch phải ">>".	Vòng lặp n lần (n là số lần dịch phải): Dịch phải phần tử thứ hai 1 bit. Gán bit dấu của phần tử thứ hai bằng bit đầu của phần tử thứ nhất. Dịch phải phần tử thứ nhất 1 bit.
rotateLeft() const	QInt	Hàm xoay trái.	Lấy bit dấu của phần tử thứ nhất. Dịch trái phần tử thứ nhất 1 bit. Gán bit đầu của phần tử thứ nhất bằng bit dấu của phần tử thứ hai. Dịch trái phần tử thứ hai 1 bit. Gán bit đầu của phần tử thứ hai bằng bit dấu của phần tử thứ nhất (đã lấy ở trên).
rotateRight() const	QInt	Hàm xoay phải.	Lấy bit đầu của phần tử thứ hai. Dịch phải phần tử thứ hai 1 bit. Gán bit dấu của phần tử thứ hai bằng bit đầu của phần tử thứ nhất. Dịch phải phần tử thứ nhất 1 bit. Gán bit dấu của phần tử thứ nhất bằng bit đầu của phần tử thứ hai (đã lấy ở trên).
add(string, string)	string	Tìm tổng hai số nguyên dương (dạng chuỗi).	Khởi gán chuỗi kết quả bằng rỗng. Cân bằng độ dài 2 chuỗi. Vòng lặp: từ cuối chuỗi về đầu chuỗi, lấy tổng của biến nhớ với 2 kí tự số tại chỉ số i trong vòng lặp, chia 10, biến nhớ bằng thương, số dư chuyển sang ký tự để thêm vào đầu chuỗi kết quả.
sub(string, string)	string	Tìm hiệu hai số nguyên dương.	Ý tưởng gần như tính tổng 2 chuỗi.

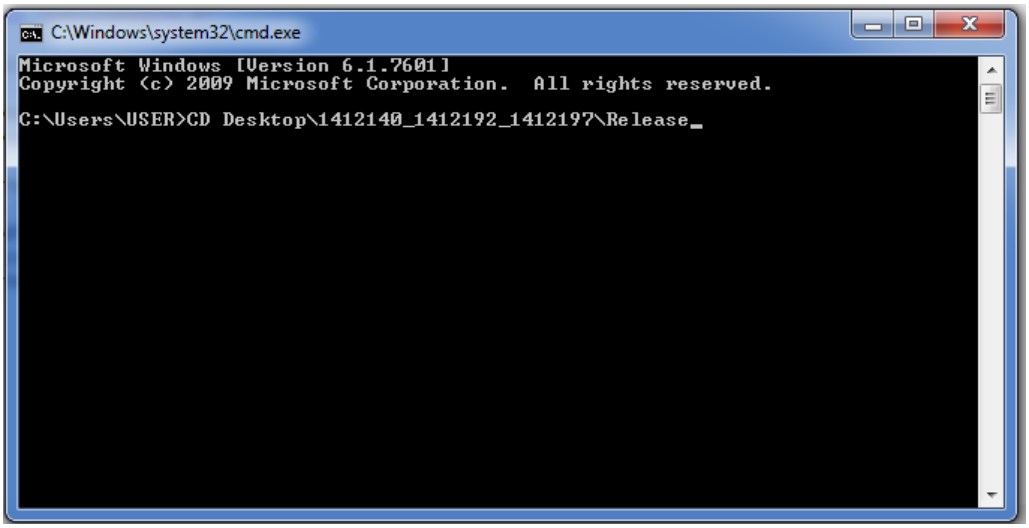
coutDec(const QInt&)	void	Xuất số kiểu QInt dưới dạng thập phân.	Khởi gán chuỗi kết quả bằng "0". Vòng lặp (127 lần), từ bit thứ 0 đến 126: Nếu bit là 1, tìm số thập phân tương ứng với bit 1tại vị trí đó, cộng số tìm được (dạng string) vào chuỗi kết quả. Nếu số kiểu QInt âm, gán chuỗi kết quả bằng hiệu của 2^127 (dạng string) với chuỗi kết quả, thêm dấu '-' vào đầu chuỗi kết quả.
coutBin(const QInt&)	void	Xuất số kiểu QInt dưới dạng nhị phân.	Tìm vị trí bit 1 đầu tiên tính từ trái sang, xuất tất cả các bit từ vị trí đó. Nếu số kiểu QInt bằng 0 thì xuất 0, thoát hàm.
coutHex(const QInt&)	void	Xuất số kiểu QInt dưới dạng thập lục phân.	Từ trái sang phải, xét từng bộ 4 bit liên tiếp, tìm vị trí bộ đầu tiên có biểu diễn thập phân khác 0. Từ vị trí tìm được, từ trái sang phải, xuất số hoặc ký tự tương ứng với từng bộ 4 bit nhị phân.
intput()	void	Đọc phép toán trên 1 dòng của file.	Đọc hệ số thứ nhất. Đọc chuỗi Nếu chuỗi bằng "~" hoặc "rol" hoặc "ror", gán toán tử bằng chuỗi, đọc toán hạng thứ nhất. Nếu chuỗi là toán hạng, đọc tiếp toán tử Nếu toán tử là chuỗi số, gán toán hạng thứ nhất bằng toán tử, gán toán tử bằng rỗng, chuyển chuỗi về số nguyên, gán cho hệ số thứ hai. Nếu toán tử bằng "<<" hoặc ">>", đọc tiếp hệ số thứ hai. Trường hợp còn lại, đọc tiếp toán hạng thứ hai.
readFile(vector <CalculateQInt> &)	void	Đọc file.	Khai báo chuỗi ký tự kiểm tra (char). Đọc file và lưu vào mảng phép toán chừng nào kết quả con trỏ trả về khác NULL khi đọc bằng chuỗi kiểm tra.
writeFile(vector <CalculateQInt>)	void	Ghi kết quả vào file.	Vòng lặp n lần (n = số phép toán): ghi kết quả phép toán vào file dựa vào hệ số thứ nhất (hoặc thứ hai nếu phép toán là chuyển đổi cơ số).
calculate(vector <CalculateQInt>)	void	Tính toán phép toán.	Dựa vào ký tự cuối của chuỗi toán tử ('+', '-', '&', '>', 'l', 'r', …) gọi hàm tính toán tương ứng. Nếu chuỗi rỗng, kết quả bằng toán hạng thứ nhất.

4. Chạy kiểm tra

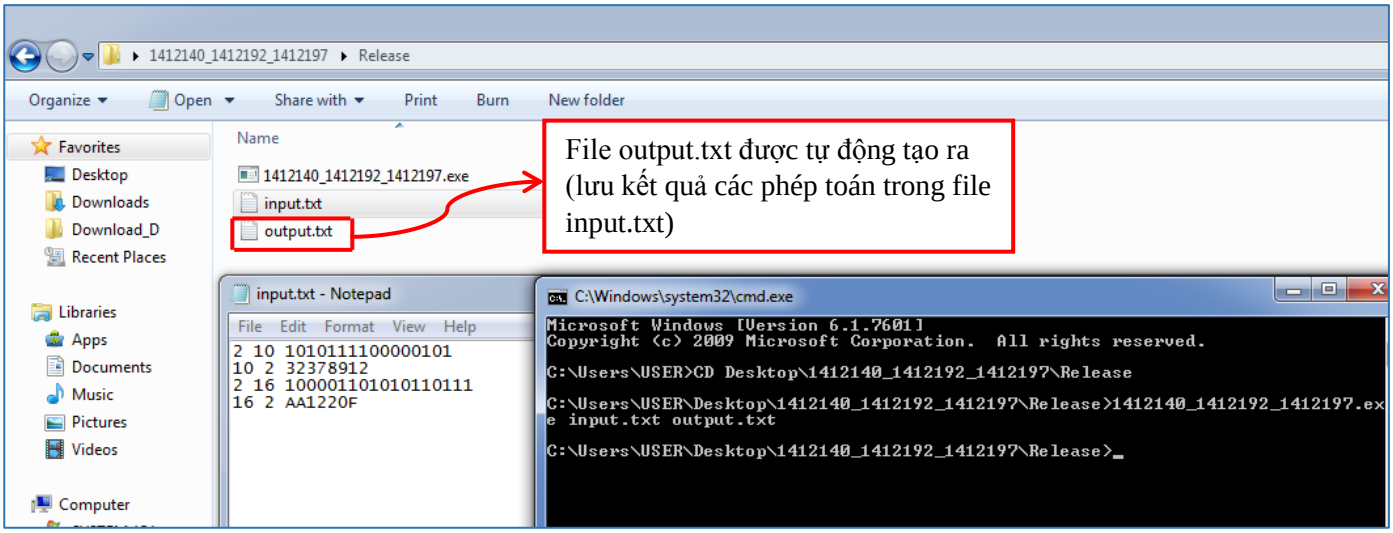


Hình 1.1. Biên dịch chương trình thành công



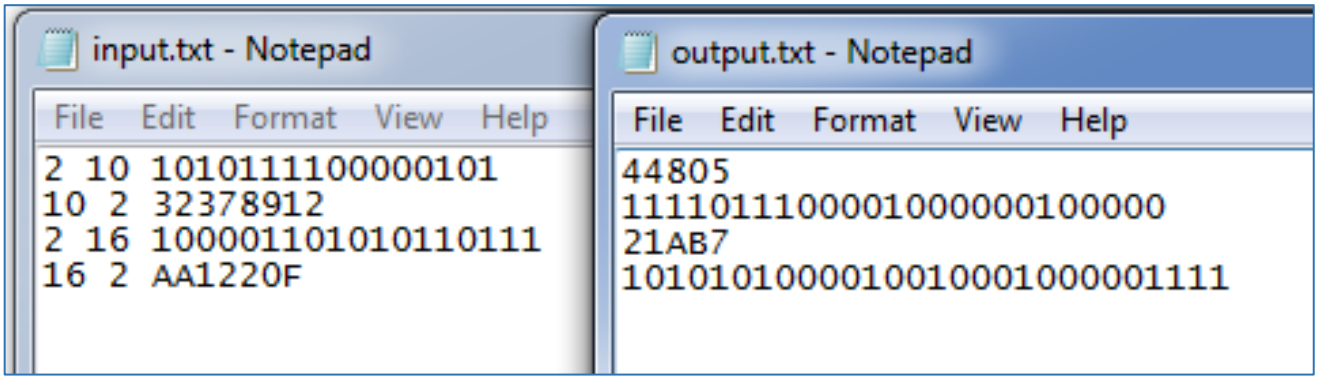


*Hình 1.2. Mở hộp thoại cmd (Command Prompt), gõ đường dẫn tới thư mục chứa file *.exe*



Hình 1.3. Tạo file input.txt, chạy tham số dòng lệnh

4.1. Chuyển đổi cơ số giữa các hệ



Hình 1.4. Thao tác với phép toán chuyển đổi cơ số của một số

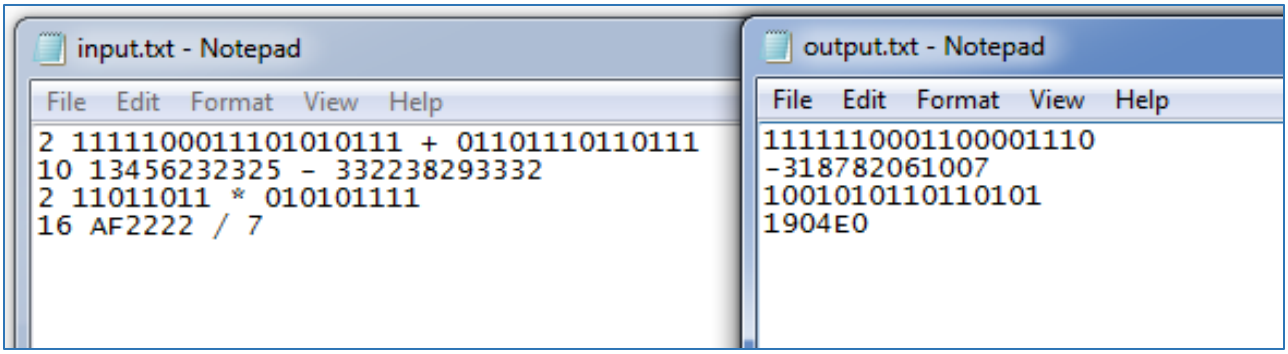
Dòng 1: Chuyển từ hệ 2 sang hệ 10

Dòng 2: Chuyển từ hệ 10 sang hệ 2

Dòng 3: Chuyển từ hệ 2 sang hệ 16

Dòng 4: Chuyển từ hệ 16 sang hệ 2

4.2. Thực hiện các phép cộng, trừ, nhân, chia



Hình 1.5. Thao tác với các phép toán cộng "+", trừ "-" , nhân "", chia "/"*

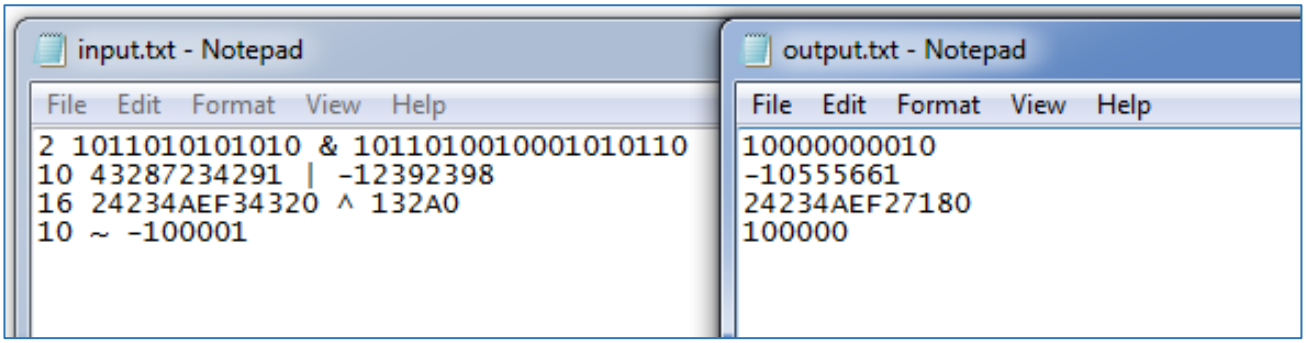
Dòng 1: Phép cộng "+" trong hệ 2

Dòng 2: Phép trừ "-" trong hệ 10

Dòng 3: Phép nhân "*" trong hệ 2

Dòng 4: Phép chia "/" trong hệ 16

4.3. Các toán tử AND, OR, XOR, NOT



Hình 1.6. Thao tác với các phép toán and "&", or "|", xor "^", not "~"

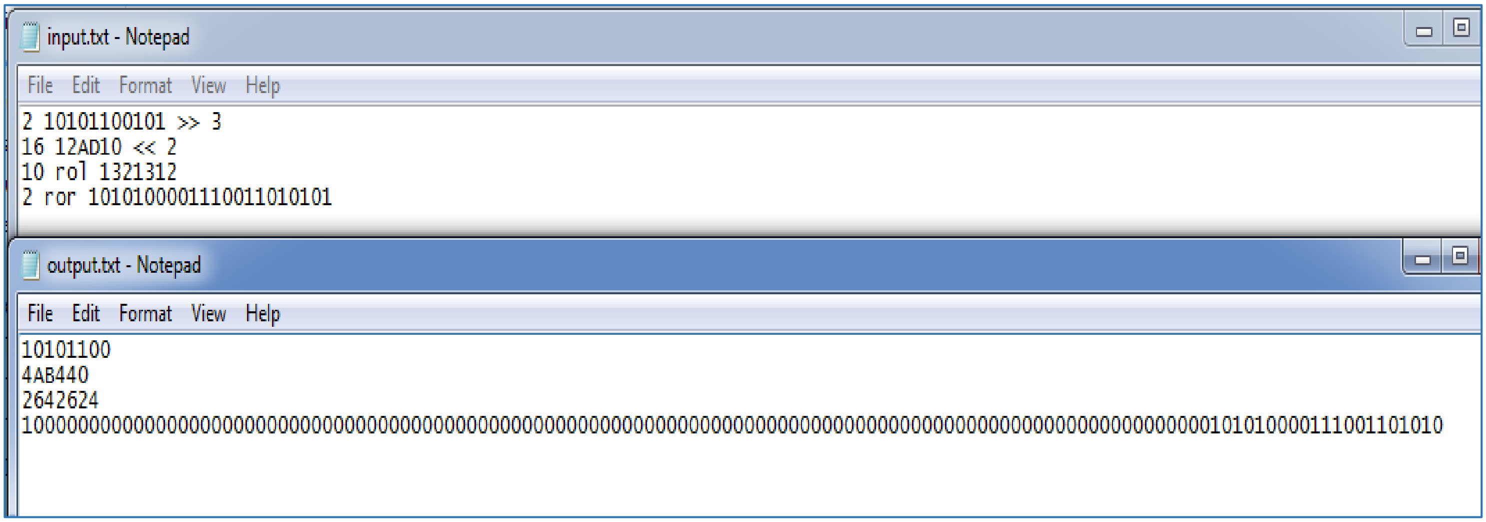
Dòng 1: Phép and "&" trong hệ 2

Dòng 2: Phép or "|" trong hệ 10

Dòng 3: Phép xor "^" trong hệ 16

Dòng 4: Phép not "~" trong hệ 10

4.4. Các thao tác dịch trái, dịch phải, xoay trái, xoay phải



Hình 1.7. Thao tác với các phép toán dịch trái "<<", dịch phải ">>", xoay trái "rol", xoay phải "ror"

Dòng 1: Dịch phải trong hệ 2

Dòng 2: Dịch trái trong hệ 16

Dòng 3: Xoay trái trong hệ 10

Dòng 4: Xoay phải trong hệ 2

5. Đánh giá quá trình thực hiện

Chức năng	Mức độ hoàn thành
Chuyển đổi số QInt từ hệ thập phân sang hệ nhị phân và ngược lại	Hoàn thành
Chuyển đổi số QInt từ hệ nhị phân sang thập lục phân và ngược lại	Hoàn thành
Các operator =, operator +, operator -, operator *, operator /	Hoàn thành
Các toán tử AND "&", OR "|", XOR "^", NOT "~"	Hoàn thành
Các toán tử: dịch trái "<<", dịch phải ">>"	Hoàn thành
Các phép xoay trái "rol", xoay phải "ror"	Hoàn thành

- Đánh giá chung: Đã hoàn thành 100% đồ án

6. Các nguồn tài liệu tham khảo

- Sách Lập trình hướng đối tượng - Trần Đan Thư, Đinh Bá Tiến, N. T. T. Minh Khang

- Wikipedia………

Ví dụ: https://vi.wikipedia.org/wiki/Ph%C3%A9p_to%C3%A1n_thao_t%C3%A1c_bit

https://vi.wikipedia.org/wiki/H%E1%BB%87_nh%E1%BB%8B_ph%C3%A2n