

# BÁO CÁO ĐỒ ÁN 1

MÔN: KIẾN TRÚC MÁY TÍNH VÀ HỢP NGỮ

## THÀNH VIÊN NHÓM:

1612700 Nguyễn Ngô Tín

1612714 Nguyễn Hồng Tới

1612726 Nguyễn Minh Trí

## **Mục Lục:**

<b>I. Thông tin thành viên và bảng phân công công việc:</b>	2
<b>II. Nội dung đồ án:</b>	3
1. Môi trường lập trình:	3
2. Ý tưởng:	3
3. Các chức năng đã hoàn thành:	4
a) Nhập dữ liệu:	4
b) Xuất dữ liệu:	4
c) Hàm chuyển đổi Qint từ thập phân sang nhị phân:	4
d) Hàm chuyển đổi Qint từ nhị phân sang thập phân:	4
e) Hàm chuyển đổi Qint từ nhị phân sang thập lục phân:	5
f) Hàm chuyển đổi Qint từ thập lục phân sang nhị phân:	6
g) Các operator toán tử:	7
h) Các hàm toán tử:	8
4. Test Case:	10
<b>III. Tham Khảo:</b>	12

## I. Thông tin thành viên và bảng phân công công việc:

MSSV	Họ và tên	Công việc	Đóng góp	Hoàn thành	Ký tên
1612700	Nguyễn Ngô Tín	<ul style="list-style-type: none"><li>- Lên ý tưởng và tìm kiếm tài liệu</li><li>- Viết hàm chuyển đổi và nhập dữ liệu:<ul style="list-style-type: none"><li>• BinToQint</li><li>• DexToQint</li><li>• HexToQint</li><li>• Chuyển từ hệ thập phân sang hệ nhị phân và ngược lại</li><li>• Chuyển đổi số QInt từ hệ nhị phân sang hệ thập lục phân và ngược lại</li></ul></li></ul>	33%	100%	
1612714	Nguyễn Hồng Tới	<ul style="list-style-type: none"><li>- Viết báo cáo</li><li>- Viết các toán tử:<ul style="list-style-type: none"><li>• AND “&amp;”</li><li>• OR “ ”</li><li>• XOR “^”</li><li>• NOT “~”</li><li>• Dịch trái “&lt;&lt;”</li><li>• Dịch phải “&gt;&gt;”</li><li>• Xoay trái “rol”</li><li>• Xoay phải “ror”</li></ul></li><li>- Tổng hợp source</li></ul>	33%	100%	
1612726	Nguyễn Minh Trí	<ul style="list-style-type: none"><li>- Viết các toán tử<ul style="list-style-type: none"><li>• Operator +</li><li>• Operator –</li><li>• Operator *</li><li>• Operator /</li></ul></li><li>- Viết hàm main</li><li>- Xử lý input, output</li><li>- Xử lý biên dịch command line</li></ul>	33%	100%	

## II. Nội dung đồ án:

### 1. Môi trường lập trình:

- Visual Studio 2015

### 2. Ý tưởng:

- Như ta đã biết số int là kiểu số nguyên 4 byte, khi thao tác nhập xuất trên màn hình ta nhìn thấy chúng biểu diễn ở hệ thập phân nhưng thực chất bên trong máy tính chỉ hiểu đó là 32bits liên tục(chuỗi 32 kí tự 0 hoặc 1) vậy giờ ta muốn biểu diễn một số nguyên 128bits thì chúng ta sẽ dùng một mảng gồm 4 số nguyên kiểu int, khi đó ta sẽ có 1 chuỗi 128bits liên tục và xử lí trên 128 bit này
- Đầu tiên thì ta cần phải nhập xuất các số nguyên 128bits này ( Qint)  
+Nhập thì chúng ta cần chuyển chuỗi nhập ban đầu từ thập phân, thập lục phân về nhị phân. Ở phần Qint khi vừa khai báo thì 4 số trong mảng ta cho bằng 0 nên 128bits ban đầu đều bằng 0. Sau khi chuyển về chuỗi nhị phân, ta duyệt từng phần tử trong chuỗi và nếu phần tử nào bằng 1 thì ta sẽ SetBit 1 vào đúng vị trí đó trong chuỗi 128bits của Qint bằng hàm SetBit, chú ta sẽ dùng phép or để bật lên bit 1 ở vị trí cần SetBit. Về cách chuyển về hệ nhị phân từ hệ thập phân thì ta cứ chia 2 lấy dư rồi đọc ngược số dư lên, chuyển từ hệ thập lục phân sang nhị phân thì cứ 1 kí tự thành 4 bit ( ví dụ A = 1010)  
+Xuất : sau khi nhập xong, thì ta có thể lấy ra được hệ thập phân của 4 số int trong mảng( không phải là hệ thập phân của Qint) từ 4 số này ta có thể chuyển về 4 chuỗi nhị phân sau đó ghép lại và xóa số 0 thừa ở đầu thì ta có được chuỗi nhị phân, từ chuỗi nhị phân ta tiến hành gộp 4 bit lại cho 1 kí tự thì sẽ được chuỗi thập lục phân, từ chuỗi nhị phân ta tiến hành nhân 2 và cộng thêm 1 hoặc 0 thì sẽ có được chuỗi thập phân ( ví dụ string a = 1010 thì đầu tiên lấy  $0 * 2 + a[0] = 1$ , tiếp theo  $1 * 2 + a[1] = 2$ , tiếp theo  $2 * 2 + a[2] = 5$ , tiếp theo  $5 * 2 + a[3] = 10$ , ta có được 10 là biểu diễn thập phân của chuỗi 1010).
- Sau khi nhập từ hệ 2,10,16 vào Qint và xuất từ Qint ra hệ 2,10,16 thì ta có thể linh động lấy ra bất kì chuỗi 2,10,16 nào phù hợp cho các phương thức mà chúng ta cần, ví dụ như cộng trừ trên hệ 2 sẽ dễ thì ta sẽ lấy ra hệ 2 để + - sau đó lấy lưu vào Qint và có thể xuất ra lại bất kì hệ nào

- Phạm vi biểu diễn: vì sử dụng số bù 2 để biểu diễn số âm nên phạm vi biểu diễn sẽ là  $-2^{127}$  đến  $2^{127} - 1$

### 3. Các chức năng đã hoàn thành:

#### a) Nhập dữ liệu:

- Xử lý và nhập dữ liệu từ chuỗi, chuỗi có thể nhập số hệ 2, 10, 16
- Lưu dữ liệu vào Qint bằng bit 0, 1

#### b) Xuất dữ liệu:

- Xuất dữ liệu từ chuỗi sang Qint, Qint có thể là số hệ 2, 10, 16

#### c) Hàm chuyển đổi Qint từ thập phân sang nhị phân:

- `void NhapTuHe10(string a);`
- `string LayHe2();`
- Nhập chuỗi số hệ 10 vào hàm `NhapTuHe10(string)` và xuất ra hệ nhị phân bằng hàm `LayHe2()`
- Chương trình minh họa:

```

Qint.cpp
Main.cpp
Header.h

Qint (Global Scope)
26
27     return res;
28
29
30 int main(int argc, char* argv[])
31 {
32     Qint a;
33     string s;
34     cout << "Nhap so he 10: ";
35     getline(cin, s);
36     a.NhapTuHe10(s);
37     cout << "He co so 2 cua " << s << " la: " << a.LayHe2() << endl;
38
39
40 // string a;
41 // fstream f, fo;
42 // char* p;
43
44 // f.open("input.txt", ios::in);
45 // fo.open("output.txt", ios::out);
46 // while(!f.eof()){
47 //     getline(f, a);
48 //     if (demKhoangTrang(a) == 3)

```

Output window shows:

```

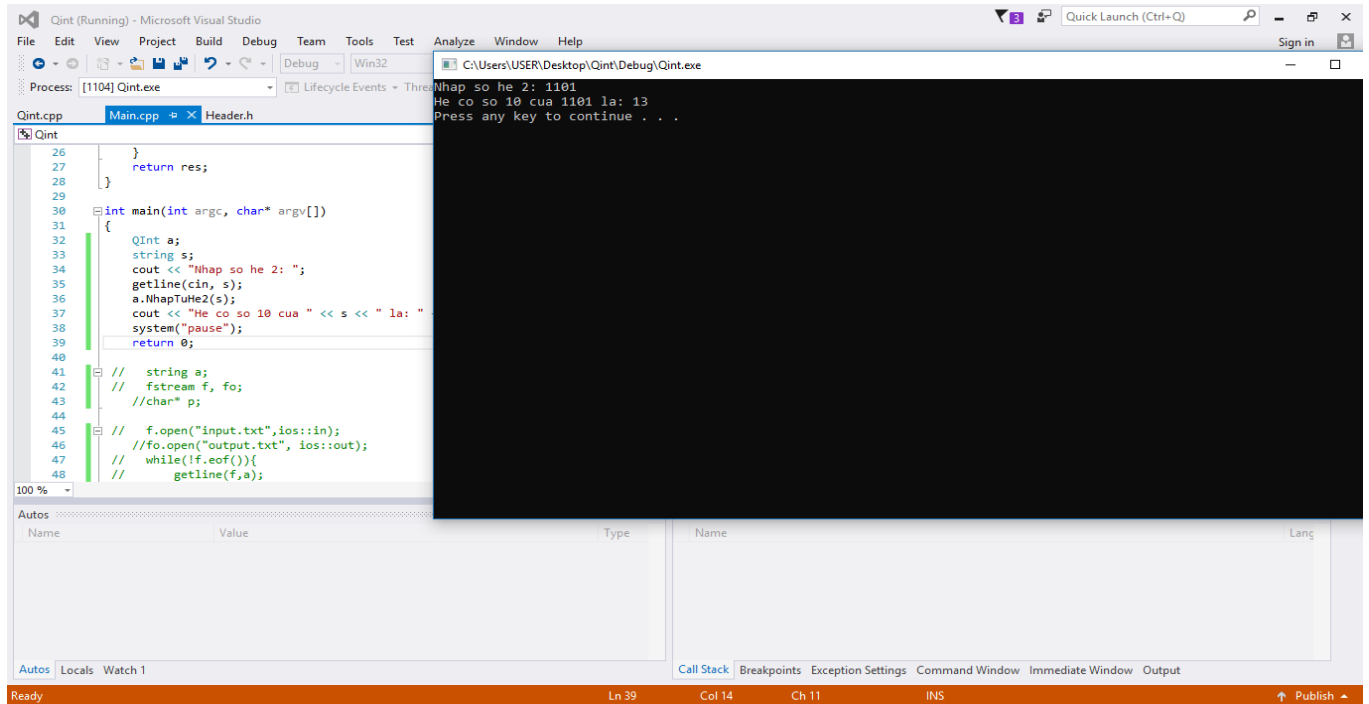
C:\Users\USER\Desktop\Qint\Debug\Qint.exe
Nhap so he 10: 10
He co so 2 cua 10 la: 1010

```

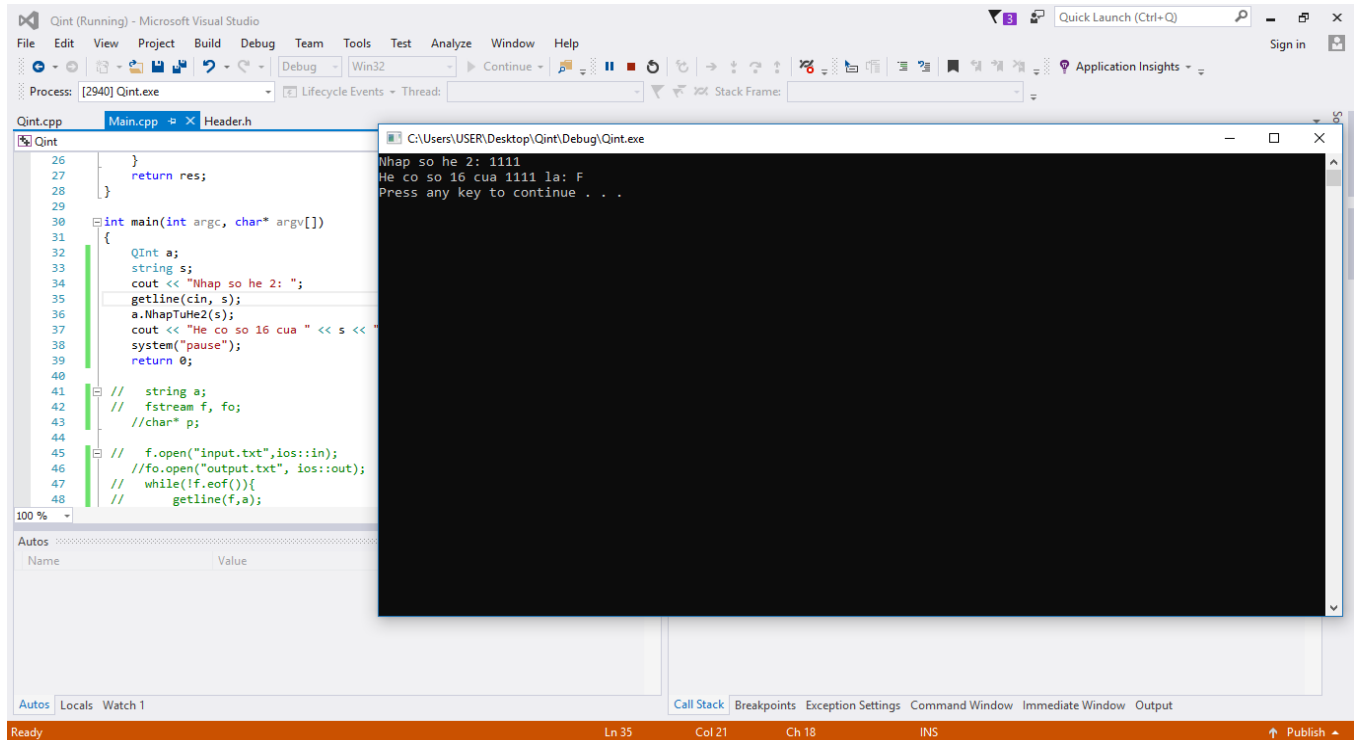
#### d) Hàm chuyển đổi Qint từ nhị phân sang thập phân:

- `void NhapTuHe2(string a);`
- `string LayHe10();`

- Nhập chuỗi số hệ 2 vào hàm NhapTuHe2(string) và xuất ra hệ nhị phân bằng hàm LayHe10()
- Chương trình minh họa:

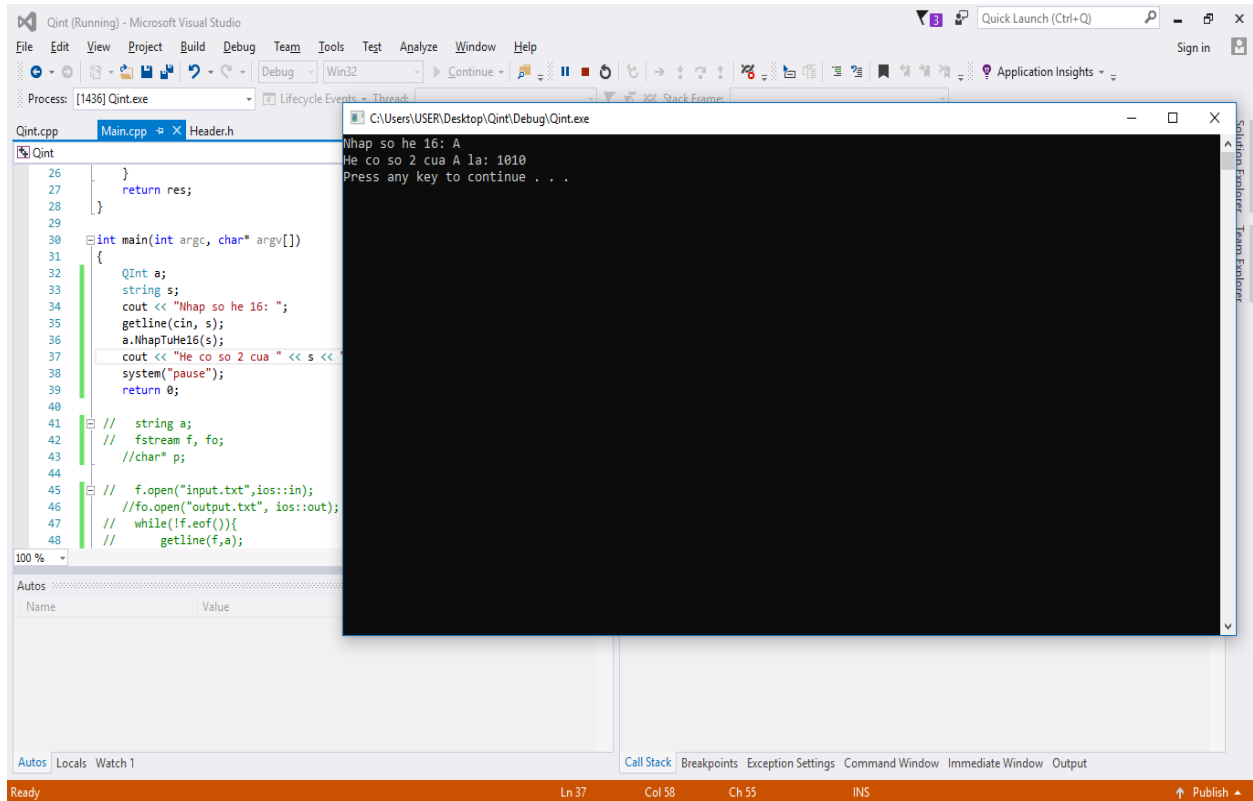


- e) Hàm chuyển đổi Qint từ nhị phân sang thập lục phân:
- void NhapTuHe2(string a);
  - string LayHe16();
  - Nhập chuỗi số hệ 2 vào hàm NhapTuHe2(string) và xuất ra hệ nhị phân bằng hàm LayHe16()
  - Chương trình minh họa:



f) Hàm chuyển đổi QInt từ thập lục phân sang nhị phân:

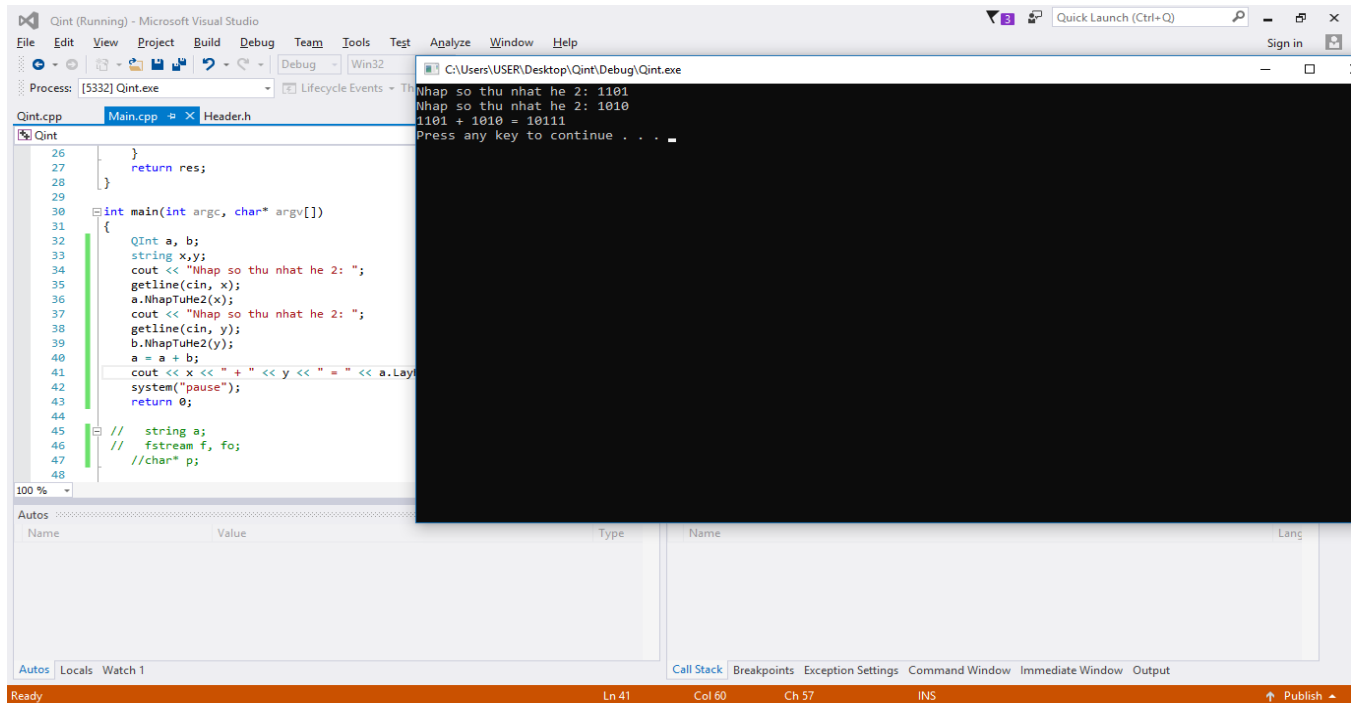
- `void` NhapTuHe16(`string` a);
- `string` LayHe2();
- Nhập chuỗi số hệ 16 vào hàm NhapTuHe16(`string`) và xuất ra hệ nhị phân bằng hàm LayHe2()
- Chương trình minh họa:



g) Các operator toán tử:

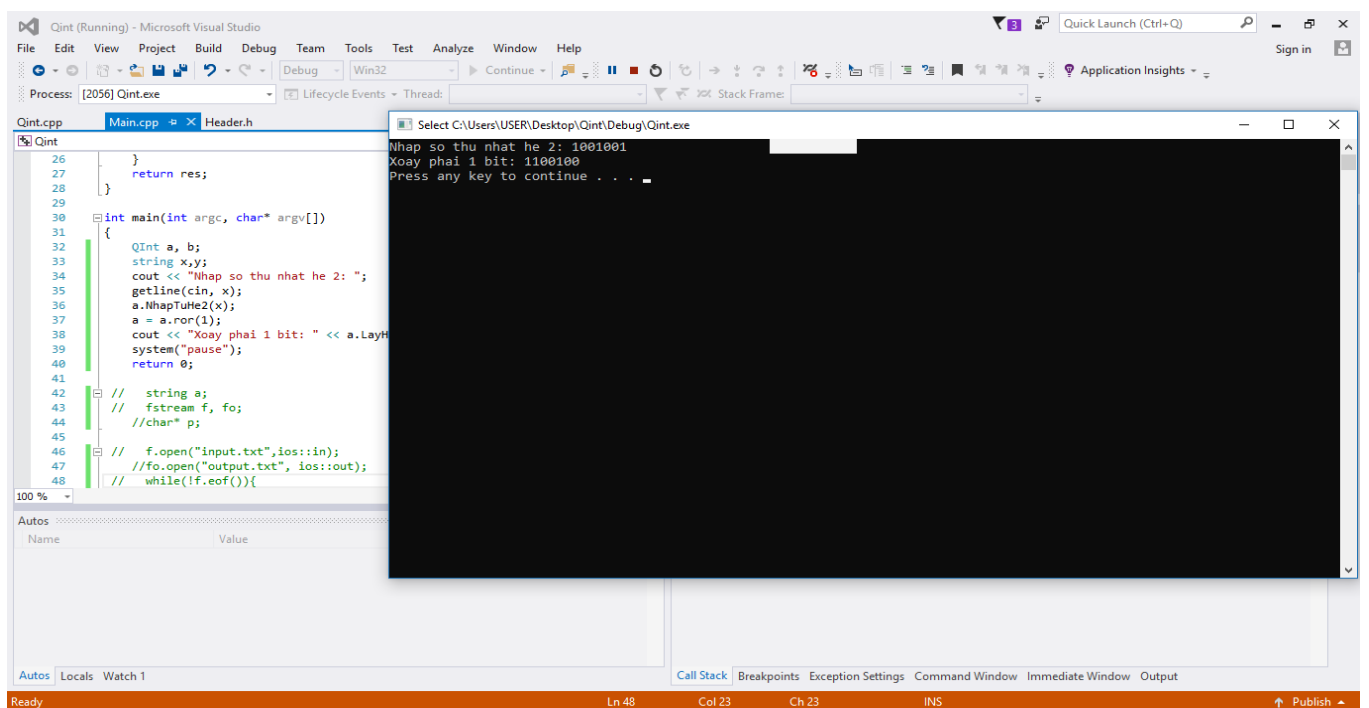
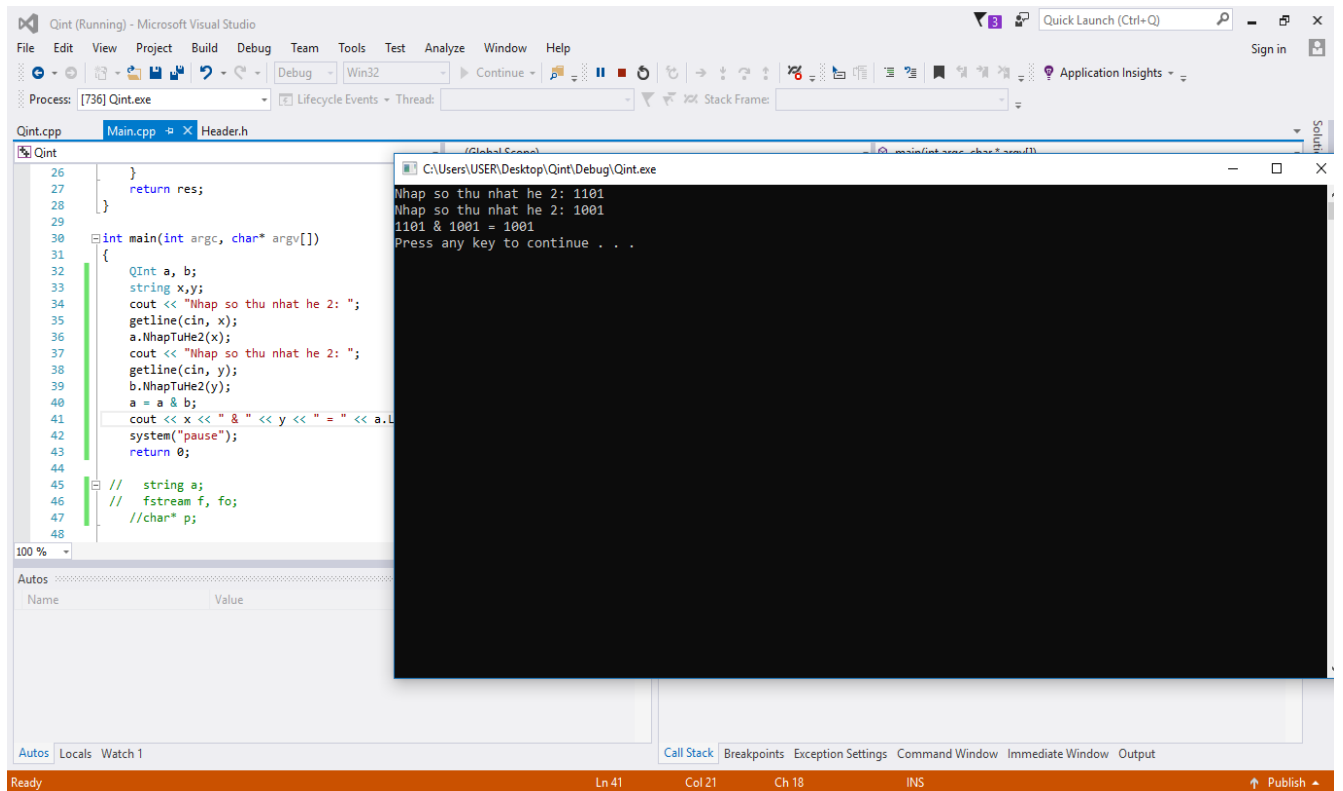
- **QInt operator +(QInt b);**
- **QInt operator -(QInt b);**
- **QInt operator \*(QInt b);**
- **QInt operator /(QInt b);**
- +, -, \*, / ( thực hiện các toán tử trên hệ cơ số 2)
- Chương trình minh họa: (operator +)





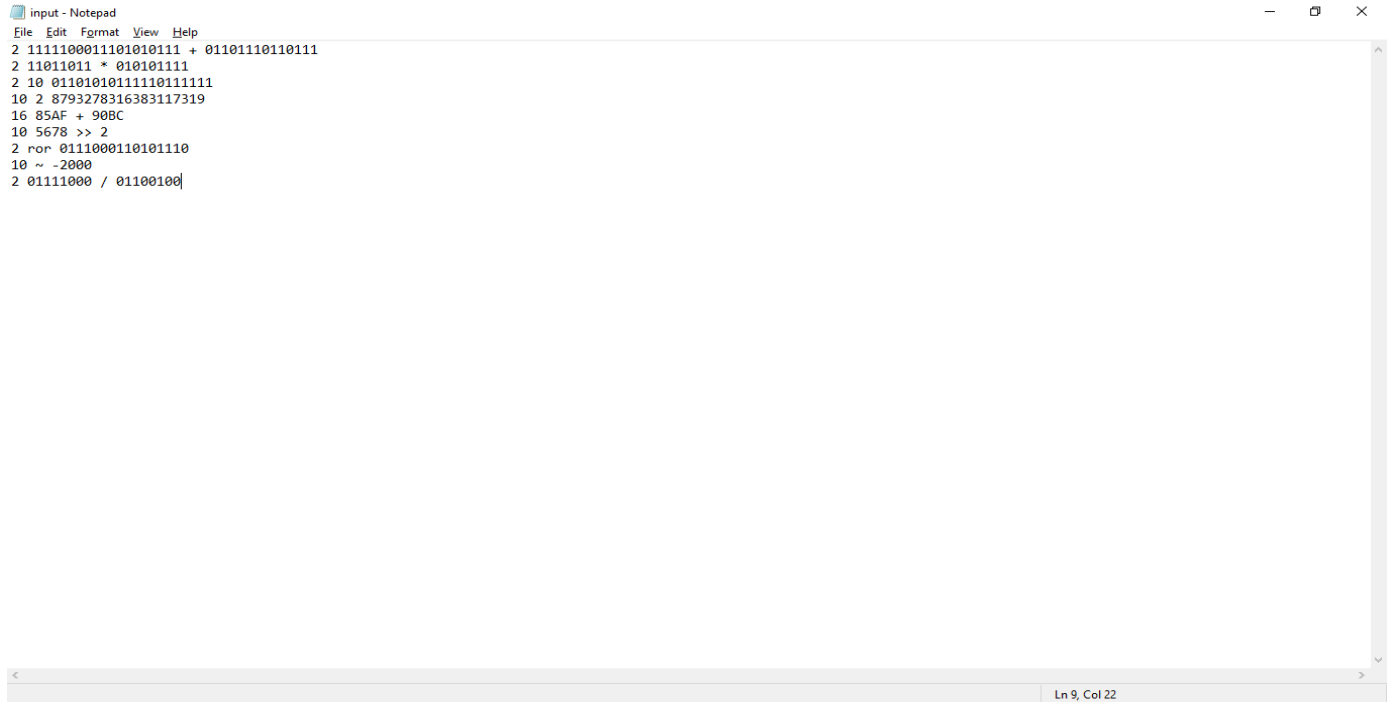
## h) Các hàm toán tử:

- `QInt operator&(QInt b);`
- `QInt operator|(QInt b);`
- `QInt operator^(QInt b);`
- `QInt operator~(QInt b);`
- `QInt operator>>(QInt b);`
- `QInt operator<<(QInt b);`
- `QInt rol(int k);`
- `QInt ror(int k);`
- AND “&”, OR “|”, XOR “^”, NOT “~”, dịch trái “<<”, dịch phải “>>”, xoay trái “rol”, xoay phải “ror” thực hiện trên hệ cơ số 2.
- Chương trình minh họa: ( operator & và ror )



## 4. Test Case:

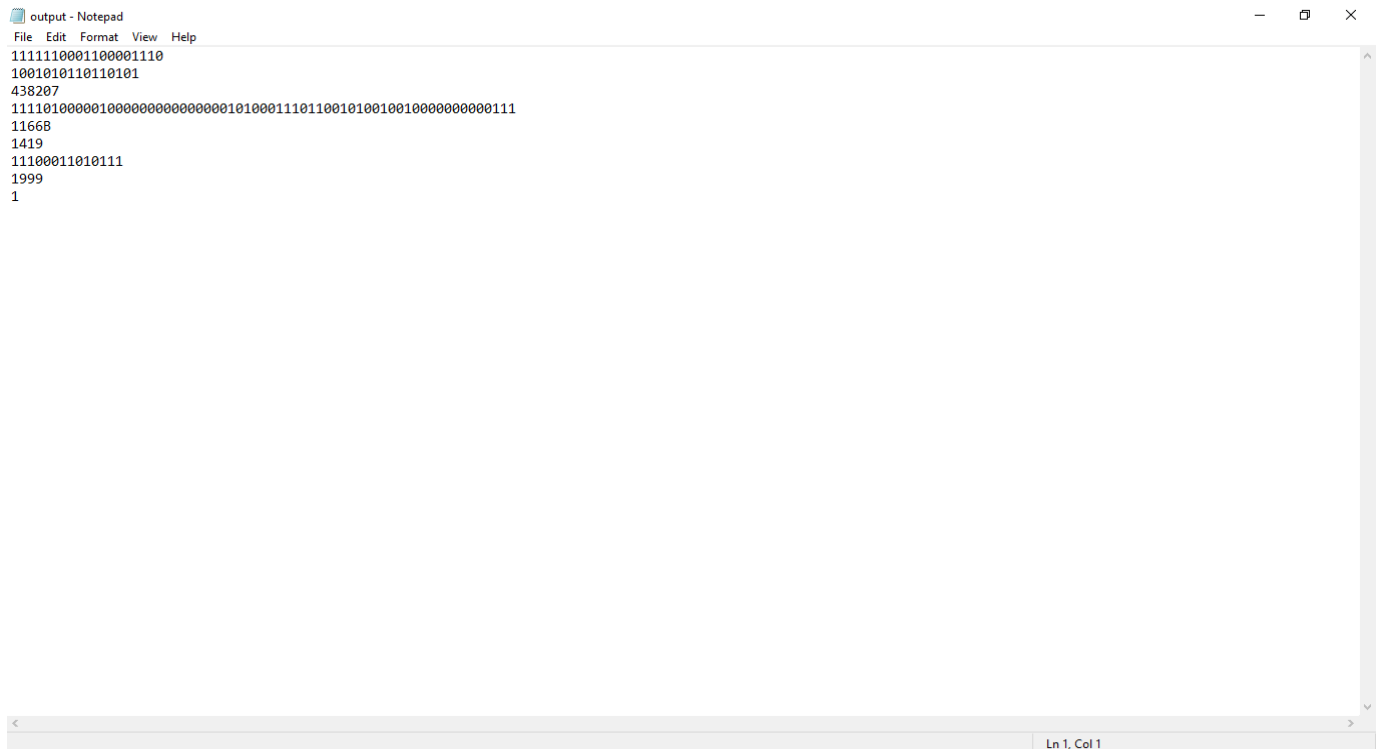
- File input.txt



```
input - Notepad
File Edit Format View Help
2 111110001110101011 + 01101110110111
2 11011011 * 01010111
2 10 01101010111110111111
10 2 8793278316383117319
16 85AF + 90BC
10 5678 >> 2
2 nor 0111000110101110
10 ~ -2000
2 01111000 / 01100100
```

Ln 9, Col 22

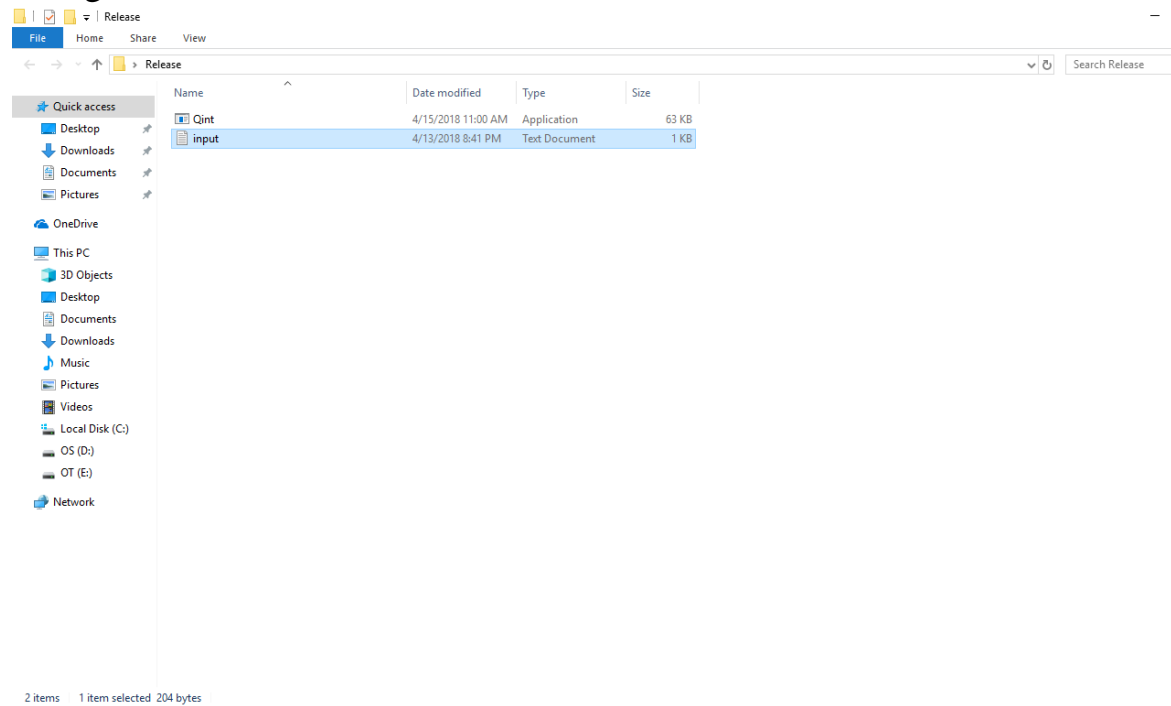
- File output.txt



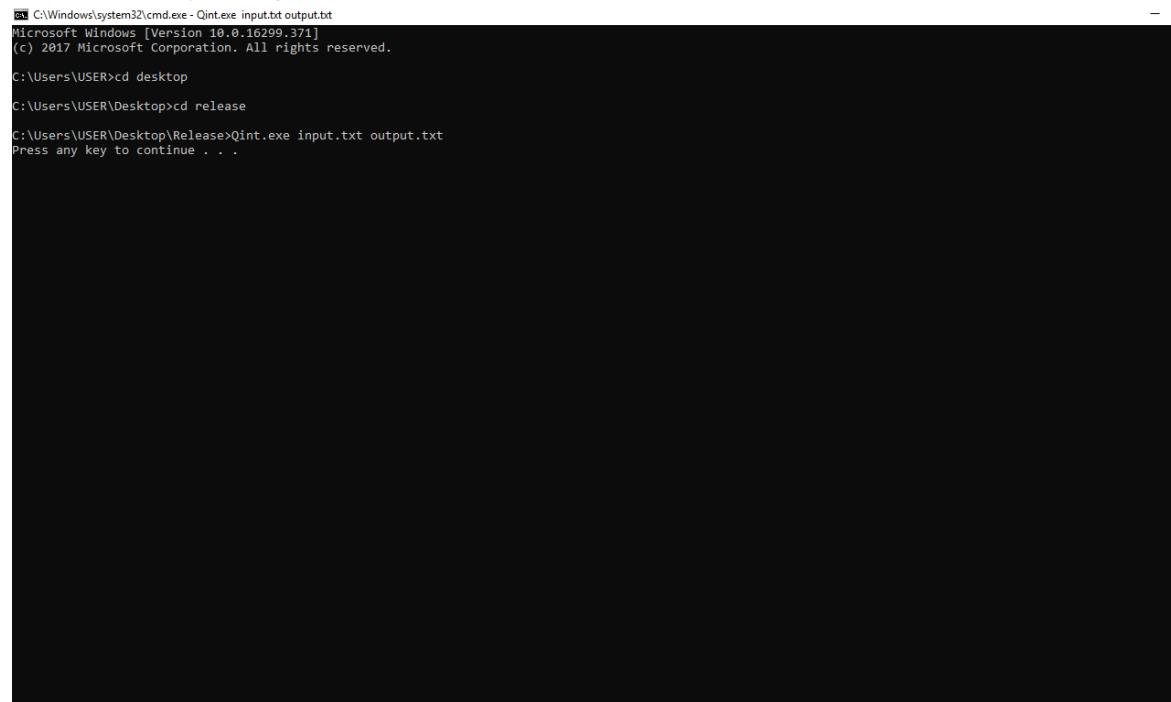
```
output - Notepad
File Edit Format View Help
1111110001100001110
1001010110110101
438207
111101000001000000000000000101000111011001001001000000000111
1166B
1419
11100011010111
1999
1
```

Ln 1, Col 1

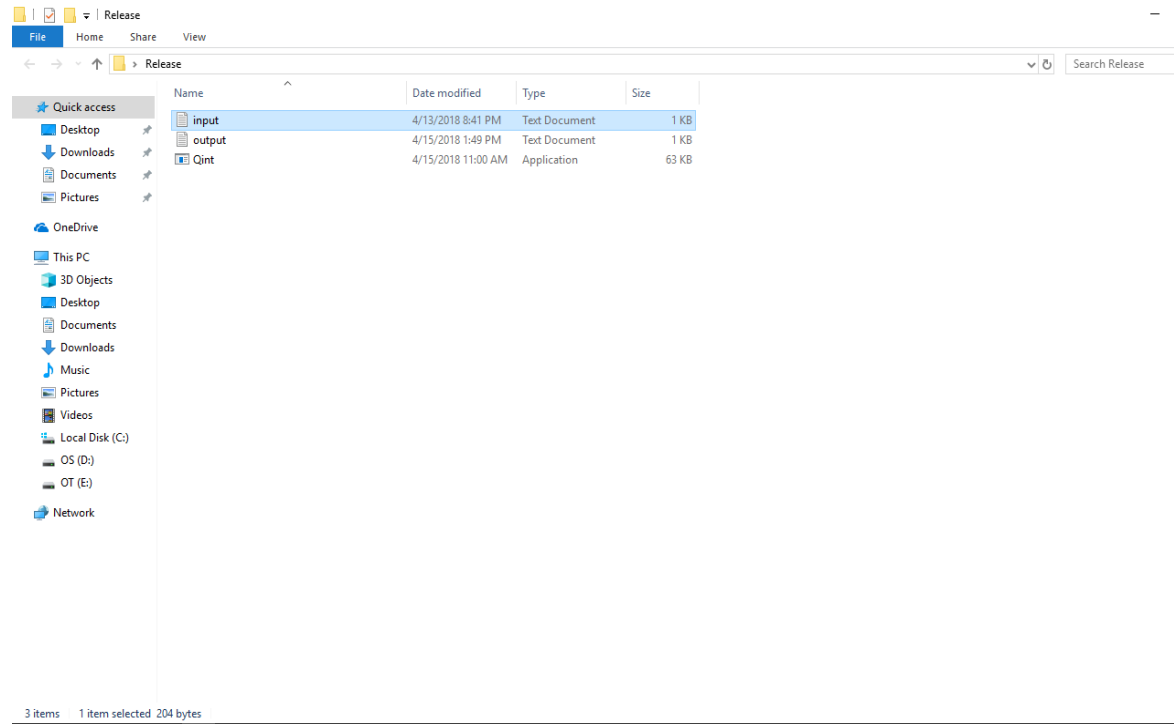
- Chạy bằng command line: (File release)
  - Trong File release:



- Sau khi chạy xong command line:



- File release sau khi chạy xong command line:



### III. Tham Khảo:

- [1] [stackoverflow.com](https://stackoverflow.com)
- [2] [daynhauhoc.com](https://daynhauhoc.com)
- [3] [www.cplusplus.com](https://www.cplusplus.com)