

**TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN**  
**KHOA CÔNG NGHỆ THÔNG TIN**

TPHCM, 2019



# **BIỂU DIỄN VÀ TÍNH TOÁN SỐ HỌC TRÊN MÁY TÍNH**

**MÔN: KIẾN TRÚC MÁY TÍNH VÀ HỢP NGỮ**

HUỲNH VĂN TÚ - 1712856

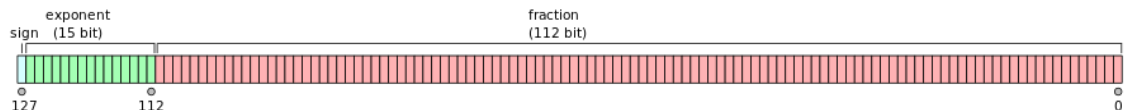
BÙI THỊ CẨM NHUNG – 1712645

NGUYỄN NGỌC BĂNG TÂM – 1712747

## Chương 1. MỞ ĐẦU

### 1.1. Mô tả đồ án

- Thiết kế kiểu dữ liệu QInt - số nguyên lớn có dấu với độ lớn 16 bytes, tức 128 bits.
- Thiết kế kiểu dữ liệu QFloat - số chấm động có độ chính xác cao với độ lớn 128 bits có cấu trúc như sau:



### 1.2. Đánh giá mức độ hoàn thành

Nhìn chung, nhóm đã hoàn thành 100% chức năng được yêu cầu. Cụ thể:

Chức năng	QInt	QFloat
Hàm nhập	X	X
Hàm xuất	X	X
Hàm chuyển đổi nhị phân sang thập phân	X	X
Hàm chuyển đổi thập phân sang nhị phân	X	X
Hàm chuyển đổi nhị phân sang thập lục phân	X	
Hàm chuyển đổi thập phân sang thập lục phân	X	
Các toán tử: “+”, “-”, “*”, “/”	X	X
Các toán tử so sánh và phép gán: “<”, “>”, “==”, “<=”, “>=”, “=”	X	
Các toán tử: AND “&”, OR “ ”, XOR “^”, NOT “~”	X	
Các toán tử: dịch trái “<<”, dịch phải “>>”, xoay trái “rol”, xoay phải “ror”	X	
Chương trình minh họa (console + tham số dòng lệnh)	X	X

## Chương 2. NỘI DUNG ĐỒ ÁN

### 2.1. Sơ đồ UML

Sơ đồ UML thể hiện thiết kế của lớp QInt (128-bit) và QFloat (128-bit).

QInt	QFloat
-data: int[4]	-data: unsigned short[8]
+QInt() +QInt(const QInt&) -lastBit(): bool -firstBit(): bool -changeBit(int pos, bool value): void -fillOnes(): void -strDiv2(string&): void -strMul2(string& s, int times): void -convertTo2sComplement(bool*): bool* +operator=(const QInt&): QInt& +operator<(const QInt&): bool +operator>(const QInt&): bool +operator<=(const QInt&): bool +operator>=(const QInt&): bool +operator==(const QInt&): bool +operator!=(const QInt&): bool +operator+(const QInt&): QInt +operator-(const QInt&): QInt +operator*(const QInt&): QInt +operator/(const QInt&): QInt +operator&(const QInt&): QInt +operator (const QInt&): QInt +operator^(const QInt&): QInt +operator~(): QInt +operator>>(int): QInt +operator<<(int): QInt +ror(int): QInt +rol(int): QInt +decToBin(): bool* +binToDec(bool*): QInt& +decToHex(): string +binToHex(bool*): string +hexToBin(string): bool* +hexToDec(string): QInt& +scanQInt(string): void +printQInt(): string «friend»+operator>>(istream&, QInt&): istream& «friend»+operator<<(ostream&, const QInt&): ostream& +addBitsArray(const bool*, const bool*): bool* +addStrings(const string&, const string&): string	+QFloat() +QFloat(const QFloat&) +firstBit(): bool +changeBit(int pos, bool value): void +exponent(): int +strDiv2(string&): void +strMulN(string& s, int, int, int): void +fracMul2(string&): bool +convertTo2sComplement(bool*, int): bool* +convertToBias(int): bool* +combineBits(bool, int, bool*, int, int): bool* +shiftRight(bool*, int, int, int): void +shiftLeft(bool*, int, int, int): void +checkOverflow(bool*, int, int, int&): bool +checkUnderflow(bool*, int, int, int, int&): bool +operator+(const QFloat&): QFloat +operator-(const QFloat&): QFloat +operator*(const QFloat&): QFloat +operator/(const QFloat&): QFloat +decToBin(): bool* +binToDec(bool*): QFloat& +scanQFloat(string): void +printQFloat(): string «friend»+operator>>(istream&, QFloat&): istream& «friend»+operator<<(ostream&, const QFloat&): ostream& +addBitsArray(const bool*, const bool*, int): bool* +subtractBitsArray(const bool*, const bool*, int): bool* +addString(const string&, const string&): string

Hình 1. Sơ đồ UML của lớp QInt và QFloat

## 2.2. Phạm vi biểu diễn

### 2.2.1. Phạm vi biểu diễn của kiểu dữ liệu QInt (128-bit):

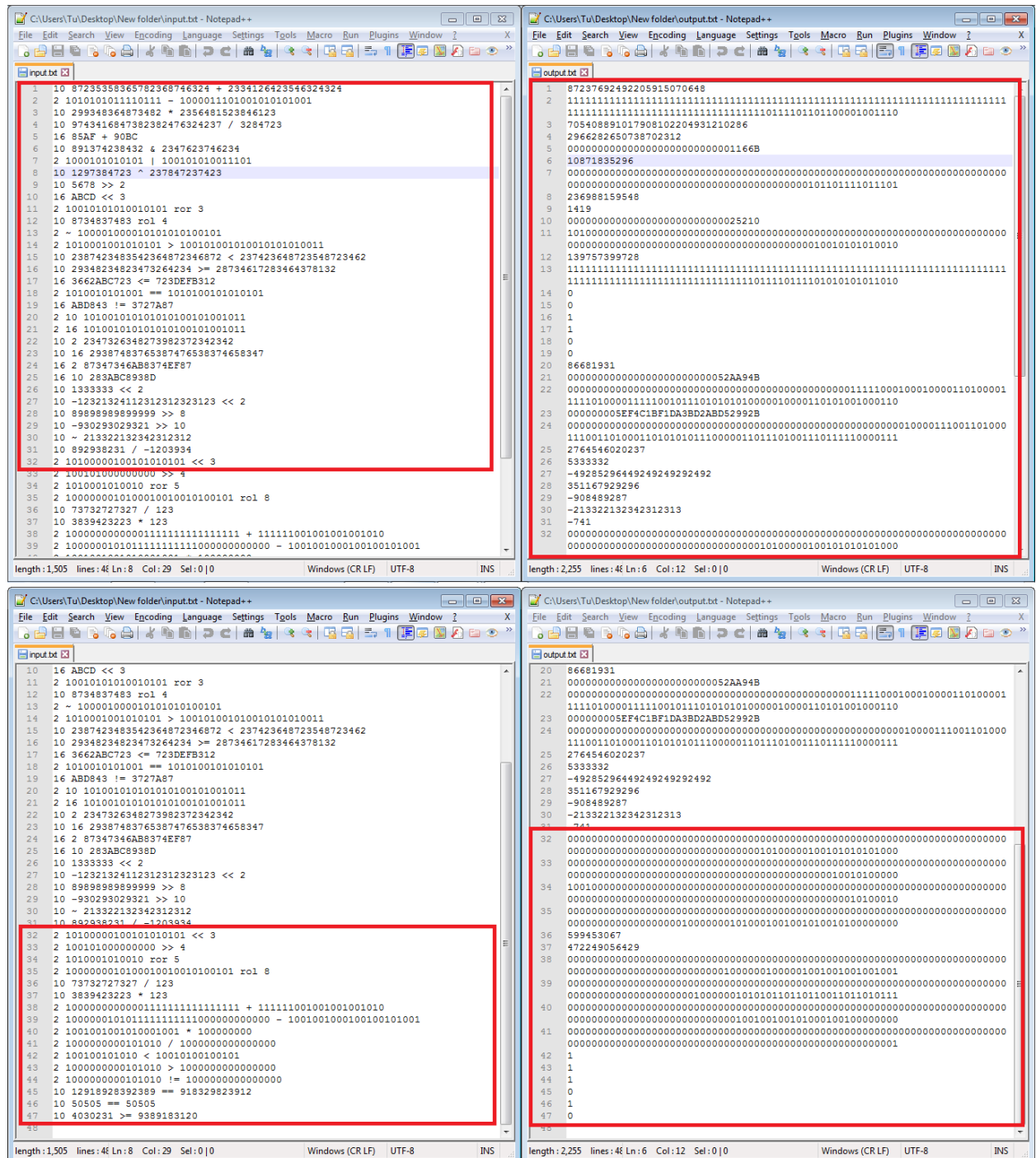
Số	Biểu diễn nhị phân	Giá trị
Max	011111111...11111111	$2^{127} - 1$
Min	100000000...00000000	$-2^{127}$

### 2.2.2. Phạm vi biểu diễn của kiểu dữ liệu QFloat (128-bit):

Số	Dấu	Mũ	Trị	Giá trị
+0	0	0000...0000	0000...0000	+0.0
-0	1	0000...0000	0000...0000	-0.0
+INF	0	1111...1111	0000...0000	+INF
-INF	1	1111...1111	0000...0000	-INF
NAN	0/1	1111...1111	Dãy bit khác 0	NAN
Chuẩn max (+)	0	1111...1110	1111...1111	$(2 - 2^{-112}) \times 2^{16383}$
Chuẩn min (+)	0	0000...0001	0000...0000	$2^{-16382}$
Chuẩn max (-)	1	0000...0001	0000...0000	$-2^{-16382}$
Chuẩn min (-)	1	1111...1110	1111...1111	$-(2 - 2^{-112}) \times 2^{16383}$
Không chuẩn max (+)	0	0000...0000	1111...1111	$(1 - 2^{-112}) \times 2^{-16382}$
Không chuẩn min (+)	0	0000...0000	0000...0001	$2^{-16494}$
Không chuẩn max (-)	1	0000...0000	0000...0001	$-2^{-16494}$
Không chuẩn min (-)	1	0000...0000	1111...1111	$-(1 - 2^{-112}) \times 2^{-16382}$

### Chương 3. THỬ NGHIỆM

### 3.1. Thử nghiệm trên kiểu dữ liệu QInt (128-bit):



### 3.2. Thử nghiệm trên kiểu dữ liệu QFloat (128-bit):





The image shows two side-by-side Notepad++ windows. The left window, titled 'C:\Users\Tu\Desktop\New folder (2)\input.txt - Notepad++', displays a long string of binary data (0s and 1s) with line numbers on the left. A red box highlights a section of the data starting around line 42. The right window, titled 'C:\Users\Tu\Desktop\New folder (2)\output.txt - Notepad++', displays the processed binary data, also with line numbers on the left. A red box highlights a section of the data starting around line 47. Both windows show a status bar at the bottom indicating file length, line count, and encoding (UTF-8).



## Chương 4. TÀI LIỆU THAM KHẢO

1. Phạm Tuấn Sơn, Lecture notes: “*Biểu diễn số nguyên*”
2. Phạm Tuấn Sơn, Lecture notes: “*Số chấm động*”
3. Wikipedia, “*Floating-point arithmetic*”
4. Colorado State University, Lecture notes: “*Floating Point Addition Example*”
5. University of Wisconsin-Madison, Lecture notes: “*Floating Point Arithmetic*”