

BÁO CÁO ĐỒ ÁN

BIỂU DIỄN VÀ TÍNH TOÁN SỐ HỌC
TRÊN MÁY TÍNH

Giảng viên phụ trách: Lê Viết Long

Thành viên nhóm:

1. Võ Quốc Thắng - 1712162
2. Lê Nguyễn Nhật Trường - 1712195
3. Lê Tuấn Đạt - 1712329

MỤC LỤC

1. GIỚI THIỆU	2
1.1. Nội dung và yêu cầu	2
1.2. Phân tích đồ án	2
1.2.1. QInt:	2
1.2.2. QFloat	2
1.2.3. Giao diện	2
1.3. Thông tin nhóm	2
1.3.1 Thông tin thành viên	2
1.3.2 Phân công công việc	3
2. QINT	4
2.1. Mô tả cấu trúc lưu trữ	4
2.2. Sơ đồ lớp	4
2.3. Các toán tử cộng, trừ, nhân và chia:	5
2.4. Phạm vi biểu diễn	6
3. QFLOAT	6
3.1. Mô tả cấu trúc lưu trữ	6
3.2. Sơ đồ lớp	7
3.3. Một số hàm chính	7
3.4. Phạm vi biểu diễn	9
4. GIAO DIỆN VÀ DEMO	9
4.1. Giao diện chương trình	9
4.1.1. Màn hình khởi động	9
4.1.2. QInt Calculator	10
4.1.3. QFloat Calculator	11
4.2. Test cases	11
4.2.1. QInt test	11
4.2.2. QFloat test	12
4.3. Link Demo	13
5. TỔNG KẾT	13
6. ĐÁNH GIÁ	13
6.1. Đánh giá kết quả đồ án	13
6.2. Đánh giá các thành viên	14
TÀI LIỆU THAM KHẢO	16

1. GIỚI THIỆU

1.1. Nội dung và yêu cầu

Sử dụng kiến thức đã học, xây dựng kiểu dữ liệu số lớn và thực hiện các phép tính trên đó, bao gồm:

- QInt
- QFloat

Yêu cầu nâng cao

- Tạo giao diện cho chương trình
- Trang bị các toán tử cho QFloat

1.2. Phân tích đồ án

1.2.1. QInt:

Sử dụng bốn biến mang kiểu dữ liệu **unsigned int (32bit)** để biểu diễn đúng **128 bit của QInt**. Đầu bài đưa vào input lưu dưới dạng chuỗi, xây dựng các hàm thao tác trên chuỗi, thao tác trên bit, các phép logic và chuyển đổi qua lại giữa các hệ.

1.2.2. QFloat

Sử dụng bốn biến kiểu dữ liệu **unsigned int (32bit)** tương tự như QInt để biểu diễn 128 bit cho QFloat. Tuy nhiên khi thực hiện phép tính cần chú ý đến phần mũ trong biểu diễn số chấm động. Ngoài ra còn xử lý cho các trường hợp đặc biệt bao gồm số 0, Infinity, NaN và số không chuẩn (denormal number).

1.2.3. Giao diện

Sau khi đã cài đặt hai lớp thư viện QInt và QFloat, tiến hành xây dựng giao diện cho người dùng. Sử dụng thư viện MFC trong C++ để tạo các phím chọn và khung nhập (GUI). Ngoài ra còn hỗ trợ đọc dữ liệu từ file và trả kết quả ra file.

1.3. Thông tin nhóm

1.3.1 Thông tin thành viên

MSSV	Họ và tên	Email	Vai trò
1712162	Võ Quốc Thắng	voquocthang1999@gmail.com	Xây dựng thư viện QFloat (Developer)
1712195	Lê Nguyễn Nhật Trường	truongthk62014@gmail.com	Xây dựng QInt (Developer)

1712329	Lê Tuấn Đạt	letuandat2110@gmail.com	Xây dựng giao diện, kiểm định chương trình và tạo test (Tester)
---------	-------------	--	---

Bảng 1.1. Bảng thông tin thành viên

1.3.2 Phân công công việc

Thành viên	Công việc	Mô tả
Võ Quốc Thắng	Phân công công việc	Phân tích đề án và phân công công việc cho các thành viên
	Xây dựng cài đặt thư viện QFloat	Sử dụng các kiến thức đã học để lưu trữ, thực hiện phép tính trên các số chấm động 128 bit
	Viết báo cáo tổng kết	Báo cáo mô tả cách thức thực hiện đề án và tổng kết, đánh giá
Lê Nguyễn Nhật Trường	Xây dựng cài đặt thư viện QInt	Sử dụng các kiến thức đã học để lưu trữ, thực hiện phép tính, chuyển đổi hệ trên các số nguyên 128 bit
	Xây dựng Test cho QInt	Đưa ra một số trường hợp cho kiểu dữ liệu QInt
Lê Tuấn Đạt	Thiết kế và xây dựng giao diện cho người dùng	Sử dụng thư viện MFC để tạo nên giao diện là các Calculator phù hợp với kiểu input đầu vào. Ngoài ra còn hỗ trợ đọc và xuất file
	Xây dựng bộ test và giới hạn số	Tạo các test để thử độ đúng đắn cho chương trình. Sử dụng công cụ online có sẵn, xây dựng các trường hợp test đặc biệt

	Tổng hợp source code, chỉnh sửa để phù hợp với file input	Đưa các thư viện QInt và QFloat và giao diện MFC
--	---	--

Bảng 1.2. Bảng phân công công việc

2. QINT

2.1. Mô tả cấu trúc lưu trữ

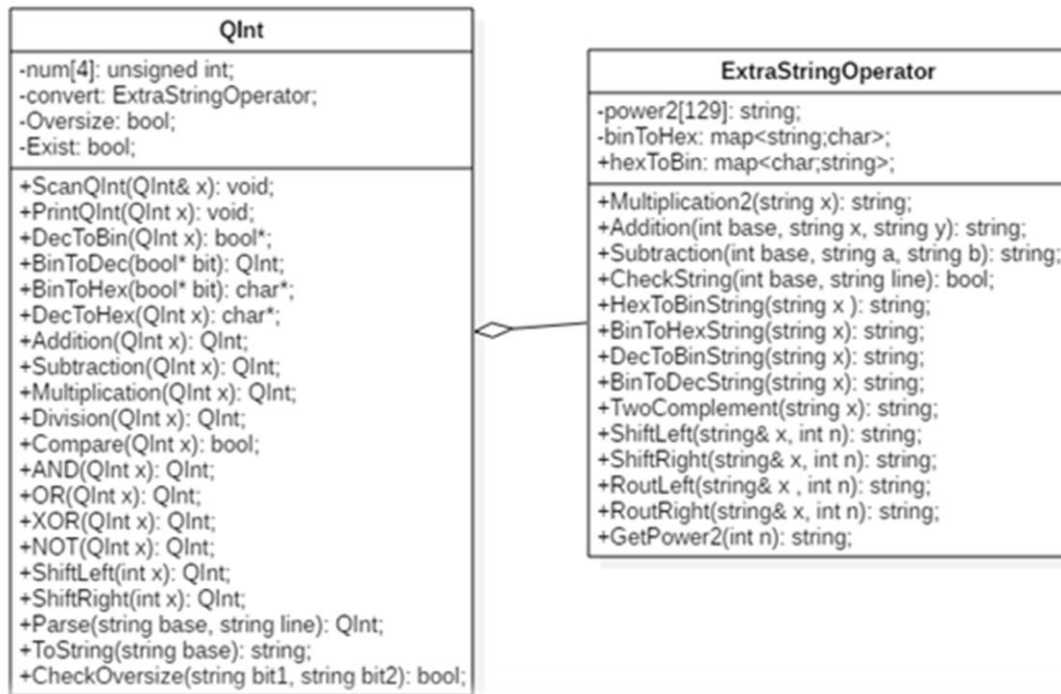
- Cấu trúc **QInt**: Để tạo kiểu dữ liệu số nguyên có dấu **128 bits** ta cần dành một bit thứ 127 để lưu dấu của số nguyên **bit thứ 127** bằng 1 là số **nguyên âm** còn ngược lại là **nguyên dương**.
- Các toán tử trong **QInt**: kiểu dữ liệu số nguyên QInt sẽ có phép toán thông thường như +, -, *, / cùng với đó các phép toán trên bit như >>, <<, AND, OR, XOR, NOT ...
- Kiểu lưu trữ QInt được biểu diễn như sau:

```
class QInt
{
// Attribute
private:
unsigned int data[4] = { 0 };
....
public:
QInt() {}
~QInt() {}
....
}
```

2.2. Sơ đồ lớp

Mô tả: Để tạo ra một kiểu dữ liệu **QInt** để lưu trữ số nguyên có dấu 128 bits. Tôi đã chọn cách tạo ra 4 class: **Query**, **Tokenizer**, **ExtraStringOperator**, **QInt**. Hai class đầu tiên khá đơn giản class **Query** hỗ trợ việc nhập xuất các truy vấn từ đề bài yêu cầu. Còn class **Tokenizer** hỗ trợ việc tách các chuỗi trong đầu vào. Tiếp đến ta cần tạo ra hai class **ExtraStringOperator**, **QInt**. Cấu trúc của hai class như sau.

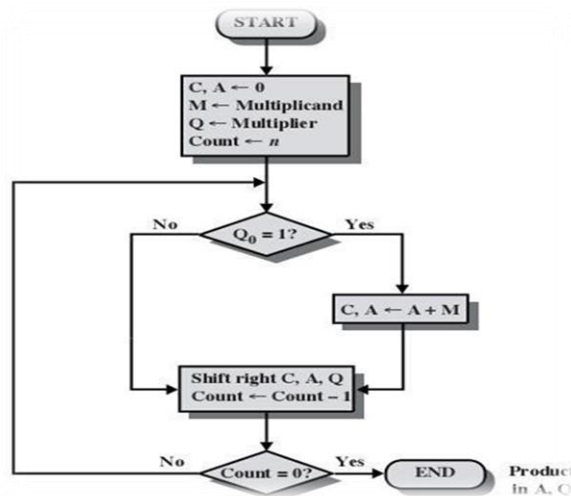
Computer Architecture Project 1



Hình 2.1. Sơ đồ lớp QInt

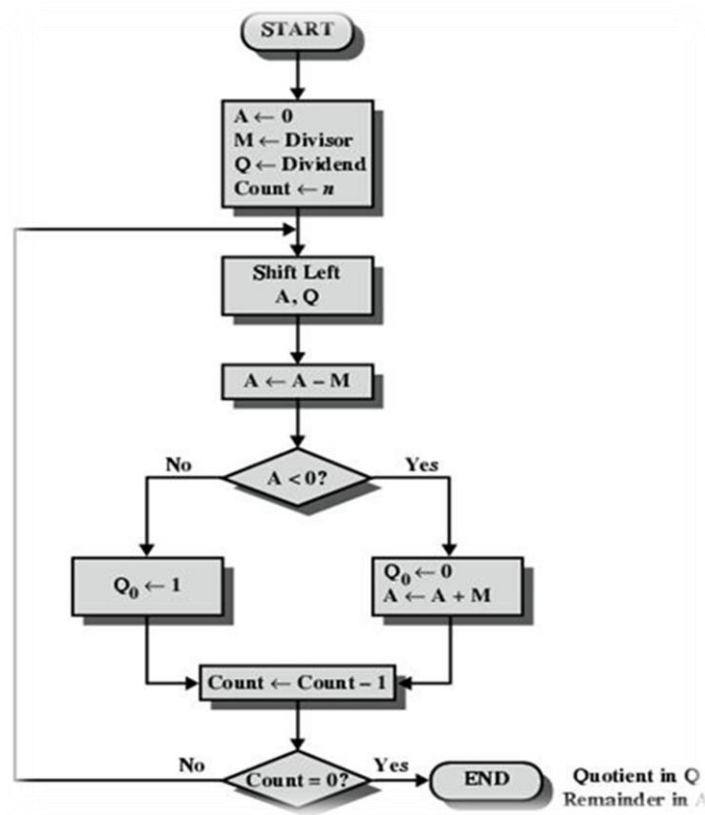
2.3. Các toán tử cộng, trừ, nhân và chia:

- **Phép cộng** thì ta chỉ cần chuyển về dạng nhị phân rồi cộng bit bình thường và ta cần check input và output xem có **Oversize** chưa.
- **Phép trừ** thì trước tiên ta của số trừ thành dạng **Two's complement** rồi làm như phép cộng.
- **Phép nhân** thì áp dụng **phép nhân** các số không dấu sao giải quyết việc thêm dấu vào.



Hình 2.2 Lưu đồ thực hiện phép nhân hai số QInt

- **Phép chia** thì áp dụng ý chia không dấu và tái sử dụng **phép trừ**.



Hình 2.3 Lưu đồ thực hiện phép chia hai số *QInt*

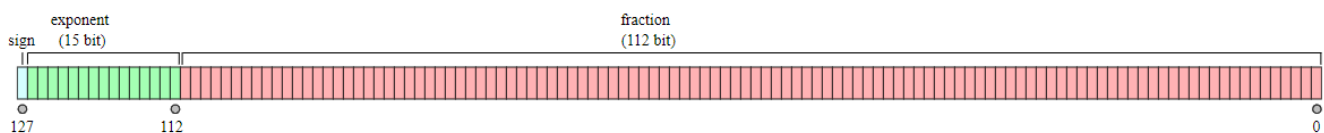
2.4. Phạm vi biểu diễn

- Do kiểu số nguyên *QInt* tôi biểu diễn theo kiểu **bù 2** nên số **lớn nhất** có thể biểu diễn là $2^{127} - 1$ số **nhỏ nhất** có thể biểu diễn là -2^{127} .

3. QFLOAT

3.1. Mô tả cấu trúc lưu trữ

- Ta sử dụng một mảng gồm bốn phần tử có kiểu dữ liệu **unsigned int** để lưu trữ 128 bit của số chấm động.
- Lý do: Một số có kiểu **unsigned int** trong C++ chiếm 16 byte (32 bit) trong bộ nhớ. Vì vậy, bốn số kiểu unsigned int sẽ cho ta lưu trữ đúng 128 bit.

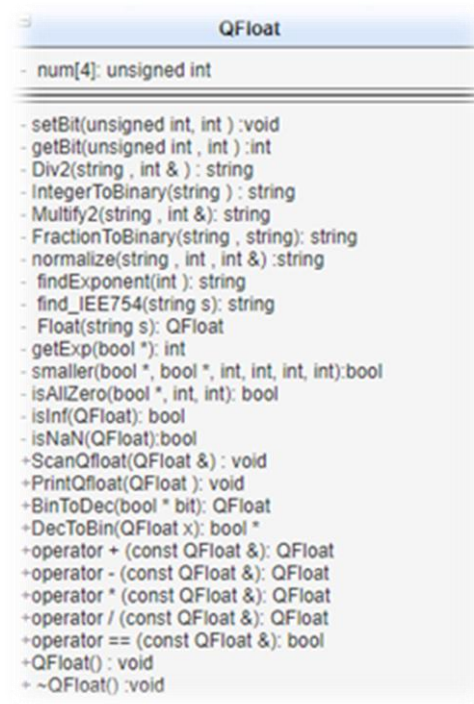


Hình 3.1 Mô tả cách lưu trữ *QFloat* trong máy tính

- Dưới đây là khai báo của lớp **QFloat** trong C++

```
class QFloat
{
private:
    unsigned int num[4];
    ...
public:
    QFloat();
    ~QFloat();
}
```

3.2. Sơ đồ lớp



Hình 3.2 Sơ đồ lớp *QFloat*

3.3. Một số hàm chính

- **PrintQFloat(QFloat &) :**

Hàm sẽ nhập vào một số được biểu diễn dưới dạng chuỗi rồi từ đó sử dụng các hàm hỗ trợ để lưu trữ theo kiểu QFloat.

- **ScanfQFloat(QFloat) :**

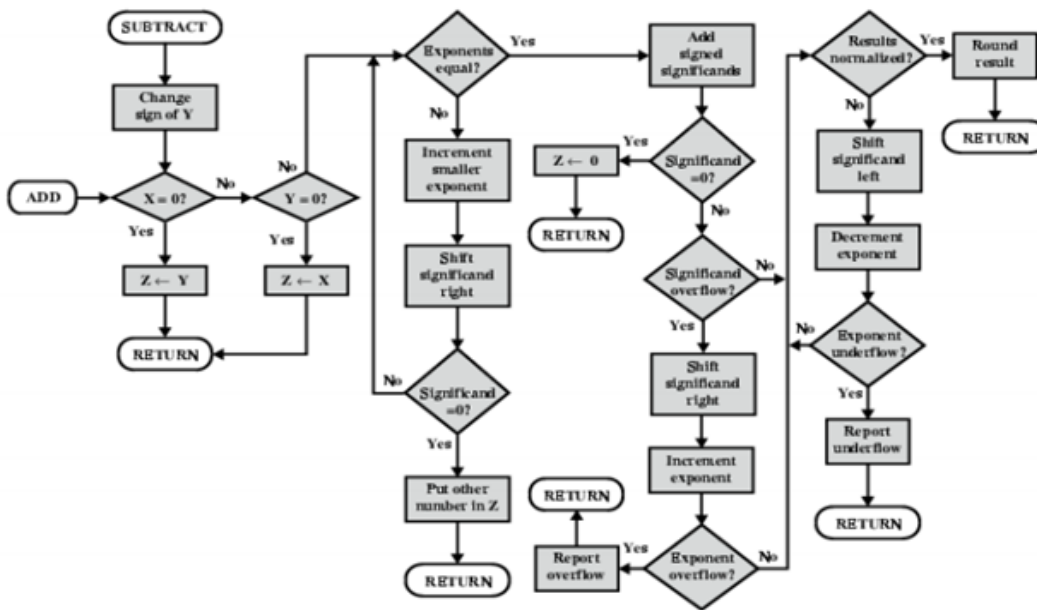
Nhập vào một số kiểu QFloat và xuất kết quả cho người xem. Lưu ý: Để đảm bảo độ chính xác nhất có thể, kết quả được hiển thị dưới dạng hệ nhị phân như sau:

Ví dụ : Kết quả là 5.5 sẽ được hiển thị là: **1.011x 2²**

- **BinToDec(bool *):** Hàm có dữ liệu đưa vào là một mảng bool chứa số chấm động của một số thập phân và sẽ trả về dạng biểu diễn QFloat của nó

- **DecToBool(QFloat):** Từ kiểu dữ liệu QFloat trả về dạng số chấm động của nó.

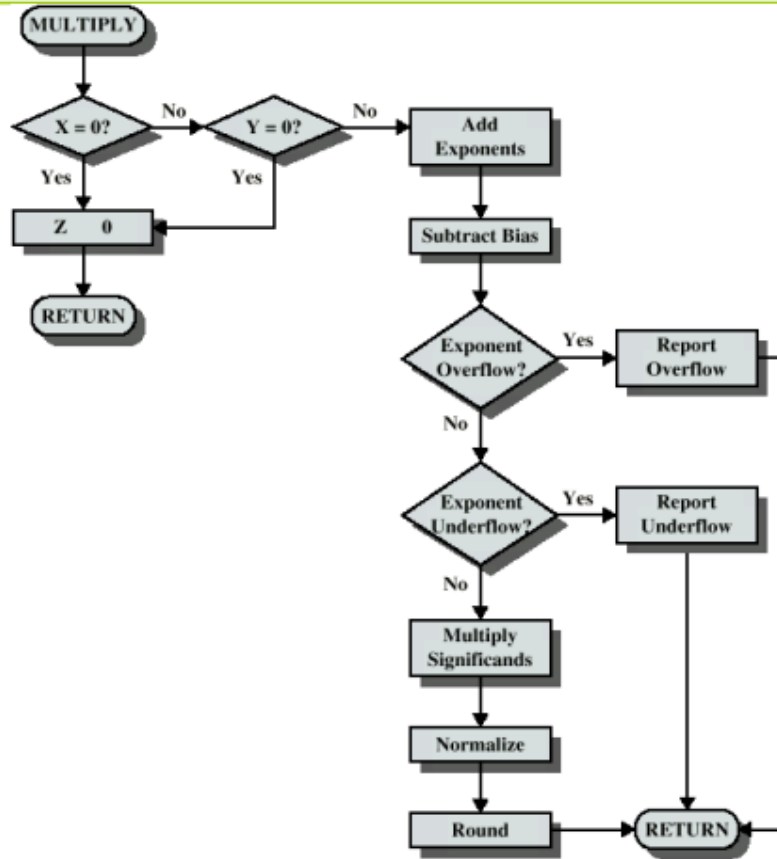
- **operator + / operator - :** Cho phép thực hiện phép tính cộng (trừ) giữa hai số chấm động. Dựa trên sơ đồ sau:



Hình 3.3 Lưu đồ thực hiện phép cộng hoặc nhân cho hai số QFloat

Theo trang : <https://www.geeksforgeeks.org/computer-arithmetic-set-2/>

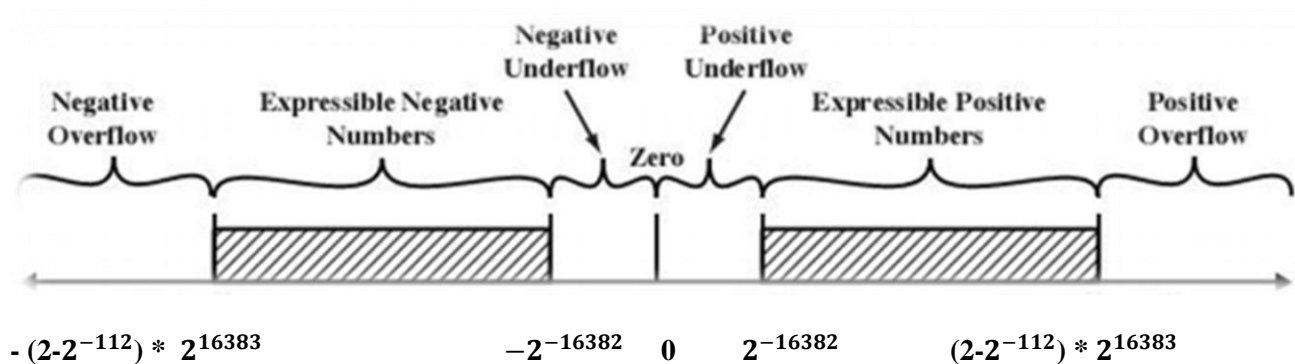
- **operator * và operator /:** Cho phép thực hiện phép tính nhân (chia) giữa hai số chấm động. Dựa trên sơ đồ sau:



Hình 3.4 Lưu đồ thực hiện phép nhân hoặc chia hai số QFloat

Theo trang: <https://www.geeksforgeeks.org/computer-arithmetic-set-2/>

3.4. Phạm vi biểu diễn



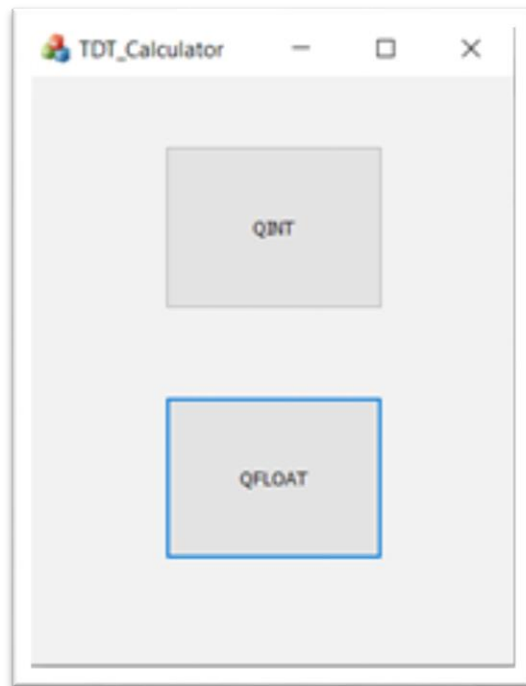
Hình 3.5 Phạm vi biểu diễn số QFloat

4. GIAO DIỆN VÀ DEMO

4.1. Giao diện chương trình

4.1.1. Màn hình khởi động

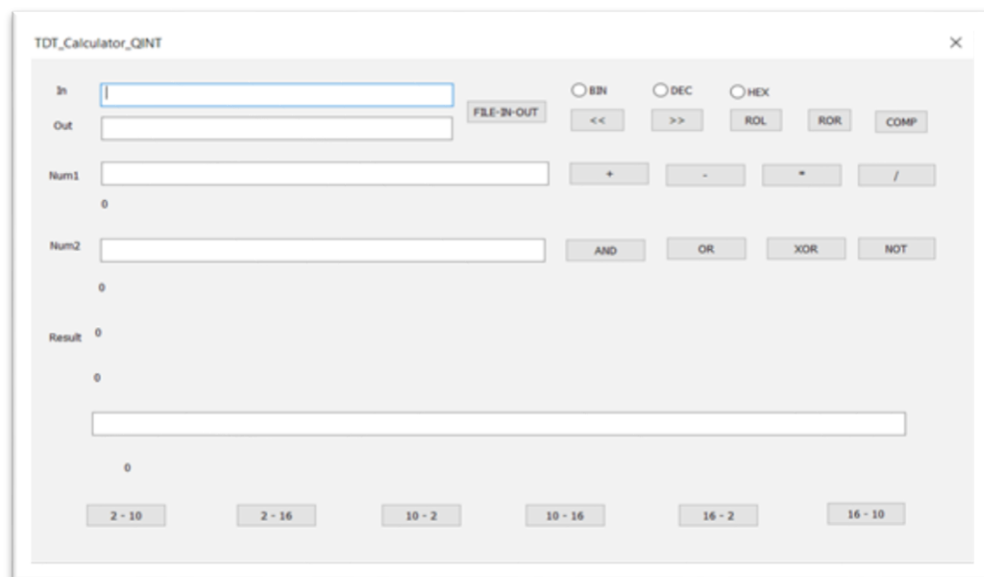
Khi chương trình chạy, một màn hình hiện lên yêu cầu người dùng chọn kiểu dữ liệu muốn thực hiện thao tác trên đó (QInt or QFloat)



Hình 4.1 Màn hình chọn chức năng

4.1.2. QInt Calculator

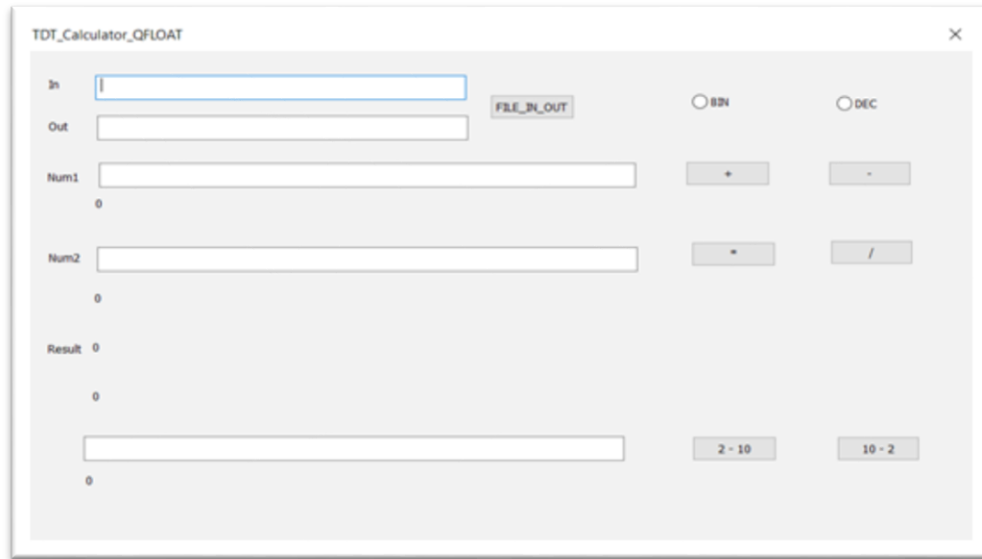
Khi đã chọn chức năng QInt, màn hình QInt Calculator sẽ hiện lên cho phép người dùng thực hiện các phép tính, chuyển đổi số trên dữ liệu nhập từ file hoặc bằng tay.



Hình 4.2 Màn hình QInt Calculator

4.1.3. QFloat Calculator

Khi đã chọn chức năng QFloat, màn hình QFloat Calculator sẽ hiện lên cho phép người dùng thực hiện các phép tính, chuyển đổi số trên dữ liệu nhập từ file hoặc bằng tay.



Hình 4.3 Màn hình QFloat Calculator

4.2. Test cases

4.2.1. QInt test

File input:



Hình 4.4 File Input của chương trình QInt

File output:



Hình 4.5 File Output của chương trình QInt

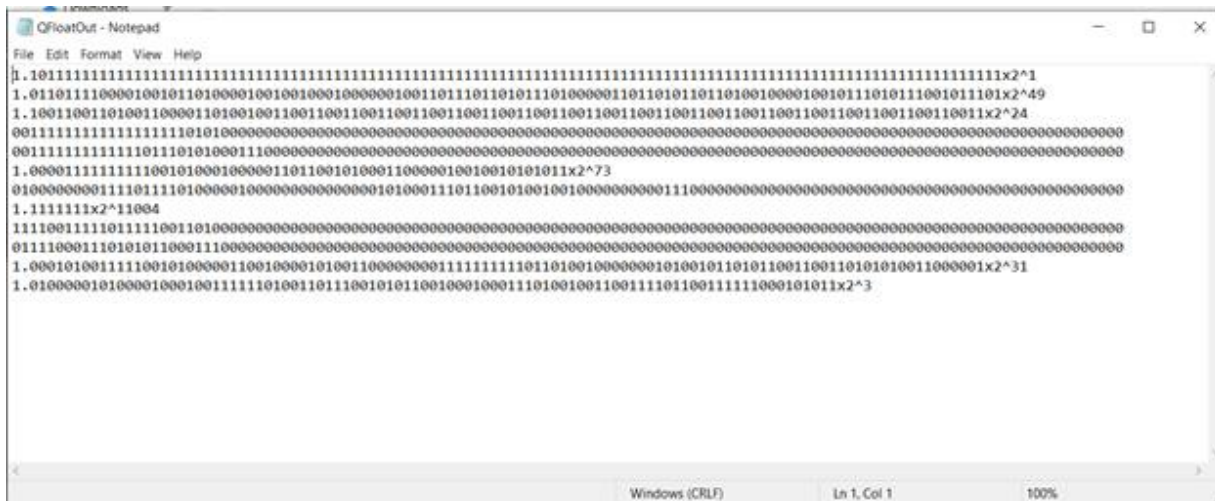
4.2.2. QFloat test

File input:



Hình 4.6 File Input của chương trình QFloat

File output:



Hình 4.4 File output của chương trình QFloat

4.3. Link Demo

<https://youtu.be/brZfdGvoayU>

5. TỔNG KẾT

Thông qua đồ án, nhóm đã biết được cách máy tính lưu trữ các số nguyên lớn và số chấm động, tính toán trên hệ nhị phân, giới hạn số học trong các cách biểu diễn. Ngoài ra còn biết cách sử dụng thư viện MFC để tạo giao diện đơn giản mà đầy đủ các tính năng phục vụ cho môn học.

6. ĐÁNH GIÁ

6.1. Đánh giá kết quả đồ án

	QInt	QFloat	Giao diện
Đã hoàn thành	<p>Biểu diễn được số QInt đúng 128 bit như đề bài.</p> <p>Thực hiện được các phép tính toán toán học và logic.</p> <p>Thực hiện chuyển đổi qua lại được giữa các hệ.</p> <p>Báo cho người dùng các trường hợp tràn dấu.</p> <p>Trong chương trình còn cho phép thực hiện phép gán với nhiều hệ khác nhau.</p> <p>Thực hiện tất cả các yêu cầu trong đề bài</p>	<p>Biểu diễn được số QFloat đúng 128 bit như đề bài.</p> <p>Thực hiện được các phép tính +, -, *, / với sai số chấp nhận được</p> <p>Thực hiện chuyển đổi qua lại được giữa các hệ 2 và 10.</p> <p>Có xử lý đối với các trường hợp đặc biệt</p> <p>Thực hiện tất cả các yêu cầu trong đề bài</p>	<p>Giao diện cho phép thực hiện tất cả các chức năng của QInt và QFloat một cách trực quan và thân thiện với người dùng.</p> <p>Báo lỗi cho người dùng khi nhập dữ liệu sai</p> <p>Cho phép nhập xuất file.</p>
Chưa hoàn thành		<p>Chưa thực hiện làm tròn số trong các phép tính</p> <p>Chưa thực hiện phép tính đối với số không chuẩn</p> <p>Chưa xử lý tràn số khi thực hiện phép tính</p>	
Mức độ hoàn thiện	100%	98%	100%

Bảng 6.1 Bảng đánh giá công việc

6.2. Đánh giá các thành viên

Họ và tên	Khối lượng công việc	Mức độ hoàn thành
Lê Nguyễn Nhật Trường	40%	100%
Võ Quốc Thắng	30%	100%
Lê Tuấn Đạt	30%	100%

Bảng 6.2 Bảng đánh giá từng thành viên

TÀI LIỆU THAM KHẢO

- [1] <https://en.wikipedia.org/wiki/128-bit>
- [2] <https://www.geeksforgeeks.org/computer-arithmetic-set-2/>
- [3] <https://www.youtube.com/watch?v=KiWz-mGFqHI&t=2s>
- [4] https://en.wikipedia.org/wiki/Quadruple-precision_floating-point_format
- [5] <https://www.youtube.com/watch?v=03fhijH6e2w&t=516s>
- [6] https://www.tutorialspoint.com/mfc/mfc_windows_fundamentals.htm