



ĐẠI HỌC QUỐC GIA TP HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA CÔNG NGHỆ THÔNG TIN



BÁO CÁO ĐỒ ÁN 1

BIỂU DIỄN VÀ TÍNH TOÁN SỐ HỌC TRÊN MÁY TÍNH

Môn: *Kiến trúc Máy tính và hợp ngữ*

Giảng viên hướng dẫn: *Ths. Lê Viết Long*

Thành viên: *Phạm Trọng Long - 18120135*
Nguyễn Thái Sơn - 18120537
Phan Đình Kha - 18120127

Tp. Hồ Chí Minh, tháng 5 năm 2020

Mục lục

1. Sơ lược về cách tổ chức dữ liệu và chương trình	3
2. Thông tin thành viên và mức độ đóng góp của từng thành viên.....	5
3. Đánh giá mức độ hoàn thành (%) ứng với từng yêu cầu và trên toàn bộ project	5
4. Cho biết phạm vi biểu diễn của các kiểu dữ liệu đã thiết kế	6
5. File output ứng với các input tương ứng (chụp hình lại)	7
6. Sơ lược về chương trình minh họa	10
Các nguồn tài liệu tham khảo.....	13

1. Sơ lược về cách tổ chức dữ liệu và chương trình

a. Số nguyên lớn

- Kiểu `QInt` có độ lớn 16 byte được thiết kế trong *class* `QInt`, bao gồm các thuộc tính và phương thức như sau:

<code>class QInt {</code>
<code>private:</code>
<code>int m_arr[4]; // dùng mảng 4 phần tử int (4*4 = 16 byte) để lưu số nguyên</code>
<code>public:</code>
<code>QInt();</code>
<code>QInt(string str, int base); // Khởi tạo bằng string</code>
<code>void ScanQInt(istream& f, int base); // Đọc dữ liệu theo base 2, 10 hoặc 16</code>
<code>void PrintQInt(ostream& f, int base); // Đọc dữ liệu theo base 2, 10 hoặc 16</code>
<code>string toDec(); // chuyển số được lưu thành số thập phân</code>
<code>string toBin(); // chuyển số được lưu thành số nhị phân</code>
<code>string toHex(); // chuyển số được lưu thành số thập lục phân</code>
<code>// Các toán tử theo yêu cầu của đề án</code>
<code>QInt operator + (const QInt& other);</code>
<code>QInt operator - (const QInt& other);</code>
<code>QInt operator * (const QInt& other);</code>
<code>pair<QInt, QInt> operator / (const QInt& other);</code>
<code>bool operator > (QInt other);</code>
<code>bool operator >= (QInt other);</code>
<code>bool operator < (QInt other);</code>
<code>bool operator <= (QInt other);</code>
<code>bool operator != (QInt other);</code>
<code>bool operator == (QInt other);</code>
<code>QInt& operator = (const QInt& other);</code>
<code>QInt& operator = (string);</code>
<code>QInt operator & (const QInt& other);</code>
<code>QInt operator (const QInt& other);</code>
<code>QInt operator ^ (const QInt& other);</code>
<code>QInt operator ~ ();</code>
<code>QInt operator >> (int x);</code>
<code>QInt operator << (int x);</code>
<code>QInt rol (int x);</code>
<code>QInt ror (int x);</code>
<code>// Và một số hàm khác để hỗ trợ cho việc thực hiện các hàm trên</code>
<code>...</code>
<code>};</code>

b. Số chấm động chính xác cao

- Số chấm động 128 bit được thiết kế trong *class QFloat*, bao gồm các thuộc tính và phương thức như sau:

```
class QFloat
{
private:
    int m_arr[4]; // dùng mảng 4 phần tử int (4*4 = 16 byte) để lưu
public:
    QFloat();
    QFloat(vector<int> vtBin);
    QFloat(string str, int base);
    ~QFloat();

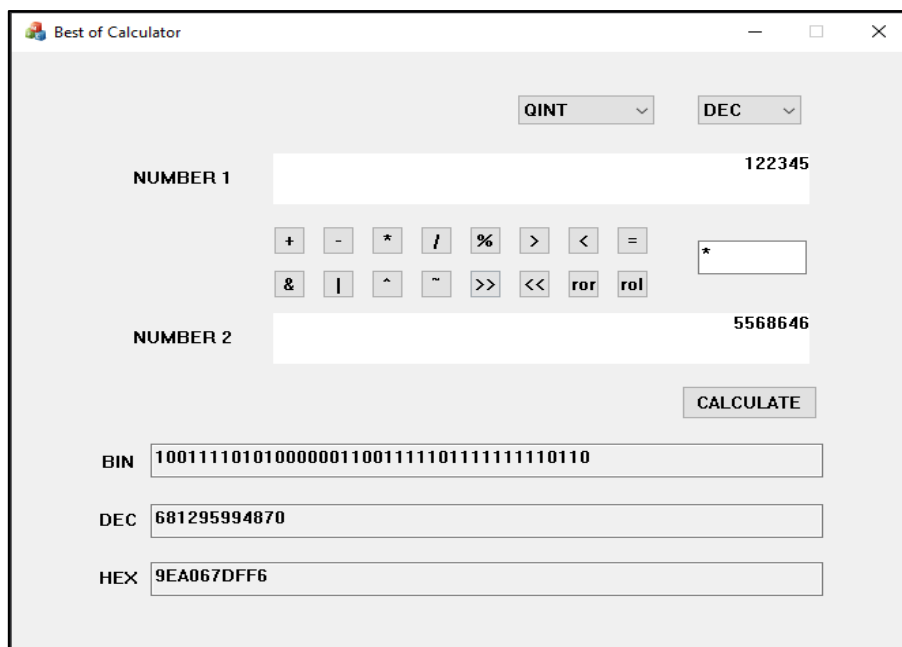
    void ScanQFloat(istream& f, int base); // đọc dữ liệu số theo base 2 hoặc 10
    void PrintQFloat(ostream& f, int base); // xuất dữ liệu số theo base 2 hoặc 10

    string toDec(); // chuyển số chấm động được lưu thành số thập phân
    string toBin(); // chuyển số chấm động được lưu thành số nhị phân

    // Các toán tử +, -, *, /
    QFloat operator + (const QFloat& other);
    QFloat operator - (const QFloat& other);
    QFloat operator * (const QFloat& other);
    QFloat operator / (const QFloat& other);
    QFloat& operator = (const QFloat& other);
    // Và một số hàm khác để hỗ trợ cho việc thực hiện các hàm trên
    ...
};
```

c. Chương trình minh họa

- Được làm dựa trên nền tảng MFC bằng ngôn ngữ C++ của Visual Studio.
- Thể hiện được các chức năng trên kiểu số nguyên lớn (16 byte) và số chấm động có độ chính xác cao (128 bit)



2. Thông tin thành viên và mức độ đóng góp của từng thành viên

a. Các công việc đã phân chia cho các thành viên

	Phan Đình Kha	Phạm Trọng Long	Nguyễn Thái Sơn
QInt	<ul style="list-style-type: none"> - Cài đặt các phương thức: toDec(), toBin(), toHec() - Cài đặt các toán tử so sánh 	<ul style="list-style-type: none"> - Cài đặt các toán tử: cộng, trừ, nhân chia. - Cài đặt các toán tử: dịch trái/phải, xoay trái/phải 	<ul style="list-style-type: none"> - Cài đặt các phương thức: Scan(), print() - Cài đặt các toán tử and, or, xor, not trên QInt
QFloat	<ul style="list-style-type: none"> - Cài đặt các phương thức: toDec(), toBin() - Cài đặt toán tử chia 	<ul style="list-style-type: none"> - Cài đặt các toán tử: cộng, trừ 	<ul style="list-style-type: none"> - Cài đặt các phương thức Scan(), print() - Cài đặt toán tử nhân
Chương trình minh họa	<ul style="list-style-type: none"> - Đề xuất các ý tưởng thiết kế 	<ul style="list-style-type: none"> - Tìm hiểu về MFC trong Visual Studio - Tổng hợp các ý tưởng, thiết kế chương trình minh họa 	<ul style="list-style-type: none"> - Đề xuất ý tưởng thiết kế
Các công việc khác	<ul style="list-style-type: none"> - Tạo test - Viết báo cáo 	<ul style="list-style-type: none"> - Fix một số bug trong chương trình minh họa - Tạo test 	<ul style="list-style-type: none"> - Tạo test - Viết hàm chạy test
Đánh giá mức độ đóng góp	33.33%	33.33%	33.33%

b. Đánh giá về thái độ làm việc và mức độ hoàn thành công việc của các thành viên:

- Các thành viên trong nhóm đều tham gia và tích cực trao đổi trong quá trình làm đồ án.
- Mỗi thành viên đều hoàn thành tốt các công việc được phân chia.
- Hỗ trợ, giúp đỡ lẫn nhau khi gặp các vấn đề khó khăn.

3. Đánh giá mức độ hoàn thành (%) ứng với từng yêu cầu và trên toàn bộ project

a. Dựa trên từng yêu cầu của đồ án

Yêu cầu		Mức độ hoàn thành	Ghi chú
QInt	Hàm nhập	100%	

	Hàm xuất	100%	
	Chuyển <i>nhị nhân</i> sang <i>thập phân</i>	100%	
	Chuyển <i>thập phân</i> sang <i>nhị phân</i>	100%	
	Chuyển <i>nhị phân</i> sang <i>thập lục phân</i>	100%	
	Chuyển <i>thập phân</i> sang <i>thập lục phân</i>	100%	
	Các toán tử: cộng, trừ, nhân, chia	100%	
	Các toán tử so sánh và gán	100%	
	Các toán tử: AND, OR, XOR, NOT	100%	
QFloat	Hàm nhập	100%	
	Hàm xuất	100%	
	Chuyển <i>nhị phân</i> sang <i>thập phân</i>	100%	
	Chuyển <i>thập phân</i> sang <i>nhị phân</i>	100%	
	Các toán tử: cộng, trừ, nhân, chia	100%	
Chương trình minh họa	Làm trên console hoặc MFC	100%	Làm trên MFC

b. Đánh giá trên toàn bộ project

- Dựa trên những đánh giá của phần a, chúng em đánh giá mức độ hoàn thành trên toàn bộ project là: **100 %**

4. Cho biết phạm vi biểu diễn của các kiểu dữ liệu đã thiết kế

a. Phạm vi biểu diễn của số nguyên lớn (16 byte)

- Số nguyên kiểu QInt được biểu diễn theo dạng bù 2, khi đó ta sẽ có bảng giá trị sau:

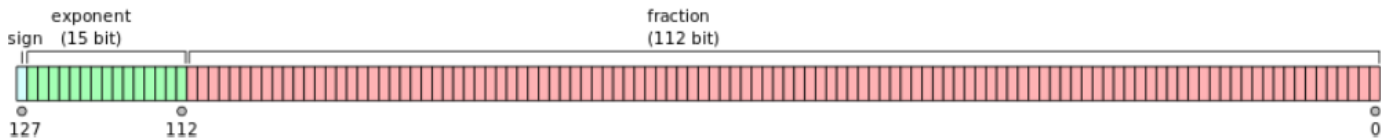
Biểu diễn dạng bù hai	Giá trị trong hệ thập phân
100000...00000	-2^{127}
100000...00001	$-2^{127} + 1$

100000...00010	$-2^{127} + 2$
...	...
11111...11111	-1
00000...00000	0
00000...00001	1
...	...
01111...11110	$2^{127} - 2$
01111...11111	$2^{127} - 1$

- Như vậy, kiểu QInt biểu diễn được số nguyên trong khoảng:
 $[-2^{127}; 2^{127} - 1] \approx [-1.701 \times 10^{38}; 1.701 \times 10^{38}]$

b. Phạm vi biểu diễn của số chấm động có độ chính xác cao (128 bit)

- Cấu trúc biểu diễn :



- Miền giá trị:

		Biểu diễn chấm động	Giá trị hệ thập phân
Số dương	Lớn nhất	1 11...110 11...111	$(2 - 2^{112}).2^{16383} \approx 1.19 \times 10^{4933}$
	Nhỏ nhất	1 00...000 00...001	$2^{-16494} \approx 6.475 \times 10^{-4966}$
Số âm	Lớn nhất	0 00...000 00...001	$-2^{-16494} \approx -6.475 \times 10^{-4966}$
	Nhỏ nhất	0 11...110 11...111	$-(2 - 2^{112}).2^{16383} \approx -1.19 \times 10^{4933}$

5. File output ứng với các input tương ứng (chụp hình lại)

Lưu ý: Vì file test (*.txt) có những số có nhiều chữ số nên không thể chụp hết tất cả, mọi người có thể vào trong thư mục test trong file gửi kèm để xem chi tiết hơn.

a. Đối với QInt

- Chuyên đổi giữ các hệ cơ số 2, 10, 16

- Các phép toán cộng, trừ, nhân, chia

[illegible]

- Các toán tử so sánh, dịch trái/phải, xoay trái/phải, các toán tử and, or, xor, not

[illegible]

b. Đối với QFloat

- Chuyển từ nhị phân sang thập phân

- Chuyển từ thập phân sang nhị phân

[illegible]

- Các toán tử cộng, trừ, nhân, chia

[illegible]

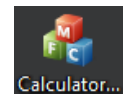
6. Sơ lược về chương trình minh họa

a. Các chức năng

- Đối với số nguyên lớn 128 bit (*QInt*):
 - Chuyển đổi giữa các hệ cơ số 2, 10, 16
 - Cộng, trừ, nhân, chia
 - Các toán tử như trong yêu cầu của đề án
 - Ngoài ra, còn có thêm phép tính chia lấy dư (%)
- Đối với số chấm động 128 bit (*QFloat*):
 - Chuyển đổi giữa nhị phân và thập phân
 - Các phép toán cộng, trừ, nhân, chia

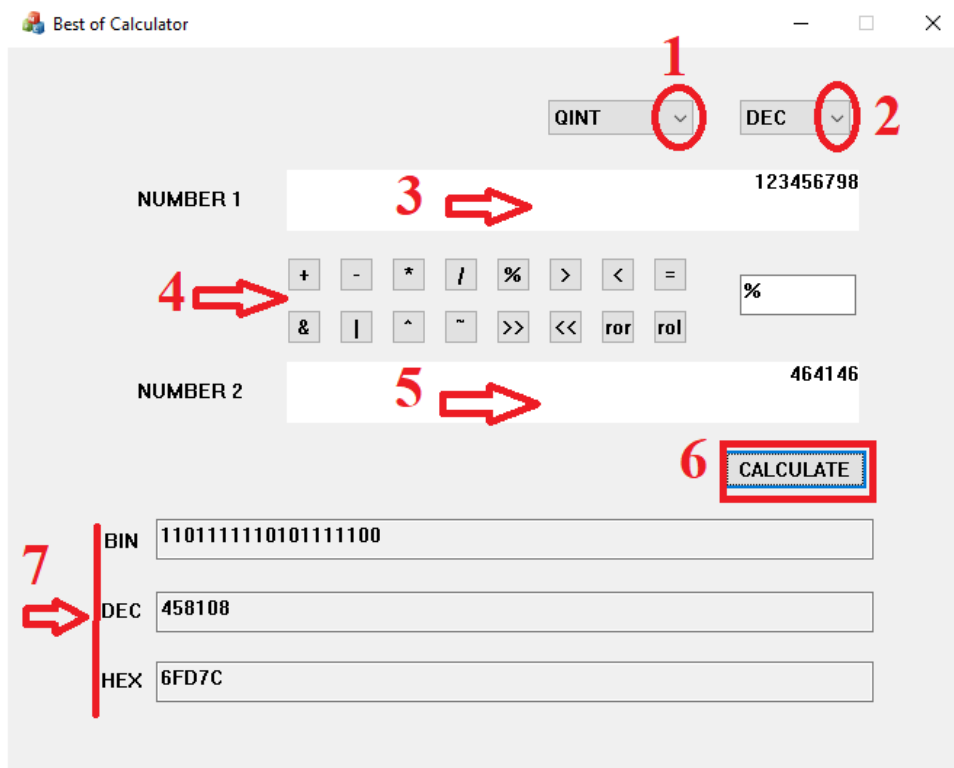
b. Cách sử dụng

- Bước 1: Nhấp đúp chuột vào file thực thi có biểu tượng.



Sau đó sẽ xuất hiện phần mềm tính toán như sau:

- Bước 2: lựa chọn các tùy chọn để sử dụng máy tính



- Trong đó: 1: chọn *QInt* hoặc *QFloat*
- 2: chọn hệ cơ số của các số nhập vào (*Bin, Dec, Hex*)
- 3: nhập số thứ nhất
- 4: chọn các tùy chọn để tính toán
- 5: nhập số thứ hai
- 6: nhấn vào button “*CALCULATE*” để thu được kết quả
- 7: hiển thị kết quả (*Bin, Dec, Hex*)

Lưu ý: Để thực hiện việc đổi 1 số sang hệ cơ số khác, thì ta chỉ cần nhập số đó vào 1 trong 2 ô *NUMBER1* hoặc *NUMBER2*, và ô còn lại nhập 0.

Các nguồn tài liệu tham khảo

1. Slide chương 2 – Biểu diễn số nguyên, Thầy Phạm Tuấn Sơn
2. Slide chương 3 – Số chấm động, Thầy Phạm Tuấn Sơn