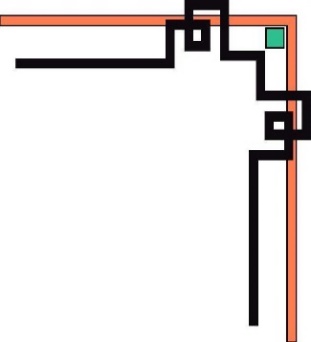
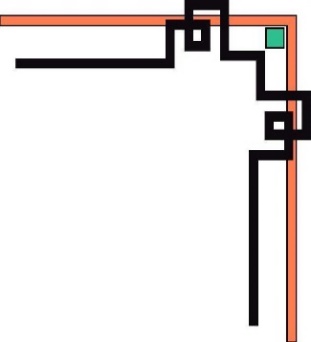
**TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN**



**KHOA CÔNG NGHỆ THÔNG TIN**

🙢🙢🙢 🕮 🙠🙠🙠

Đồ án 1

Kiến trúc máy tính và hợp ngữ

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

**Nhóm thực hiện:**

**Học kỳ II   
Năm học: 2017- 2018**

**LỜI CẢM ƠN**

Trong quá trình thực hiện đồ án đề tài, nhóm em đã nhận được rất nhiều sự giúp đỡ, hỗ trợ từ các thầy cô Trường Đại học Khoa học Tự nhiên – ĐHQG TP.HCM và các bạn bè trong trường. Nhóm em xin bày tỏ lòng cảm ơn chân thành về sự hướng dẫn, chỉ bảo của mọi người.

Đặc biệt, nhóm em xin bày tỏ lòng biết ơn sâu sắc đến các thầy cô khoa Công nghệ thông tin, cụ thể hơn chúng em xin cảm ơn thầy Phan Tuấn Sơn đã hỗ trợ, giảng dạy rất kĩ lưỡng từng phần nhỏ để chúng em có một đồ án thật hoàn chỉnh và phù hợp nhất. Chính những thứ tưởng chừng như nhỏ nhặt này đã góp phần to lớn giúp chúng em hoàn thành đồ án và bảng báo cáo này thật hoàn chỉnh nhất. Một lần nữa, chúng em xin bày tỏ lòng biết ơn sâu sắc đến với các thầy cô và bạn bè.

Ngoài ra, nhóm chúng em còn nhận được rất nhiều sự khích lệ tinh thần, động viên cổ vũ không chỉ từ các bạn đồng trang lứa mà còn đến từ các anh chị khoa Công nghệ thông tin khóa trước và các thầy cô.

Chúng em xin chân thành cảm ơn!

Mục lục

[1 CHƯƠNG I: MỞ ĐẦU 3](#_Toc510122503)

[1.1 Giới thiệu nhóm và phân công công việc 3](#_Toc510122504)

[1.2 Mô tả đồ án 3](#_Toc510122505)

[2 CHƯƠNG II: GIỚI THIỆU ĐỒ ÁN 4](#_Toc510122506)

[2.1 Thời gian thực hiện 4](#_Toc510122507)

[2.2 Các bước thực hiện đồ án 4](#_Toc510122508)

[3 CHƯƠNG III: NỘI DUNG ĐỒ ÁN 5](#_Toc510122509)

[3.1 Sơ đồ UML 5](#_Toc510122510)

[3.2 Lớp BigNum 5](#_Toc510122511)

[3.2.1 Hàm get\_bit(int, bool) 5](#_Toc510122512)

[3.2.2 Hàm get\_bit(int) 5](#_Toc510122513)

[3.3 Lớp BigInt 5](#_Toc510122514)

[3.3.1 Toán tử cộng 5](#_Toc510122515)

[3.3.2 Toán tử trừ 6](#_Toc510122516)

[3.3.3 Toán tử nhân 6](#_Toc510122517)

[3.3.4 Toán tử chia 6](#_Toc510122518)

[3.4 Lớp BigFloat 6](#_Toc510122519)

[3.4.1 Toán tử cộng 6](#_Toc510122520)

[3.4.2 Toán tử trừ 6](#_Toc510122521)

[3.4.3 Toán tử nhân 6](#_Toc510122522)

[3.4.4 Toán tử chia 6](#_Toc510122523)

[3.5 Lớp BigDec 6](#_Toc510122524)

[4 CHƯƠNG V: ĐÁNH GIÁ VÀ TỔNG KẾT QUÁ TRÌNH 7](#_Toc510122525)

[4.1 Đánh giá hoàn thành đồ án 7](#_Toc510122526)

[4.2 Giao diện chương trình với testcase 8](#_Toc510122527)

[5 TÀI LIỆU THAM KHẢO 8](#_Toc510122528)

# CHƯƠNG I: MỞ ĐẦU

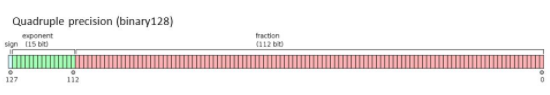
## Giới thiệu nhóm và phân công công việc

* Tên nhóm:
* Số thành viên: 3 người
* Trưởng nhóm:

|  |  |  |  |
| --- | --- | --- | --- |
| STT | MSSV | Họ và tên | Công việc |
| 1 |  |  |  |
| 2 |  |  |  |
| 3 | 1612899 | Hoàng Xuân Trường |  |

## Mô tả đồ án

* Biểu diễn và tính toán số học trên máy tính.
  + Thiết kế kiểu dữ liệu biểu diễn có số nguyên lớn có dấu 16 bytes(128bit)
  + Thiết kế kiểu dữ liệu biểu diễn số chấm động có độ chính xác Quadruple-precision (độ chính xác gấp 4 lần) độ lớn 128bit có cấu trúc biểu diễn như sau:



* Viết chương trình calculator đơn giản gồm các chức năng sau:
  + Chuyển đổi giá trị qua lại giữa các hệ 2,10,16 (số chấm động không chuyển đổi qua hệ 16)
  + Tính toán cộng, trừ, nhân, chia giữa 2 số bất kỳ.

# CHƯƠNG II: GIỚI THIỆU ĐỒ ÁN

## Thời gian thực hiện

* Đồ án bắt đầu từ ngày 10/3 đến ngày 31/3

## Các bước thực hiện đồ án

# CHƯƠNG III: NỘI DUNG ĐỒ ÁN

## Sơ đồ UML

* Fasd

## Lớp BigNum

* Diễn tả là gì
* Các tổ chức bignum
* Phạm vi biểu diễn số trong lớp BigNum:
* Một số hàm quan trong trong lớp BigNum: get\_bit(), set\_bit()

### Hàm get\_bit(int, bool)

* Hàm để bật hoặc tắt bit tại một vị trí cụ thể trong data[2] được khai báo trong lớp BigNum
* Code:

1. void BigNum::set\_bit(int index, bool value) {
2. if (value == 1)
3. data[index >> 6] |= 1 ll << (index & 63);
4. } else {
5. data[index >> 6] &= ~(1 ll << (index & 63));
6. }
7. }

### Hàm get\_bit(int)

* Hàm get\_bit(int) trả về kiểu bool để lấy một bit tại một vị trí
* Code:

1. bool BigNum::get\_bit(int index) const {
2. return (data[index >> 6] >> (index & 63)) & 1;
3. }

## Lớp BigInt

* Lớp BigInt dùng để lưu giá trị của một số nguyên lớn có dấu có 128 bit.
* Lớp BigInt được kế thừa từ lớp cha BigNum với mục đích sử dụng lại các hàm ví dụ như get\_bit(), set\_bit() để dễ dàng hơn trong các thao tác xử lý.
* Phạm vi biểu diễn số trong lớp BigInt:

### Hàm chuyển từ hệ thập phân sang nhị phân

### Hàm chuyển từ hệ nhị phân sang hệ thập lục phân

### Toán tử cộng

* Dựa trên [Full adder](https://en.wikipedia.org/wiki/Adder_%28electronics%29#Full_adder), ta chạy từng bit từ trái sang phải trên data[2]. Trong mỗi lần lặp, ta cần tính tổng của 3 bit: 2bit của hai số BigInt cho trước và 1bit của *Carry*. Biến *Carry* được tính bằng cách OR của tất cả các cặp. Thuật toán cụ thể:

1. Sum = a XOR b XOR carry; //a và b là hai bit đang xét của hai số BigInt
2. carry = (a & b) | (a & c) | (b & c);

### Toán tử trừ

* Duyệt từng bit của số BigInt, ta thực hiện phép tính trừ trên hai bit của hai số BigInt và số mượn *borrow.* Mô tả cách thực hiện:

1. Kết quả = (a - b - borrow) and 1;
2. If(kết quả < 0) thì borrow = 1 ngược lại borrow = 0;

### Toán tử nhân

### Toán tử chia

## Lớp BigFloat

* Lớp BigFloat là kiểu dữ liệu biểu diễn số chấm động có độ chính xác Quadruple-precision độ lớn 128bit
* Lớp BigFloat được kế thừa từ lớp BigNum
* Phạm vi biểu diễn của lớp BigFloat:

### Hàm chuyển từ hệ thập phân sang nhị phân

### Hàm chuyển từ hệ nhị phân sang hệ thập lục phân

### Toán tử cộng

### Toán tử trừ

### Toán tử nhân

### Toán tử chia

## Lớp BigDec

* Mô tả:
* Mục đích:

# CHƯƠNG V: ĐÁNH GIÁ VÀ TỔNG KẾT QUÁ TRÌNH

## Đánh giá hoàn thành đồ án

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Tên lớp | Yêu cầu | | | Tên hàm | Mức độ hoàn thành |
| BigInt | Hàm nhập | | | friend istream& operator>>(istream&, BigInt&); |  |
| Hàm xuất | | | friend ostream& operator<<(ostream&, const BigInt&); |  |
| Chuyển đổi từ hệ thập phân sang hệ nhị phân | | |  |  |
| Chuyển đổi từ hệ nhị phân sang hệ thập phân | | |  |  |
| Chuyển đổi từ nhị phân sang hệ thập lục phân | | |  |  |
| Chuyển đổi từ thập phân sang thập lục phân | | |  |  |
| Các phép tính | Cộng | | BigInt operator+(const BigInt&) const; |  |
| Trừ | | BigInt operator-(const BigInt&) const; |  |
| Nhân | | BigInt operator\*(const BigInt&) const; |  |
| Chia | Lấy phần nguyên | BigInt operator/(const BigInt&) const; |  |
| Lấy phần dư | BigInt operator%(const BigInt&) const; |  |
| BigFloat | Hàm nhập | | |  |  |
| Hàm xuất | | |  |  |
| Chuyển đổi từ hệ thập phân sang hệ nhị phân | | |  |  |
| Chuyển đổi từ hệ nhị phân sang hệ thập phân | | |  |  |
| Các phép tính | Cộng | | BigFloat operator+(const BigFloat&) const; |  |
| Trừ | | BigFloat operator-(const BigFloat&) const; |  |
| Nhân | | BigFloat operator\*(const BigFloat&) const; |  |
| Chia | | BigFloat operator/(const BigFloat&) const; |  |

## Giao diện chương trình với testcase

# TÀI LIỆU THAM KHẢO

[https://en.wikipedia.org/wiki/Adder\_%28electronics%29#Full\_adder](https://en.wikipedia.org/wiki/Adder_%28electronics%29%23Full_adder)