

**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH**  
**TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN**  
**KHOA CÔNG NGHỆ THÔNG TIN**



**BÀI BÁO CÁO BÀI TẬP LAB1**

**Giảng viên hướng dẫn: Bùi Văn Thạch**

**Trợ giảng hướng dẫn: Phan Quốc Kỳ**

**Sinh viên thực hiện: Phạm Văn Hữu Tài**

**MSSV: 24120435 - Lớp: 24CTT3**

**TP.Hồ Chí Minh, ngày 07 tháng 05 năm 2025**

# Mục Lục

1. Yêu cầu \_\_\_\_\_
2. Đầu vào \_\_\_\_\_
3. Đầu ra \_\_\_\_\_
4. Ý tưởng, phương pháp \_\_\_\_\_
  - a. Ý tưởng \_\_\_\_\_
  - b. File 24120435.cpp \_\_\_\_\_
  - c. File cong.h \_\_\_\_\_
  - d. File tru.h \_\_\_\_\_
  - e. File nhan.h \_\_\_\_\_
  - f. File chia.h \_\_\_\_\_
  - g. File SoSanhNhoHon.h \_\_\_\_\_
  - h. File DaoNguocMang.h \_\_\_\_\_
5. Liên kết giữa các tệp trong chương trình \_\_\_\_\_
6. Tóm tắt ý tưởng \_\_\_\_\_
7. Hỗ trợ \_\_\_\_\_

## 1. Yêu cầu

- Viết chương trình C++ tính toán các biểu thức số học dạng infix chứa các số nguyên lớn hỗ trợ nhiều toán hạng, các phép toán cơ bản(+, -, \*, /) và dấu ngoặc để ưu tiên phép toán.
- Chương trình phải nhận hai đối số dòng lệnh: argv, argc.

## 2. Đầu vào

- Một biểu thức dạng infix.
- Biểu thức có thể chứa số nguyên lớn lên đến 100 chữ số.
- Các toán tử: + - \* /.
- Dấu ngoặc () để ưu tiên phép toán.
- Đầu vào được đọc trong tệp tests.txt.
- Mỗi dòng trong tệp tests.txt là một biểu thức.

## 3. Đầu ra

- Kết quả của mỗi biểu thức được ghi vào trong tệp có tên "output\_MSSV.txt".
- Kết quả cũng được in ra màn hình console hoặc terminal.
- Nếu có lỗi trong quá trình tính cần phải ghi lỗi vào tệp ra và in ra màn hình.

## 4. Ý tưởng, phương pháp:

### a. Ý tưởng:

- + Vì các số nguyên rất lớn nên mình đã chọn biểu diễn mỗi số dưới dạng một vector<int> chứa các chữ số riêng lẻ.
- + Chuyển biểu thức dạng infix sang postfix.
- + Tính toán đánh giá biểu thức postfix sử dụng các hàm tự code.
- + Xử lý các trường hợp lỗi.

### b. File 24120435.cpp

- Cấu trúc BigInt:
  - + Lưu trữ số lớn bằng vector<int> digits và bool isNegative (dấu).
  - + Ví dụ số 123 được lưu trong cấu trúc là BigInt({3, 2, 1}, false).
- Chuyển chuỗi thành BigInt (StringtoBigInt):
  - + Đọc chuỗi (ví dụ: -123), xác định dấu isNegative = true, chuyển từng ký tự thành chữ và đảo ngược (3, 2, 1).

- Chuyển BigInt thành chuỗi (bigintToString):
  - + Chuyển digits thành chuỗi đảo ngược thêm dấu âm nếu có.
  - + Ví dụ `BigInt({3, 2, 1}, true)` -> -123.
- Chuyển infix sang postfix (infixToPostfix):
  - + Dùng một stack để lưu toán tử và dấu ngoặc.
  - + Duyệt biểu thức: Nếu là số hoặc số âm thêm vào kết quả, nếu là dấu ngoặc mở đẩy vào stack nếu là dấu ngoặc đóng lấy các toán tử từ stack cho đến khi gặp dấu ngoặc đóng, nếu là toán tử lấy các toán tử có độ ưu tiên cao hơn hoặc bằng toán tử từ stack sau đó đẩy các toán tử hiện tại vào stack.
  - + Ví dụ: `1 + 2 * 3` -> `1 2 3 * +`.
  - + Lấy hết các toán tử từ stack để hoàn thành biểu thức postfix.
- Đánh giá biểu thức postfix (evaluatePostfix):
  - + Dùng stack để lưu các số lớn.
  - + Duyệt biểu thức postfix: Nếu là số chuyển thành bigint và đẩy vào stack, nếu là toán tử lấy hai số từ stack thực hiện phép toán và đẩy vào stack.
  - + Kết quả còn lại là chữ số cuối cùng trong stack.
- Hàm main:
  - + Nhận đối số dòng lệnh.
  - + Đọc từng dòng từ tệp đầu vào, xử lý biểu thức và ghi vào tệp đầu ra.
  - + Xử lý lỗi bằng try catch.

**c. File *tong.h*:**

- Kiểm tra hợp lệ: Đảm bảo các chữ số nằm trong 0->9.
- Xử lý trường hợp đặc biệt:
  - + Nếu a b rỗng trả về 0.
  - + Nếu a rỗng trả về b với dấu của b.
  - + Nếu b rỗng trả về a với dấu của a.
- Xử lý theo dấu:
  - + Cả a b không âm thực hiện phép cộng từng chữ số xử lý nhớ carry.

- + Không âm và âm chuyển thành  $a - b$  gọi HieuHaiSo
- + Âm và không âm chuyển thành  $b - a$  gọi HieuHaiSo.
- Duyệt từng chữ số trong mảng cộng từng cặp chữ số với carry lưu kết quả vào vector.

**d. File hieu.h:**

- Kiểm tra tính hợp lệ: Đảm bảo các chữ số nằm trong 0->9.
- Xử lý trường hợp đặc biệt:
  - + Nếu a b rỗng trả về 0.
  - + Nếu a rỗng trả về b với dấu của b.
  - + Nếu b rỗng trả về a với dấu của a.
- Xử lý theo dấu:
  - + Âm và không âm chuyển thành  $-(a + b)$  gọi TongHaiSo.
  - + Không âm và âm chuyển thành  $a + b$  gọi TongHaiSo.
  - + Cả hai không âm hoặc cả hai âm: So sánh a và b để xác định dấu nếu  $a < b$  hoán đổi a và b đặt dấu âm, thực hiện phép trừ từng chữ số xử lý mượn borrow.

**e. File nhan.h:**

- Kiểm tra hợp lệ: đảm bảo các chữ số nằm trong 0->9.
- Xác định dấu: kết quả âm nếu a b có dấu khác nhau.
- Xử lý trường hợp đặc biệt: nếu a hoặc b rỗng hoặc bằng 0 kết quả là 0.
- Phép nhân:
  - + Tạo vector lưu kết quả c kích thước tối đa tổng hai kích thước của a và b
  - + Duyệt qua từng chữ số của a và b nhân từng cặp chữ số cộng nhớ carry là lưu kết quả vào vị trí thích hợp trong c.
  - + Loại bỏ số 0 thừa ở đầu.

**f. File chia.h:**

- Hàm trừ hai số: thực hiện phép trừ  $a - b$   $a \geq b$  xử lý mượn borrow dung trong chia để cập nhật số dư.
- Hàm nhân với số: nhân một số lớn với một chữ số từ 0->9 xử lý nhớ carry dung trong chia để thử các thương.

- Hàm ChiaHaiSo: Kiểm tra hợp lệ đảm bảo nằm trong 0->9. Không chia cho 0. Xử lý các trường hợp đặc biệt: Nếu a rỗng trả về 0, nếu b rỗng thì báo lỗi, nếu  $a < b$  trả về 0, nếu  $a = b$  trả về 1. Xác định dấu của kết quả âm nếu dấu a và b khác nhau. Chia như chia tay sử dụng tìm kiếm nhị phân để tìm thương q thích hợp trừ số hiện tại với  $num2 * q$  cập nhật số dư và thêm q vào thương lặp cho tới khi không chia được nữa, loại bỏ số 0 thừa ở đầu.

**g. File SoSanhNhoHon.h**

- Loại bỏ số 0 ở đầu để đo độ dài mảng.
- So sánh độ dài hai mảng nếu khác nhau số nhỏ hơn có độ dài nhỏ hơn.
- Nếu độ dài bằng nhau so sánh từng chữ số từ cuối về đầu
- Trả về true nếu  $a < b$ , trả về false nếu  $b > a$ .

**h. File DaoNguocMang.h**

- Đảo ngược vector sử dụng hai con trỏ l bắt đầu từ đầu là r bắt đầu từ cuối.
- Hoán đổi các phần tử tại r là l và cập nhật lại l và r. Cho tới khi  $l \geq r$ .

**5. Liên kết giữa các tệp trong chương trình:**

- + 24120435.cpp: Là hàm chính thực hiện chạy chương trình xử dụng các tệp các để xử lý.
- + tong.h, hieu.h, nhan.h, chia.h: Cung cấp các hàm tính toán số lớn.
- + SoSanhNhoHon.h: Hỗ trợ so sánh các số lớn dùng trong hàm hieuHaiSo và ChiaHaiSo.
- + DaoNguocMang.h: Dùng để đảo ngược mảng.

**6. Tóm tắt ý tưởng:**

- Biểu diễn số lớn: Sử dụng vector<int> để lưu chữ số và bool để lưu dấu.
- Xử lý biểu thức: Chuyển từ infix sang postfix sau đó đánh giá postfix bằng stack.

- Tính toán số lớn: cộng và trừ xử lý từng chữ số quản lý mượn nhớ. Nhân nhân từng cặp chữ số, cộng nhớ. Chia chia từng nhóm chữ số tìm thương bằng tìm kiếm nhị phân trừ số dư.

## **7. Hỗ trợ:**

- AI: Gork nói ra các trường hợp đặc biệt để xử lý, nói ra các bug để xử lý. Hỗ trợ ý tưởng viết chương trình.
- Sách: tham khảo sách Kỹ thuật lập trình. Cách tạo dựng xử lý số lớn.
- Đồ án: học kỳ 1 xử lý số lớn.
- Em nghỉ dùng các hỗ trợ khoảng 10-20%, còn lại là suy nghĩ và tự làm.

