BỘ GIÁO DỤC VÀ ĐÀO TẠO

**TRƯỜNG ĐẠI HỌC NHA TRANG**

** KHOA CÔNG NGHỆ THÔNG TIN**

**THỰC TẬP CƠ SỞ CÔNG NGHỆ THÔNG TIN**

**THỰC HIỆN CÁC PHÉP TÍNH CƠ BẢN (CỘNG, TRỪ, NHÂN, CHIA) TRÊN CÁC SỐ NGUYÊN LỚN(CÓ TRÊN 20 CHỮ SỐ)**

**Giảng viên hướng dẫn: Nguyễn Đình Hoàng Sơn**

**Sinh viên thực hiện: Cao Xuân Ngọc**

**Mã số sinh viên: 61134037**

**Khánh Hòa – 2021**

Mục Lục

[PHẦN MỞ ĐẦU 3](#_Toc91693640)

[1. Lý do chọn đề tài: 3](#_Toc91693641)

[2. Mục tiêu của đề tài. 3](#_Toc91693642)

[3. Phạm vi nghiên cứu. 3](#_Toc91693643)

[4. Phương pháp nghiên cứu. 3](#_Toc91693644)

[CHƯƠNG 1: CƠ SỞ LÝ THUYẾT 4](#_Toc91693645)

[1.1 Phép cộng 4](#_Toc91693646)

[1.1.1 Thuật toán 4](#_Toc91693647)

[1.1.2 Các bước thực hiện 4](#_Toc91693648)

[1.2 Phép trừ 4](#_Toc91693649)

[1.2.1. Thuật toán 4](#_Toc91693650)

[1.2.2. Các bước thực hiện 4](#_Toc91693651)

[1.3. Phép nhân 5](#_Toc91693652)

[1.3.1. Thuật toán 5](#_Toc91693653)

[1.3.2. Các bước thực hiện 5](#_Toc91693654)

[1.4. Phép chia 5](#_Toc91693655)

[1.4.1. Thuật toán 5](#_Toc91693656)

[1.4.2. Các bước thực hiện 5](#_Toc91693657)

[CHƯƠNG 2 : BÀI TOÁN THỰC HIỆN CÁC PHÉP TÍNH CƠ BẢN TRÊN SỐ NGUYÊN LỚN 6](#_Toc91693658)

[2.1. Đặc tả bài toán 6](#_Toc91693659)

[2.2. Yêu cầu hệ thống 6](#_Toc91693660)

[2.3. Phân tích thiết kế chương trình 6](#_Toc91693661)

[2.3.1. Cấu trúc chương trình 6](#_Toc91693662)

[2.3.2. Các chức năng của chương trình 14](#_Toc91693663)

[2.4. Cài đặt chương trình 14](#_Toc91693664)

[KẾT LUẬN 17](#_Toc91693665)

[1. Các kết quả đạt được 17](#_Toc91693666)

[2. Ưu và nhược điểm của đề tài 17](#_Toc91693667)

[3. Hướng phát triển 17](#_Toc91693668)

# PHẦN MỞ ĐẦU

1. **Lý do chọn đề tài:**

**Công nghệ thông tin** là một trong những ngành phát triển vượt bậc trong những năm gần đây. Ngày nay với sự phát triển nhanh chóng của xã hội thì công nghệ thông tin được ứng dụng rộng rãi ở hầu hết tất cả các lĩnh vực và ngày càng đóng vai trò quan trọng, trở thành một phần thiết yếu trong đời sống hằng ngày. Công nghệ thông tin là một ngành đòi hỏi ở người học một nền tảng kiến thức vững chắc, sự tư duy logic cao, hiểu biết sâu rộng trên nhiều lĩnh vực. Với chúng em hiện đang là những sinh viên công nghệ thông tin cần phải có sự đầu tư, không ngừng học hỏi để nâng cao kiến thức. Do đó để củng cố lại kiến thức đã học, đề tài mà em chọn để thực hiện là: **Thực hiện các phép tính cơ bản (cộng, trừ, nhân, chia) trên các số nguyên lớn (có trên 20 chữ số)**

1. **Mục tiêu của đề tài.**
   * Củng cố lại kiến thức đã học về các cấu trúc điều khiển, ôn tập lại kiến thức về phần mảng, chuỗi cũng như kiểu dữ liệu cấu trúc...
   * Rèn luyện kỹ năng lập trình trên ngôn ngữ C++.
   * Ứng dụng lý thuyết đã học giải quyết bài toán ứng dụng cụ thể
2. **Phạm vi nghiên cứu.**
   * Lý thuyết về cộng, trừ, nhân, chia trên số nguyên
   * Bài toán thực hiện các phép tính cơ bản trên các số nguyên lớn
3. **Phương pháp nghiên cứu.**
   * Nghiên cứu giáo trình Kỹ thuật lập trình và Cấu trúc dữ liệu đã được học.
   * Tìm kiếm và nghiên cứu trên mạng Internet.

# CHƯƠNG 1: CƠ SỞ LÝ THUYẾT

## 1.1 Phép cộng

### 1.1.1 Thuật toán

Thông thường, khi cộng hai số có nhiều chữ số, chúng ta thực hiện từ phải qua trái. Với mỗi lần cộng, kết quả của nó còn được cộng với giá trị nhớ của lần trước đó và đồng thời cũng xác định giá trị nhớ mới. Ở đây, vì các số được lưu theo chiều ngược lại nên chúng ta thực hiện từ trái qua phải, tức là từ đầu đến cuối xâu.

### 1.1.2 Các bước thực hiện

- Bước 1: Loại bỏ các chữ số 0 vô nghĩa ở 2 mảng a và b

- Bước 2: Thêm chữ số 0 ở đầu mảng có độ dài ngắn hơn để 2 mảng có cùng độ dài (ví dụ: a = 5678, b = 34 thì ta thấy mảng a có 4 phần tử, mảng b có 2 phần tử; vậy để 2 mảng này có số phần tử bằng nhau thì chúng ta thêm 2 chữ số 0 nữa vào mảng b; lúc này b = 0034)

Dùng 1 biến nhớ m (kiểu char, được khởi tạo bằng 0) để lưu trữ số nhớ sau mỗi bước tính

- Bước 3: Đảo ngược chuỗi và tính toán

\* Tổng = số 1 + số 2 + nhớ (ban đầu bằng 0)

\* Nếu tổng > 9 thì gán tổng = tổng -10 và cho nhớ bằng 1. Ngược lại nếu tổng bé hơn hoặc bằng 9 thì nhớ vẫn bằng 0

- Bước 4: Xuất kết quả ra , sau đó đảo ngược chuỗi kết quả

## 1.2 Phép trừ

### 

### 1.2.1. Thuật toán

Giống như cộng, chúng ta cũng thực hiện từ phải qua trái. Với mỗi lần trừ, đề phòng chữ số của số bị trừ nhỏ hơn nên ta cứ vay tạm một chục. Nếu lần trước đã vay rồi thì lần này phải trừ đi. Nhưng chỉ vay nếu kết quả sau khi trừ là bé hơn 10, còn lớn 10 là không phải vay

### 1.2.2. Các bước thực hiện

* Bước 1: Loại bỏ các chữ số 0 vô nghĩa ở 2 mảng a và b
* Bước 2: Thêm chữ số 0 ở đầu mảng có độ dài ngắn hơn để 2 mảng có cùng độ dài
* Bước 3: Đảo ngược chuỗi và tính toán
  + Hiệu = số 1 – số 2 – mượn (ban đầu mượn bằng 0)
  + Nếu hiệu < 0 thì { hiệu = hiệu +10; mượn 1}
  + Nếu hiệu > 0 thì mượn bằng 0
* Bước 4: Xuất kết quả ra, sau đó đảo ngược chuỗi kết quả

## 1.3. Phép nhân

### 1.3.1. Thuật toán

Thông thường, khi nhân a với b, chúng ta thực hiện từ phải qua trái. Mỗi lần sẽ nhân 1 chữ số của b với số a và đặt kết quả dịch sang trái 1 chữ số. Nhưng trong mỗi lần đó chúng ta lại lần lượt nhân từng chữ số của a với chữ số nói trên của b. Cũng như phép cộng, kết quả phụ thuộc vào việc nhớ của lần nhân trước và xác định giá trị nhớ mới. Việc nhận này được thực hiện từ trái qua phải

### 1.3.2. Các bước thực hiện

* Bước 1: Duyệt từ cuối xâu b về đầu xâu
* Bước 2: Tách từng phần tử của xâu b nhân với xâu a
  + Tích = số nhỏ \* k + nhớ (k là số được tách từ xâu a)
  + Nhớ = tich / 10
  + Tích = tích % 10

( lưu ý cộng thêm nhớ nếu nhớ khác 0 )

* Bước 3: Cộng liên tiếp các kết quả thu được
* Bước 4: Xử lí kí tự ‘0’ trước xâu sau khi cộng

## 1.4. Phép chia

### 1.4.1. Thuật toán

Phép chia không là gì khác ngoài một phép trừ lặp đi lặp lại. Gọi phương thức trừ lặp đi lặp lại đó cho đến khi số đó nhỏ hơn số bị chia thì dừng lại. Số lần lặp ở đây sẽ là thương và số còn dư lại là số dư. Và nếu như từ đầu trị tuyệt đối của số a nhỏ hơn trị tuyệt đối của số b thì thương sẽ bằng 0 và dư bằng a

### 1.4.2. Các bước thực hiện

* Bước 1: Lấy số kí tự của xâu a bằng số kí tự của xâu b lưu vào xâu chia
* Bước 2: Chừng nào xâu chia còn lớn hơn xâu b thì tiến hành trừ liên tiếp xâu chia cho xâu b, ghi lại số lần trừ có thể là thương tìm được và kết quả trừ trước khi xâu chia bé hơn xâu b (vì đó là số dư)
* Bước 3: Hạ từng kí tự của xâu a xuống xâu chia

( Lưu ý: xử lí các kí tự “0” vô nghĩa của xâu a và xâu b )

# CHƯƠNG 2 : BÀI TOÁN THỰC HIỆN CÁC PHÉP TÍNH CƠ BẢN TRÊN SỐ NGUYÊN LỚN

## 2.1. Đặc tả bài toán

Số học là một nội dung rất quan trọng, nó giúp cho chúng ta hiểu cách biểu diễn và xử lí thông tin trong máy tính. Khi mới học lập trình, sinh viên bắt đầu bằng việc giải các bài toán số học cơ bản như: kiểm tra tính nguyên tố của một số nguyên dương, tìm ước chung lớn nhất của hai số nguyên dương, phân tích mọt số nguyên dương thành tích các thừa số nguyên tố, các bài toán chia hết, ... Có rất nhiều bài toán khó khi giải bằng các giải thuật trừu tượng khó hiểu đôi khi không hiệu quả. Tuy nhiên, có nhiều bài toán có thuật toán khá đơn giản nhưng đòi hỏi phải biểu diễn các số có giá trị lớn. Giá trị của nó có thể lên đến hàng trăm, hàng nghìn, thậm chí là hàng triệu chữ số. Nếu biểu diễn bằng các dữ liệu số thông thường như int , long, long long để áp dụng các phép toán số học gặp rất nhiều khó khăn, thậm chí là không biểu diễn được. Vì vậy, bài toán xử lí số nguyên lớn ra đời nhằm giải quyết vấn đề đó, cụ thể ở đây là bài toán thực hiện các phép tính cơ bản trên số nguyên lớn. Yêu cầu của bài toán đầu tiên là biểu diễn được các số nguyên lớn, sau đó là thực hiện các phép tính cơ bản trên đó một cách dễ dàng và đem lại kết quả chính xác nhất.

## 2.2. Yêu cầu hệ thống

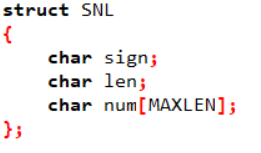
Thực hiện các phép tính cơ bản trên số nguyên lớn là chương trình đơn giản giúp người dùng cộng, trừ, nhân, chia trên số nguyên lớn. Ngoài ra, nó còn có thể mở rộng để giải quyết các bài toán phức tạp hơn.

Chương trình có thể được viết bằng nhiều ngôn ngữ lập trình khác nhau như C, C++, Python, Java... Nhưng trong bài toán này, ngôn ngữ đc sử dụng là C++.

## 2.3. Phân tích thiết kế chương trình

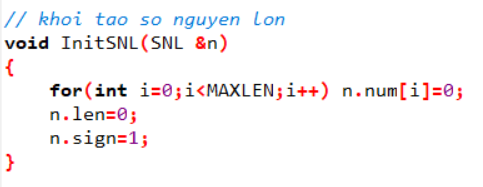
### 2.3.1. Cấu trúc chương trình

Ta dùng kiểu struct để lưu trữ các dữ liệu cần thiết cho việc mô tả một số nguyên lớn



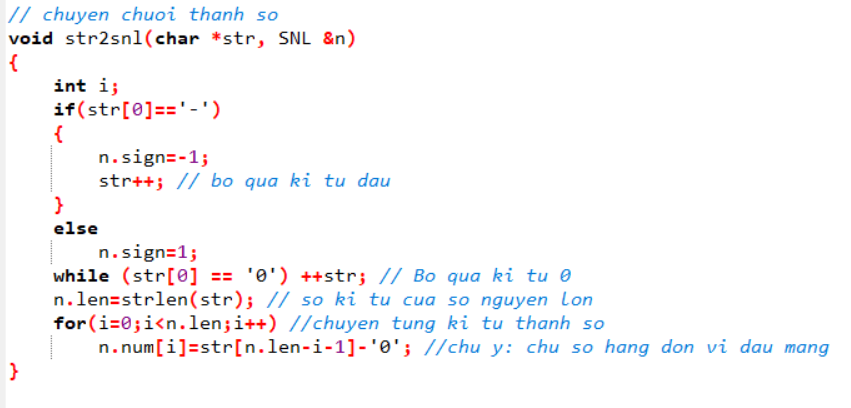
Tiếp theo để có được các số nguyên thì ta phải có 1 hàm khởi tạo số nguyên

* **Hàm khởi tạo số nguyên lớn**



* **Hàm chuyển chuỗi thành số**

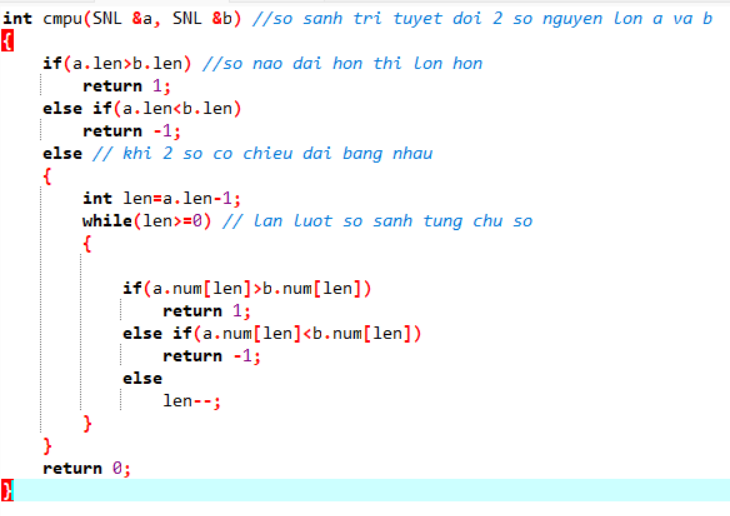
Vì trong struct ta dùng kiểu char để biểu diễn các biến nên ta phải dùng 1 hàm để chuyển chuỗi thành số



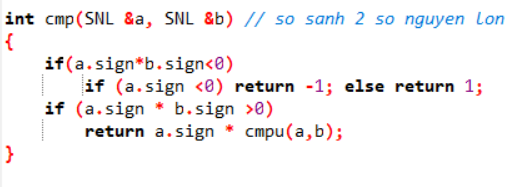
Để xác định dấu (là âm hay là dương) cho các phép tính cũng như để thực hiện được phép chia ta phải có hàm so sánh để làm điều đó

Ở đây, ta chia làm 2 hàm. Một hàm là so sánh trị tuyệt đối 2 số nguyên lớn, còn một hàm dùng để so sánh 2 số nguyên lớn. Việc chia làm 2 hàm sẽ giúp bài toán minh bạch, dễ dàng trong việc xử lí.

* **Hàm so sánh trị tuyệt đối 2 số nguyên lớn**



* **Hàm so sánh 2 số nguyên lớn**

****

Như đã thấy ở hàm trên, nếu a và b trái dấu nhau thì ta xét

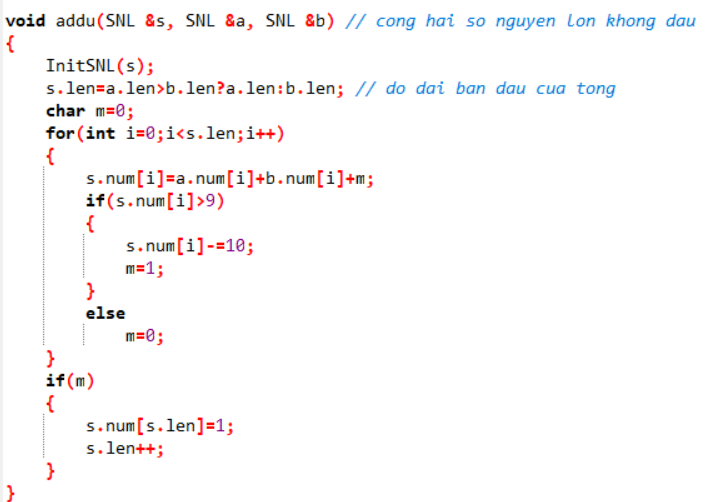
* nếu dấu của a là dấu âm thì trả về -1(tức a<b)
* nếu dấu của a là dấu dương thì trả về 1(tức a>b)

Ngược lại, nếu a và b cùng dấu thì phải so sánh trị tuyệt đối của a và b xem nó trả về bao nhiêu, sau đó nhân với a.sign để được kết quả

Và tất nhiên, trong bài toán thực hiện các phép tính trên số nguyên lớn thì không thể thiếu các hàm cơ bản là cộng, trừ, nhân, chia.

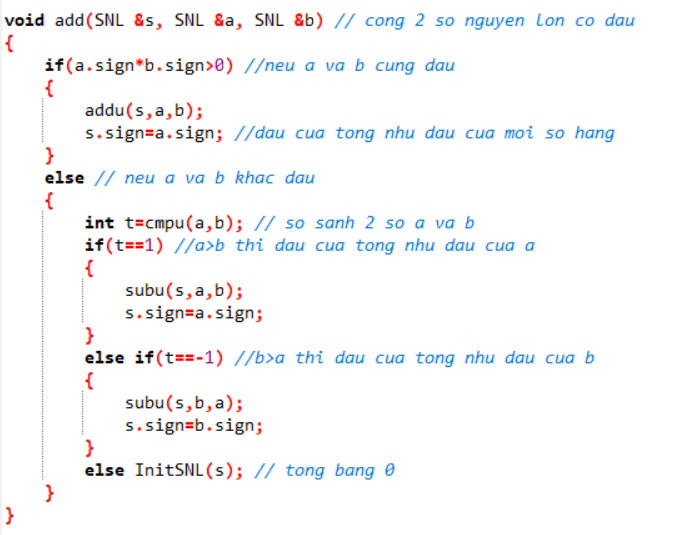
* **Hàm cộng 2 số nguyên lớn không dấu**

Làm như thuật toán nêu ở phần chương 1, ta được hàm cộng 2 số nguyên lớn không dấu



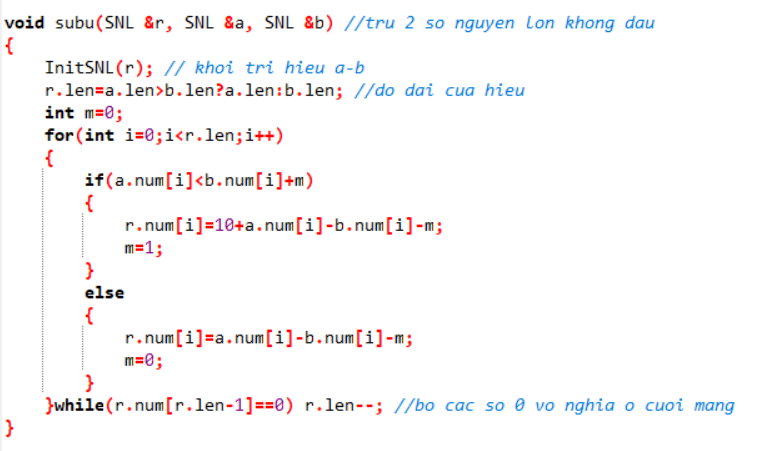
Tuy nhiên, để giải quyết bài toán cộng các số nguyên lớn thì như vậy là chưa đủ vì số nguyên ở đây có thể là số nguyên âm. Do đó, ta cần vận dụng hàm so sánh ở trên để xét dấu, hàm cộng 2 hai số nguyên lớn (đã trình bày ở trên) và hàm trừ 2 số nguyên lớn không dấu (sẽ trình bày ở phần dưới)

* **Hàm cộng 2 số nguyên lớn có dấu**



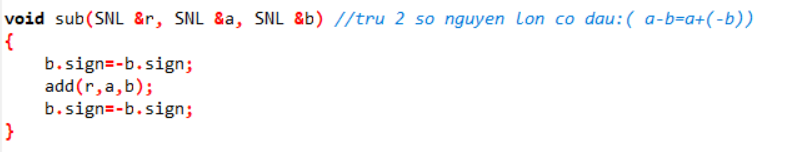
* **Hàm trừ hai số nguyên lớn không dấu**

Phép trừ ta sẽ làm tương tự như phép cộng do phép trừ chính là phép cộng cho số âm.



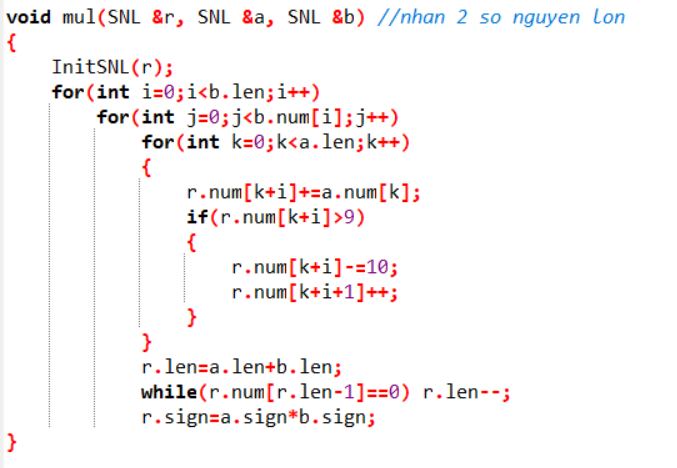
* **Hàm trừ hai số nguyên lớn có dấu**

Như đã nói ở trên, trừ chính là cộng cho số âm nên chỉ cần biến đổi hàm cộng một chút, ta sẽ có ngay hàm trừ hai số nguyên lớn có dấu



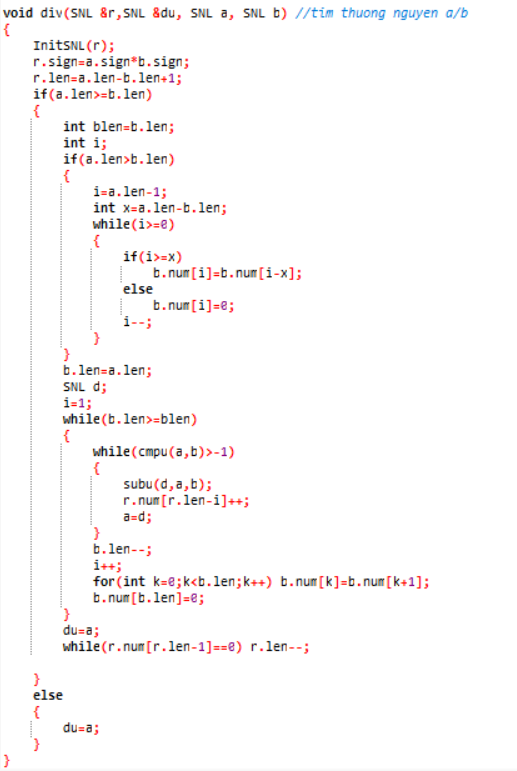
* **Hàm nhân hai số nguyên lớn**

Ta dùng 3 vòng lặp lồng nhau để có thể lấy từng số hạng của số b nhân với số a, sau đó cộng các kết quả đó lại với nhau. Và dấu của kết của kết quả sau khi nhân bằng dấu của a nhân dấu của b.



* **Hàm chia 2 số nguyên lớn**

Cuối cùng là chia 2 số nguyên lớn, đây là phép tính dài dòng và phức tạp nhất. Trong phép chia cần đến hàm so sánh và phép trừ.

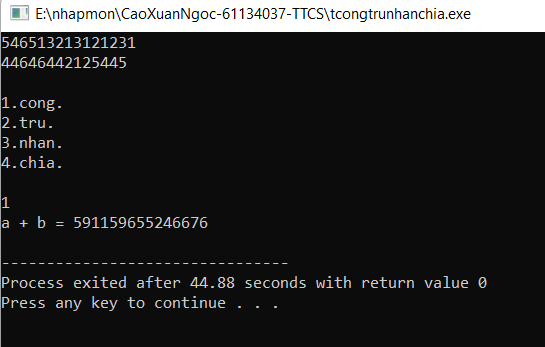


### 2.3.2. Các chức năng của chương trình

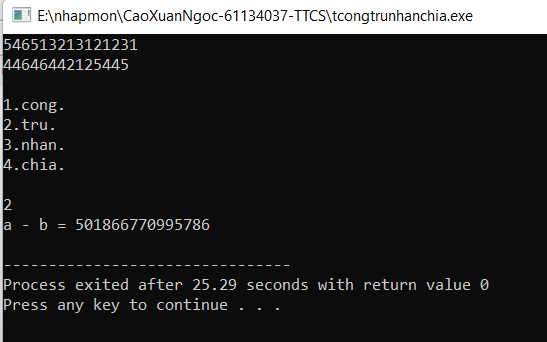
* Nhập các số nguyên lớn từ file input
* Chọn phép tính muốn thực hiện
* Xuất ra kết quả của các phép tính cộng, trừ, nhân, chia. Đối với phép chia sẽ in ra thương và số dư

## 2.4. Cài đặt chương trình

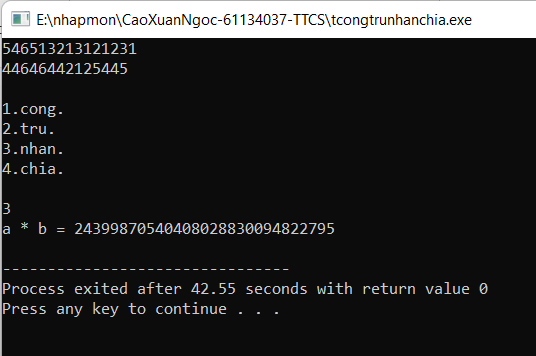
* Phép cộng



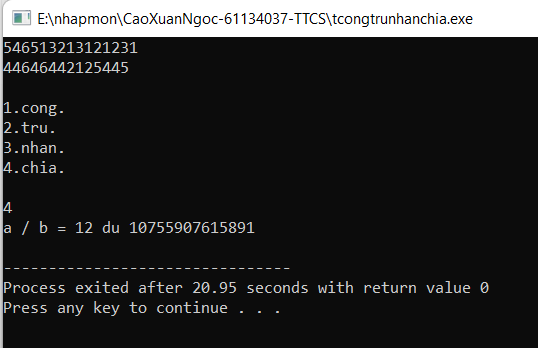
* Phép trừ



* Phép nhân



* Phép chia



# KẾT LUẬN

1. **Các kết quả đạt được**
   * Trình bày khái quát thuật toán dùng để tính toán trên số nguyên lớn
   * Xây dựng được bài toán thực hiện các phép tính cơ bản (cộng, trừ, nhân, chia) trên số nguyên lớn (có trên 20 chữ số)
2. **Ưu và nhược điểm của đề tài**

* Ưu điểm
  + Chương trình dễ sử dụng, giao diện thân thiện
  + Bài toán cộng trừ nhân chia các số nguyên lớn là có thể làm tiền đề để giải quyết các bài toán khó hơn, phức tạp hơn
* Nhược điểm:
  + Vẫn chưa chú trọng về đồ họa, mới chỉ đi về phần lập trình.
  + Code chương trình còn dài, chưa tối ưu

1. **Hướng phát triển**

Xây dựng thêm các tính năng như giao diện, thêm các phép tính phức tạp hơn như tính giai thừa, tính số mũ, tính căn....