ĐẠI HỌC BÁCH KHOA HÀ NỘI



KHOA TOÁN - TIN

**\*\*\*\*\*\*\*\*\*\***



**BÁO CÁO CUỐI KÌ**

**HỌC PHẦN: KĨ THUẬT LẬP TRÌNH – MI3310**

**Chủ Đề**: **Xây dựng thư viện làm việc với số nguyên lớn**

|  |  |
| --- | --- |
| Giảng viên hướng dẫn | : TS . Vũ Thành Nam |
| Họ và tên sinh viên | : Lê Quang Đức |
| MSSV | : 20227221 |
| Lớp | : Hệ thống thông tin 01-K67 |
| Mã lớp | : 150328 |
| Học kì | : 20232 |

**Mục lục**

Lời nói đầu......................................................................................................................................3  
I. Mô tả cấu trúc chương trình................................................................................................4  
 1. Giới thiệu về chương trình..............................................................................................4  
 2. Cấu trúc của chương trình...............................................................................................4  
II. Chức năng của chương trình................................................................................................5  
 1. Sơ đồ phân rã chức năng – FDD..................................................................................,..5  
 2. Xây dựng các mô hình chức năng...................................................................................6  
 2.1. Xây dựng hàm thực hiện chức năng chính (main)...................................................6  
 2.2. Xây dựng cấu trúc ‘bigint’.......................................................................................6  
 2.3. Xây dựng các hàm đọc/in kết qua ra màn hình/file.................................................7  
III. Thiết kế chương trình.......................................................................................................10  
 1. Phẩm chất của chương trình tốt....................................................................................10  
 2. Kỹ thuật thiết kế cấu trúc chương trình........................................................................11  
IV. Phong cách lập trình.........................................................................................................12  
 1. Khái niệm tổng quan....................................................................................................12  
 2. Trình bày mã nguồn.....................................................................................................12  
 3. Đặt tên và chú thích.....................................................................................................15  
V. Gỡ lỗi và kiểm thử...........................................................................................................17  
 1. Tìm kiếm và gỡ lỗi (Searching and Debugging) ........................................................17  
 1.1. Khái niệm.............................................................................................................17  
 1.2. Vận dụng trong chương trình bài toán ................................................................17  
 2. Kiểm thử (Testing)......................................................................................................18  
 2.1. Khái niệm.............................................................................................................18  
 2.2. Vận dụng trong chương trình bài toán.................................................................18  
 3. Các tình huống kiểm thử.............................................................................................19

**LỜI MỞ ĐẦU**

Trong bối cảnh công nghệ thông tin không ngừng phát triển, việc hiểu rõ và ứng dụng các kĩ thuật lập trình để giải quyết những vấn đề cụ thể là vô cùng cần thiết. Đặc biệt, trong lĩnh vực lập trình, các thư viện toán học đóng vai trò quan trọng, giúp tối ưu hóa quá trình xử lý và phân tích dữ liệu. Chính vì lẽ đó, môn học "Kĩ thuật lập trình" đã trang bị cho chúng ta những kiến thức nền tảng và kỹ năng thực hành quý báu để phát triển các ứng dụng và thư viện phần mềm hiệu quả.

Hướng tới mục tiêu áp dụng những kiến thức đã học vào thực tiễn, em chọn chủ đề "Xây dựng một thư viện toán học làm việc với số nguyên" cho bài báo cáo này. Thư viện toán học là một tập hợp các hàm và tiện ích giúp xử lý các phép toán cơ bản và nâng cao liên quan đến số nguyên, từ đó hỗ trợ lập trình viên trong việc triển khai các thuật toán một cách dễ dàng và hiệu quả hơn.

Em xin chân thành cảm ơn thầy Vũ Thành Nam, giảng viên chính lớp Kỹ thuật lập trình kỳ 2023.2 đã tận tình giảng dạy và hướng dẫn chúng em trong suốt học kì vừa qua. Em không chỉ được tiếp thu và nâng cao được kiến thức về môn học này mà còn được thầy trau dồi về những kỹ năng khác để vận dụng vào những dự án thực tế và xây dựng cách tự học làm sao cho hiệu quả. Tuy bài báo cáo đã được em chuẩn bị cẩn thận nhưng sẽ không thể tránh được những sai sót. Rất mong sau khi thực hiện, em sẽ có cơ hội được gặp lại và nghe thầy đưa ra những đóng góp, góp ý để bài báo cáo của em được hoàn thiện hơn.

Em xin chân thành cảm ơn !

Hà Nội, ngày 16 tháng 6 năm 2024

Sinh viên thực hiện

Lê Quang Đức

1. **Mô tả cấu trúc chương trình**
2. **Giới thiệu về chương trình**

Chương trình mà cá nhân em thực hiện là xây dựng một thư viện phép toán với số nguyên lớn theo các yêu cầu cụ thể sau:

* Số nguyên có độ lớn lên tới 1000 chữ số
* Nhận các số hạng từ bàn phím hoặc đọc từ file
* Thực hiện các phép tính toán: cộng, trừ, nhân, chia
* Hiển thị thông tin phép tính ( bao gồm kết quả), lưu ra file.

Và sau khi cân nhắc và dựa trên kiến thức mà em đã học được thì em đã thực hiện chương trình bằng ngôn ngữ lập trình C++.

Tất cả các yêu cầu của thầy đã được em thực hiện trong bài báo cáo. Ngoài ra em đã thực hiện thêm một số phần để hoàn hiện chương trình hơn

* Thêm vào chương trình các hàm dùng để so sánh ( <,>,=,>=.<= ) giữa hai số nguyên lớn
* Thêm vào chương trình các hàm dùng thực hiện tìm ước chung lớn nhất, bội chung nhỏ nhất, tìm trị tuyệt đối của số nguyên âm…
* Sử dụng thuật toán Karatsuba để nhân hai số nguyên lớn giúp giảm độ phức tạp tính toán học
* Thêm vào các hàm hỗ trợ in kết quả ra màn hình và ra file…

1. **Cấu trúc của chương trình**

Đầu tiên em đã phát triển chương trình của em có cấu trúc như sau:

BTL/:

├── main.cpp

│ ├ Struct ‘bigint’ # Định nghĩa cấu trúc ‘bigint’ và các phép toán

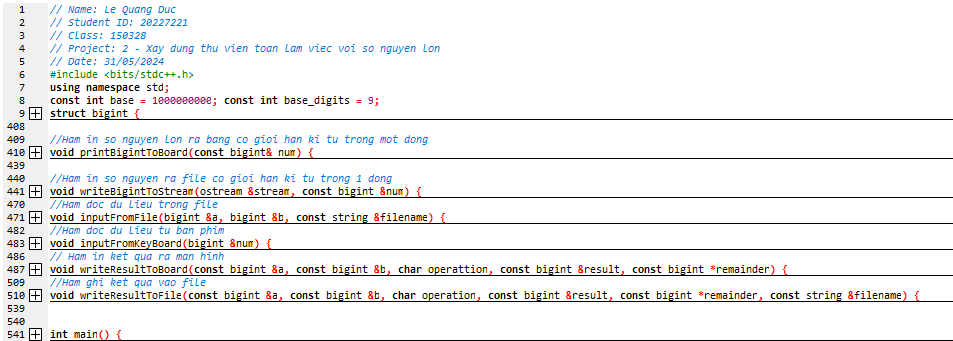
│ ├void /\* Các hàm thực hiện chức năng in kết quả \*/ ()

│ └Hàm main // Hàm chính để thực hiện chương trình chính

└── Kiem\_thu/

├── Ket\_Qua.txt //File để lưu kết quả

└── (Các trường hợp kiểm thử) .txt //File các trường hợp kiểm thử.



1. **Chức năng của chương trình**
2. **Sơ đồ phân rã chức năng – FDD**

Sơ đồ phân rã chức năng (Functional Decomposition Diagram) là một công cụ phân tích được sử dụng để phân chia các chức năng chính của một hệ thống hoặc ứng dụng thành các phần nhỏ hơn, dễ quản lý hơn. Nó bao gồm các mức chức năng liên quan đến nhau, từ mức cao nhất đại diện cho chức năng chính của hệ thống cho đến các mức dưới đại diện cho các chức năng con chi tiết hơn. Sơ đồ phân rã chức năng giúp hiểu rõ hơn cấu trúc tổ chức của hệ thống và mối quan hệ giữa các chức năng khác nhau.

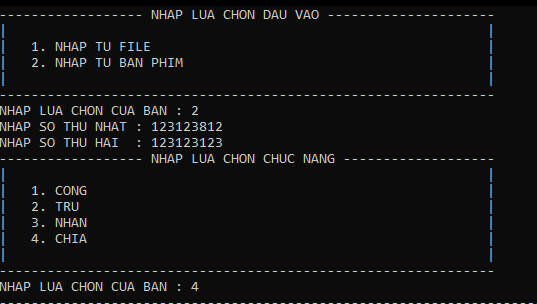
Dưới đây là sơ đồ phân rã chức năng trong thư viện làm việc với số nguyên lớn mà báo cáo này đang đề cập tới:

A diagram of a company

Description automatically generated

1. **Xây dựng các mô hình chức năng** 
   1. **Xây dựng hàm thực hiện chức năng chính chương trình (main)**

* Thực hiện nhập hai số nguyên lớn có thể từ bàn phím hoặc đọc từ file (.txt) vào để thực hiện các lệnh tính toán
* Tạo ra một giao diện dòng lệnh cho phép người dùng thực hiện một số yêu cầu sử dụng các chức năng của cấu trúc và kết thúc chương trình hệ thống



*Bảng lựa chọn*

* 1. **Xây dựng cấu trúc ‘bigint’ ( struct bigint () )**

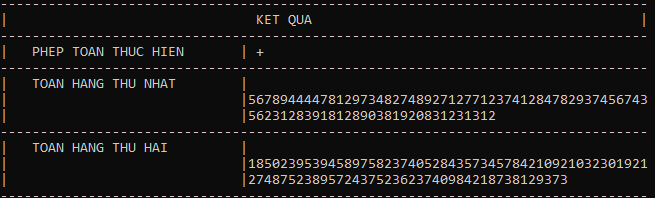
Bao gồm định nghĩa về ‘bigint’ và các phép toán làm việc với số nguyên lớn

* Constructor và Gán giá trị
* **bigint()**: Khởi tạo một đối tượng bigint mặc định với giá trị là 0.
* **bigint(long long v)**: Khởi tạo một đối tượng bigint từ một số nguyên dài (long long).
* **bigint(const string &s)**: Khởi tạo một đối tượng bigint từ một chuỗi ký tự.
* Gán giá trị
* **void operator=(const bigint &v)**: Gán giá trị từ một đối tượng bigint khác..
* **void operator=(long long v)**: Gán giá trị từ một số nguyên dài (long long).
* Các phép toán số học
* **bigint operator+(const bigint &v) const**: Cộng hai đối tượng bigint..
* **bigint operator-(const bigint &v) const**: Trừ hai đối tượng bigint.
* **bigint operator\*(int v) const :**Nhân đối tượng bigint hiện tại với một số nguyên (int), trả về kết quả mới.
* **pair<bigint, bigint> divideBigNum(const bigint &b) const**: Chia đối tượng bigint hiện tại cho một đối tượng bigint khác, trả về một cặp số gồm thương và số dư.
* **bigint operator/(const bigint &v) const**: Chia hai đối tượng bigint, trả về thương.
* **bigint operator%(const bigint &v) const**: Chia hai đối tượng bigint, trả về số dư.
* **void operator/=(int v)**: Chia đối tượng bigint hiện tại cho một số nguyên (int).
* **bigint operator/(int v) const**: Chia đối tượng bigint hiện tại cho một số nguyên (int), trả về kết quả mới.
* **int operator%(int v) const**: Chia đối tượng bigint hiện tại cho một số nguyên (int), trả về số dư.
* Các phép toán và gán
* **void operator+=(const bigint &v)**: Cộng và gán kết quả cho đối tượng bigint hiện tại.
* **void operator-=(const bigint &v)**: Trừ và gán kết quả cho đối tượng bigint hiện tại.
* **void operator\*=(const bigint &v)**: Nhân và gán kết quả cho đối tượng bigint hiện tại.
* **void operator/=(const bigint &v)**: Chia và gán kết quả cho đối tượng bigint hiện tại.
* Chuyển đổi
* **long long longValue() const**: Chuyển đổi đối tượng bigint thành số nguyên dài (long long).
* **Void read(const string &s): Đọc giá trị từ một chuỗi kí tự vào đối tượng bigint**
* Một số tiện ích khác
* **void trim()**: Loại bỏ các số 0 không cần thiết ở đầu**.**
* **bigint operator-() const**: Đổi dấu đối tượng bigint.
* **bigint abs() const**: Trả về giá trị tuyệt đối của đối tượng bigint.
* **friend bigint gcd(const bigint &a, const bigint &b)**: Tìm ước chung lớn nhất (GCD) của hai đối tượng bigint.
* **friend bigint lcm(const bigint &a, const bigint &b)**: Tìm bội chung nhỏ nhất (LCM) của hai đối tượng bigint.
* **static vll karatsubaMultiply(const vll &a, const vll &b)** : Thuật toán Karatsuba
* **Các hàm so sánh hai số ( <,>,>=,<=,==,!=)**
  1. **Xây dựng các hàm đọc/in kết quả ra màn hình/file**

Ngoài cấu trúc ‘bigint’ và hàm thực hiện chức năng chính thì còn các hàm giúp đọc dữ liệu từ file hoặc được nhập từ bàn phím và trả lại kết quả ra màn hình/file.

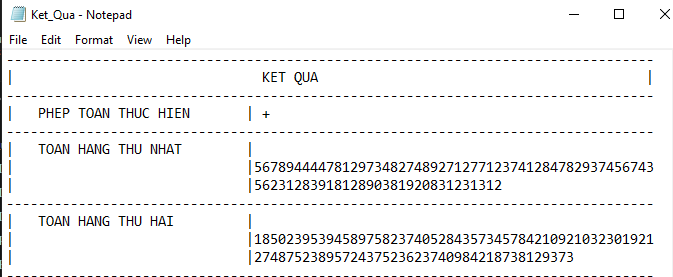
* Hàm ghi số nguyên lớn ra màn hình ( writeBigIntToBoard () )

Người dùng sử dụng bàn phím để nhập số nguyên lớn, đối với số nguyên không quá lớn ( khoảng 10 chữ số ) thì khi ghi số ra màn hình sẽ không có vấn đề gì. Nhưng đối với số nguyên lớn ( lớn hơn 100 chữ số ) thì cần một hàm hỗ trợ để ghi ra số màn hình để dễ quan sát, tránh gây ra hiện tượng tràn số .

. *Ví dụ về hàm ghi số nguyên lớn ra màn hình*

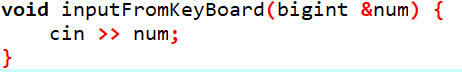
* Hàm ghi số nguyên lớn vào file ( writeBigIntToFile () )

Tương tự như hàm ghi số nguyên lớn ra màn hình, cần có một hàm hỗ trợ để ghi số ra file mà không gây ra hiện tượng tràn số.



*Ví dụ về hàm ghi số nguyên lớn ra màn hình*

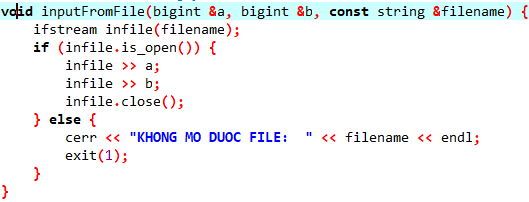
* Hàm đọc dữ liệu từ bàn phím ( inputFromBoard () )

Người dùng lựa chọn nhập số từ bàn phím theo cách phổ thông  


*Chương trình viết hàm Đoạn chương trình mô tả hàm nhập từ bàn phím*

* Hàm đọc dữ liệu từ file ( inputFromFile () )

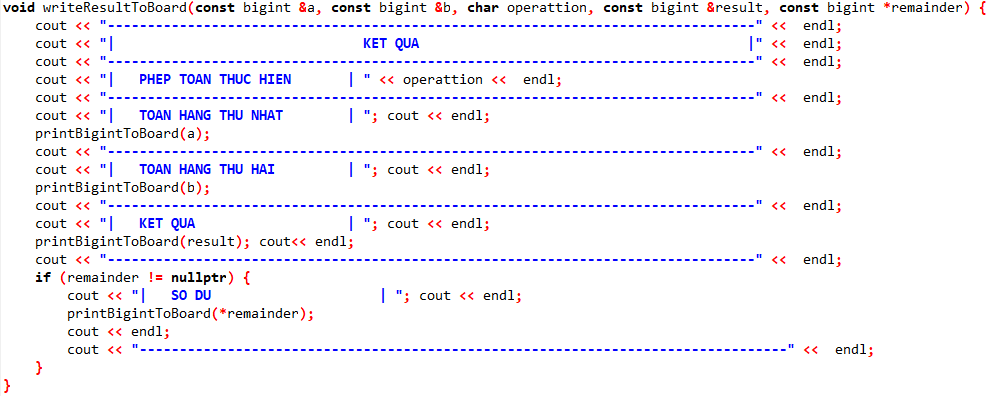
Người dùng lựa chọn nhập số từ file (.txt) đã có với dòng 1 là số hạng thứ nhất và dòng 2 là số hạng thứ hai.



*Đoạn chương trình mô tả hàm đọc từ file*

* Hàm in kết quả ra màn hình ( writeResultToBoard () )

Sau khi người dùng đã nhập hai số nguyên lớn và nhập toán tử thực hiện thì chương trình sẽ bắt đầu tính toán. Hàm writeResultToBoard giúp việc in kết quả ra màn hình dễ quan sát hơn

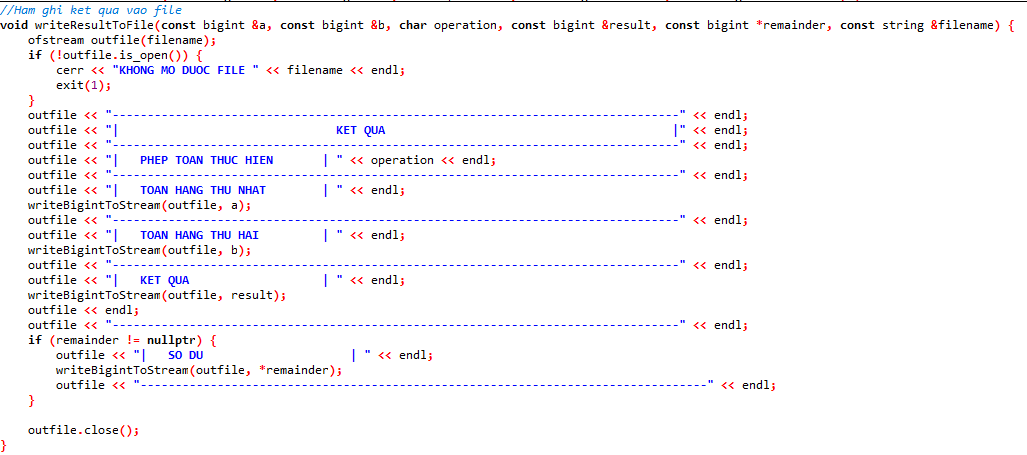


Đoạn chương trình mô tả hàm in kết quả ra màn hình

* Hàm in kết quả ra file ( writeResultToFile() )

Tương tự như hàm in kết quả ra màn hình, hàm in két quả ra file cũng có chức

năng tương tự



*Đoạn chương trình mô tả hàm in kết quả ra file*

1. **Thiết kế chương trình**
2. **Phẩm chất của chương trình tốt**
3. **Cấu trúc tốt.**

* **Chia nhỏ các hàm và module**: Chương trình được chia thành các hàm nhỏ, mỗi hàm thực hiện một nhiệm vụ cụ thể như inputFromFile, inputFromKeyBoard, writeResultToBoard, writeResultToFile, printBigintToBoard. Điều này giúp dễ dàng bảo trì và mở rộng mã nguồn.
* **Định nghĩa lớp rõ ràng**: Lớp bigint được định nghĩa rõ ràng với các thuộc tính và phương thức cụ thể, giúp dễ dàng quản lý các số nguyên lớn.

1. **Logic chương trình và các biểu thức được diễn đạt theo cách thông thường.**

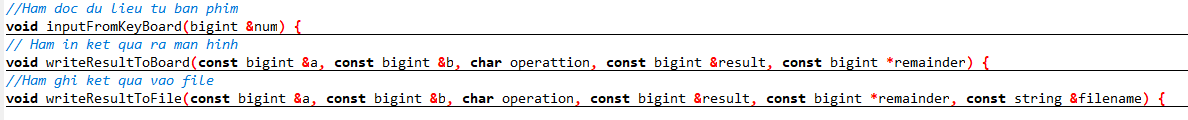
* **Logic rõ ràng**: Các phép toán số học trên số nguyên lớn (cộng, trừ, nhân, chia) được triển khai một cách rõ ràng và tuần tự. Các bước trong phép chia và nhân Karatsuba được tách biệt và có logic rõ ràng.
* **Biểu thức dễ hiểu**: Các biểu thức trong chương trình được viết theo cách thông thường, dễ hiểu và dễ theo dõi.

1. **Tên dùng trong chương trình có tính chất miêu tả.**

* **Tên biến và hàm**: Các tên biến và hàm được đặt một cách có ý nghĩa và miêu tả rõ ràng chức năng của chúng. Ví dụ: inputFromFile, inputFromKeyBoard, writeResultToBoard, writeResultToFile, printBigintToBoard, divideBigNum.
* **Tên lớp và thành viên**: Lớp bigint và các thành viên như ‘sign’, ‘a’, ‘read', ‘trim’ có tên rõ ràng và dễ hiểu.

1. **Chú thích hợp lý.**

* Các hàm đã có chú thích hợp lí giúp cho người đọc có thể hiểu rõ hơn chức năng của từng hàm



*Chú thích hợp lí và tên dùng trong chương trình có tính chất miêu tả*

1. **Tôn trọng chiến lược Divide/Conquer/Association.**

* **Chia nhỏ vấn đề**: Chương trình chia các phép toán số học phức tạp thành các hàm nhỏ hơn và thực hiện từng bước một cách tuần tự.   
  Ví dụ, phép nhân Karatsuba chia nhỏ vấn đề thành các bài toán con nhỏ hơn và giải quyết từng bài toán con.
* **Gộp kết quả**: Kết quả của các bài toán con sau đó được gộp lại để có được kết quả cuối cùng. Điều này thể hiện rõ trong các phép toán như nhân và chia.

1. **Kỹ thuật thiết kế cấu trúc chương trình**

Việc sử dụng và hiểu được các kỹ thuật thiết kế cấu trúc khác nhau trong lập trình là rất quan trọng. Với bài toán làm việc với số nguyên lớn được đưa ra, em đã xác định được các phương thức phù hợp và kết hợp chúng với nhau để xây dựng chương trình hoàn chỉnh, sao cho dễ dàng thao tác và hoàn thành. Các kỹ thuật đã được em sử dụng trong bài như sau:

* Mô thức lập trình hướng đối tượng (OOP)
* Lớp ‘bigint’ : Chương trình sử dụng lớp đối tượng ‘bigint’ để quản lý các số nguyên lớn, bao gồm các thuộc tính và phương thức để thao tác với các số nguyên lớn
* Thuộc tính:
  + ‘vector<int> a’: Lưu trữ các đoạn của số nguyên lớn
  + ‘int sign’ : Lưu trữ dấu của số.
* Bao gồm các phương thức khởi tạo (‘bigint()’, bigint(long long v)’,...) , các toán tử nạp chồng (‘operator=’, ‘operator+’,..) và các phương thức tiện ích (‘read’, ‘trim’, ‘abs’…)
* Thể hiện được 4 tính chất của mô thức lập trình hướng đối tượng, là tính đóng gói, tính kế thừa, tính trừu tượng và tính đa hình.
* Tôn trọng và tuân thủ tính cấu trúc của mô thức này.  
  Mô thức lập trình hướng đối tượng (OOP)
* Tiếp cận với giải thuật module hóa ( Modularity)
* Module hóa các nhiệm vụ chức năng nhỏ để có thể dễ dàng đọc hiểu và tiếp cận.
* Chương trình được chia thành nhiều hàm và phương thức nhỏ, mỗi cái thực hiện một nhiệm vụ cụ thể.

Vd: inputFromFile, inputFromKeyBoard, writeResultToBoard, writeResultToFile, printBigintToBoard, v.v.

* Điều này giúp mã nguồn dễ đọc, dễ bảo trì và dễ mở rộng.
* Phương pháp thiết kế Top-Down
* Chương trình chính (main) được thiết kế để gọi các hàm con cho từng nhiệm vụ cụ thể như nhập dữ liệu, chọn phép toán, thực hiện phép toán, và hiển thị kết quả.
* Xác định các chức năng của chương trình cần thể hiện. Sau đó xây dựng các hàm nhiệm vụ nhỏ, từ các hàm nhiệm vụ nhỏ lại chia thành nhiều chức năng nhiệm vụ nhỏ hơn đóng gói trong nó.
* Những lệnh đơn giản sẽ trực tiếp thiết kế code. Đối với những công việc thực hiện phức tạp hơn sẽ thay bằng lời gọi hàm.
* Điều này giúp chương trình trở nên rõ ràng và dễ theo dõi, với mỗi phần chức năng được tách biệt rõ ràng.
* Thiết kế dữ liệu
* Sử dụng lớp số nguyên lớn mô hình hóa thực thể cần được đặc tả.
* Đầy đủ các thuộc tính của số nguyên
* Các file dữ liệu được hoạt động hiệu quả, đầu vào dữ liệu tốt.

1. **Phong cách lập trình**
2. **Khái niệm tổng quan**

* Xây dựng chương trình tốt
* Nhất quán: Tuân thủ quy tắc đặt tên trong bộ chương trình, nhất quán trong việc dùng biến cục bộ
* Bao đóng: Hàm chỉ nên tác động tới duy nhất giá trị trả về của hàm.
* Rõ ràng: Chú thích rõ rằng ở đầu mỗi chương trình con
* Khúc triết: Mỗi chương trình con đều có một nhiệm vụ rõ ràng, đủ ngắn để có thể nắm bắt được, số tham số của chương trình con là tối thiểu
* Quy ước lập mã
* Quy ước lập mã là một tập hợp các quy tắc được định nghĩa ra để quy ước quá trình code trong một dự án.
* Dựng lên một bộ quy tắc để thống nhất chung cho hệ thống hoặc dự án
* Thuận tiện hơn cho những lập trình viên khác khi họ tìm hiểu dự án các dự án cũ để phát triển thêm.
* Nhờ có quy ước lập mã, những các thành viên trong dự án có thể hiểu đọc hiểu code của nhau dễ dàng hơn.

1. **Trình bày mã nguồn**

Trong source code được xây dựng của chương trình làm việc với số nguyên trên, em đã sử dụng những phương pháp, kỹ năng viết code, trình bày code như sau:

* Sử dụng khoảng trắng, cách lề, cách đoạn
* Các khối mã được căn lề rõ ràng, giúp dễ dàng nhận biết các cấu trúc điều khiển và phạm vi của các khối mã.

VD: Hàm main

A screen shot of a computer code

Description automatically generated

* Có khoảng trắng giữa các toán tử và toán hạng, giữa các từ khóa và dấu ngoặc, giúp mã nguồn trở nên rõ ràng hơn.  
  VD: Trong struct bigint

A screenshot of a computer code

Description automatically generated

* Mã nguồn được chia thành các đoạn rõ ràng, với các dòng trắng giữa các phần logic khác nhau, giúp dễ đọc và theo dõi.

VD: Các hàm được cách nhau bởi dòng trắng, giúp tách biệt các chức năng khác nhau trong mã.

A screenshot of a computer code

Description automatically generated

* **Trình bày biểu thức điều kiện**
* **Dùng các biểu thức dạng nguyên bản**: Các biểu thức toán học và logic được giữ nguyên dạng, không bị rút gọn quá mức, giúp dễ đọc và hiểu.

**VD: Hàm “bigint operator\* (const bigint &v) const “**

A close up of numbers

Description automatically generated

* **Sử dụng dấu ngoặc để tránh nhầm lẫn**: Dấu ngoặc được sử dụng một cách hợp lý để làm rõ thứ tự thực hiện các phép toán, tránh nhầm lẫn và lỗi logic.

**VD: Hàm “bigint operator\* (const bigint &v) const “**

A close up of numbers

Description automatically generated

* **Trình bày dễ đọc**: Các biểu thức được căn lề và cách dòng hợp lý, giúp mắt dễ dàng theo dõi và hiểu rõ các bước tính toán.

1. **Đặt tên và chú thích**

* Quy tắc đặt tên
* Tên biến và hàm có ý nghĩa.

Các tên biến và hàm được đặt rõ ràng và có ý nghĩa, phản ánh đúng mục đích và nội dung của chúng. Tên biến phải có ý nghĩa, có tính gợi nhớ, đặc tả chức năng của nó. Tên hàm thể hiện đầy đủ chức năng nó thực hiện, sử dụng động từ làm tên hàm

VD: Đặt tên cho hàm

A computer code with text

Description automatically generated with medium confidence

VD: Đặt tên cho biến



* Sử dụng kiểu camelCase và snake\_case

camel Case: Chữ cái đầu tiên trong từ định danh là chữ thường và chữ cái đầu tiên của mối từ nối theo sau phải được viết hoa.

snake\_case: Tách các từ trong tên bởi gạch dưới \_ và hầu hết khi sử dụng các từ viết chữ in thường.

VD: Camel Case

A screen shot of a computer code

Description automatically generated

VD: snake\_case



* Tên biến tạm thời rõ ràng

Các biến tạm thời như cur, carry được sử dụng ngắn gọn nhưng vẫn dễ hiểu trong ngữ cảnh.:

VD:

A close up of numbers

Description automatically generated

Trong ví dụ này:

- ‘cur’ là giá trị hiện tại của số hạng trong quá trình tính toán

- ‘carry’ là giá trị “nhớ” được cộng thêm vào giá trị hiện tại

* Quy tắc chú thích
* Chú thích cho từng hàm về nhiệm vụ, chức năng
* Chú thích những chức năng nhỏ của hàm.
* Viết những chú thích tương ứng với code và thay đổi khi bản thân code thay đổi

VD:

A screenshot of a computer program

Description automatically generated

1. **Gỡ lỗi và kiểm thử**
2. **Tìm kiếm và gỡ lỗi ( Searching and Debugging)**
   1. **Khái niệm**

* Là quá trình tìm ra lỗi khiến chương trình máy tính hay hệ thống không hoạt động đúng.
* Khi chương trình bị lỗi, debug là công việc cần làm để làm cho chương trình chạy thông.
* Tìm kiếm và gỡ lỗi là công việc quan trọng cần những rèn luyện qua nhiều kỹ năng, kinh nghiệm như chia để trị, sử dụng debugger, tập trung viết mã mới nhất,...
* Cần hiểu được các thông báo lỗi, đặt vấn đề, câu hỏi: Lỗi ở đâu, xảy ra như thế nào, nguyên nhân xảy ra lỗi,...
  1. **Vận dụng trong chương trình bài toán**

Trong quá trình thiết kế chương trình cho bài toán, em đã sử dụng những kỹ thuật tìm kiếm và gỡ lỗi cho đoạn code như sau:

* Sử dụng kỹ thuật debug chính là hiển thị thông tin. Ví dụ:
* Sau khi nhập các số từ file (.txt) thực hiện lệnh tính toán thì chương trình lỗi. Em thực hiện chạy lại chương trình để phát hiện lỗi trả về trên terminal và thực hiện chỉnh sửa hàm xử lý input của file
* Các câu lệnh sau khi đã test hoặc gỡ lỗi sẽ được đánh dấu lại để dễ dàng trong việc kiểm lỗi
* Tiếp tục chạy thử chương trình để hiển thị các thông tin, nếu hiển thị thông tin sai thì tiến hành debug lần nữa.
* Viết thêm các hàm để kiểm tra lỗi. Khi lỗi đã được phát hiện và gỡ lỗi sẽ chú thích lại hàm đó.
* Trải qua các phiên bản khác nhau và sửa các lỗi ở các hàm, vòng lặp, đưa ra source code cuối cùng tối ưu và tránh các lỗi sai nhất.

1. **Kiểm thử (Testing)**
   1. **Khái niệm**

* Là quá trình kiểm tra hay đánh giá 1 hệ thống hay 1 thành phần hệ thống một cách thủ công hay tự động để kiểm chứng rằng nó thỏa mãn những yêu cầu đặc thù hoặc để xác định sự khác biệt giữa kết quả mong đợi và kết quả thực tế.
* Là quá trình thực hiện 1 chương trình với mục đích tìm ra lỗi.
* Trong quá trình phát triển code, có thể nảy sinh các vấn đề như tính toán không đúng, quan hệ các dữ liệu sai, dòng lệnh chạy sai mục đích,... Lập trình viên cần sử dụng các kỹ thuật testing, thử các trường hợp có khả năng gây ra lỗi, kết hợp debugging gỡ lỗi.
  1. **Vận dụng trong chương trình bài toán**
* Kiểm thử hộp đen ( Black Box Testing )
* Tập trung vào đầu vào và đầu ra để kiểm thử chức năng của chương trình mà không cần biết mã nguồn bên trong
* VD: Kiểm thử biênA computer code with numbers and symbols

  Description automatically generated
* VD: Kiểm thử giá trị đặc biệt

A white background with black text and red and blue letters

Description automatically generated

* Kiểm thử tích hợp ( Intergration Testing )
* Kiểm tra sự tương tác giữa các module hoặc thành phần trong hệ thống để đảm bảo chúng hoạt động cùng nhau một cách chính xác
* VD: Kiểm tra việc tích hợp các hàm cộng và nhân của lớp ‘bigint’ trong một hệ thống lớn hơn, đảm bảo rằng khi các hàm này được sử dụng cùng nhau, chúng vẫn cho ra kết quả chính xác.

A computer code with numbers and symbols

Description automatically generated

* Kiểm thử hộp trắng ( White Box Testing )
* Kiểm thử hộp trắng là kỹ thuật kiểm thử dựa trên cấu trúc bên trong của mã nguồn. Người kiểm thử phải hiểu và kiểm tra từng phần của mã nguồn.
* Kiểm Thử Đường Dẫn: Đảm bảo tất cả các đường dẫn có thể xảy ra trong hàm operator+ được kiểm tra.
* Kiểm Thử Nhánh: Đảm bảo tất cả các điều kiện và nhánh của if được kiểm tra.

1. **Các tình huống kiểm thử**

**Test case 00: Khởi động chương trình**

* In ra màn hình console bảng thiết kế danh sách các chức năng của chương trình
* In ra các số để người dùng chọn phương thức sử dụng chương trình

A black screen with white text

Description automatically generated

*Bảng lựa chọn phương thức nhập*

**Test case 01: Nhập các số từ file**

* Input: Đường dẫn đến file có hai số nguyên lớn
* Output: Bảng chọn chức năng

A screenshot of a computer

Description automatically generated

*Bảng lựa chọn chức năng*

**Test case 02: Nhập các số từ file bị lỗi hoặc không tồn tại**

* Input: Đường dẫn sai hoặc không tồn tại
* Output: Thông báo lỗi và kết thúc chương trình
* Mô tả kiểm thử:
* Ở phần input nhập một link không dẫn đến file chứa số nguyên
* Kết quả trả về thông báo “KHONG MO DUOC FILE” và kết thúc

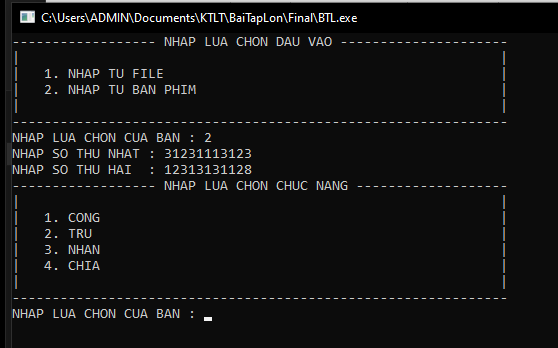
A screenshot of a computer program

Description automatically generated

*Minh họa lỗi nhập file*

**Test case 03: Nhập các số từ bàn phím**

* Input: Nhập 2 số nguyên lớn từ bàn phím
* Output: Bảng lựa chọn chức năng



*Bảng chức nănng*

**Test case 04: Nhập một lựa chọn khác 1 và 2**

* Input: Nhập lựa chọn khác 1 và 2 ở bảng lựa chọn đầu vào
* Output: Thông báo lỗi và dừng lại

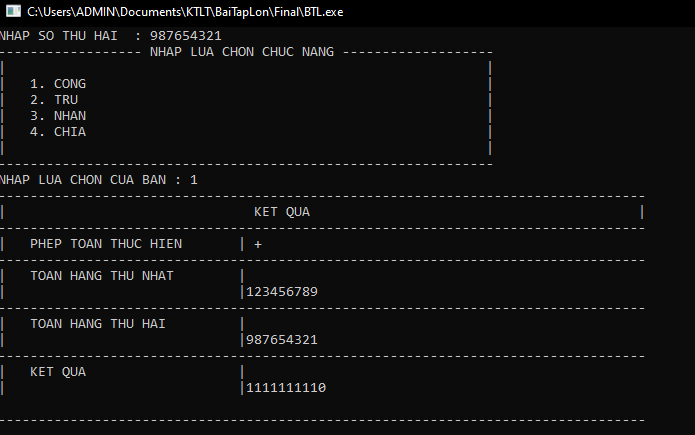
A screenshot of a computer

Description automatically generated

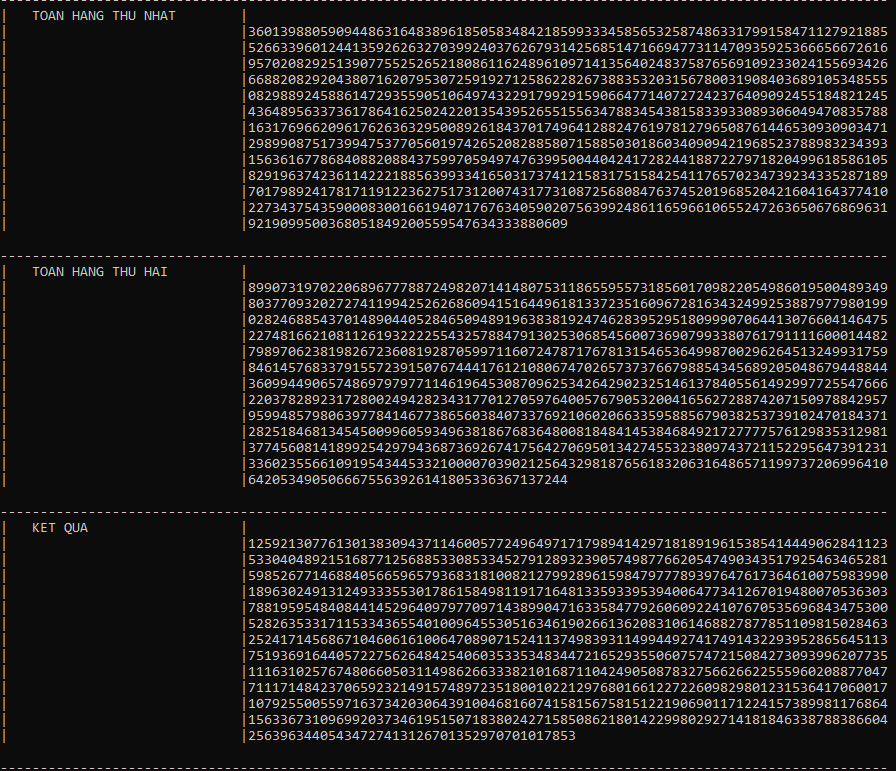
*Mô tả lựa chọn không hợp lệ*

**Test case 05: Cộng hai số nguyên dương**

* Sau khi nhập hai số nguyên dương từ bàn phím/đọc từ file thì thực hiện lựa chọn chức năng
* Input: Nhập số ứng với lựa chọn ( lựa chọn 1: thực hiện phép cộng )
* Output: Trả về kết quả của phép tính

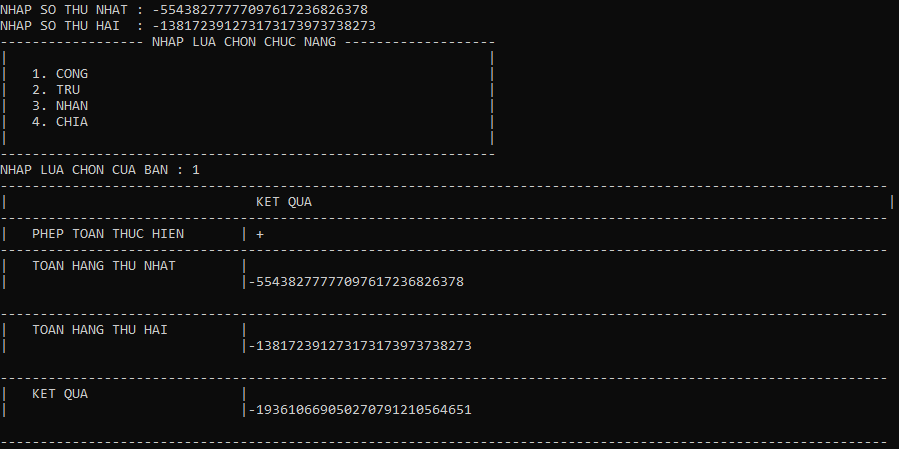


*Phép cộng hai số nguyên dương từ bàn phím*

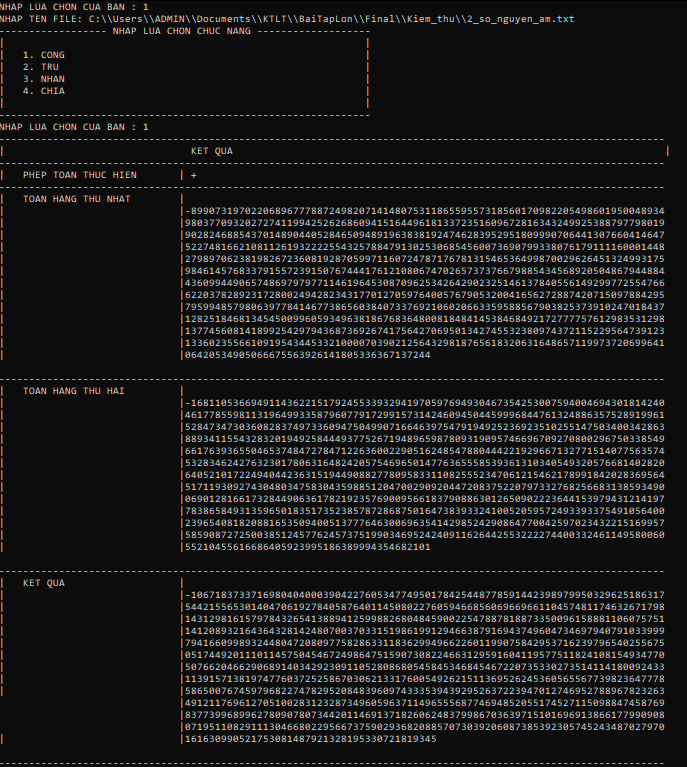
**

*Phép cộng hai số nguyên dương đọc từ file*

**Test case 06: Cộng hai số nguyên âm**

* Sau khi nhập hai số nguyên âm từ bàn phím/đọc từ file thành công thì tiếp tới ta chọn chức năng tính toán
* Input: Nhập số ứng với lựa chọn ( lựa chọn 1: thực hiện phép cộng )
* Output: Trả về kết quả của phép tính 

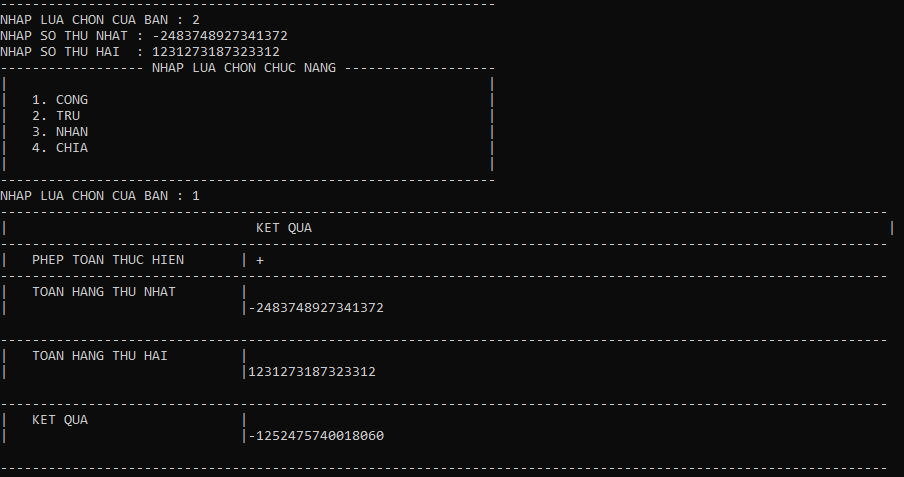
*Phép cộng hai số nguyên âm nhập từ bàn phím*



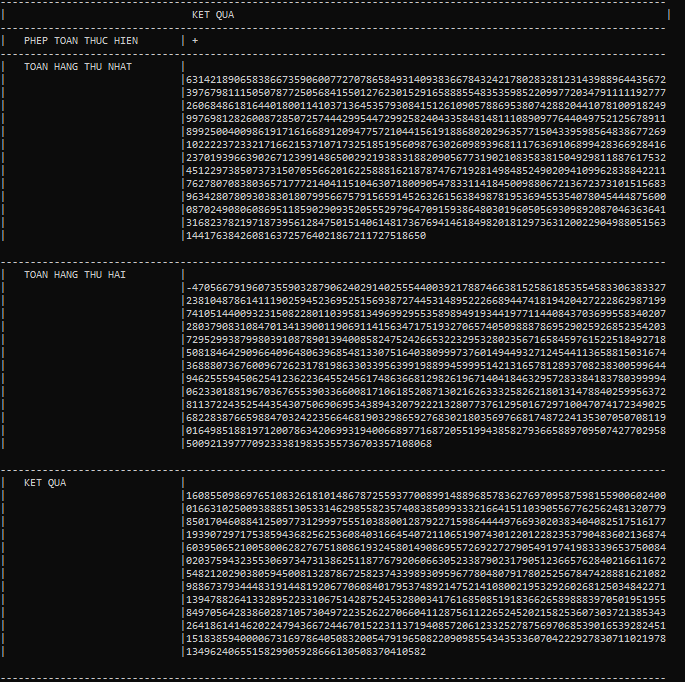
*Phép cộng hai số nguyên âm đọc từ file*

**Test case 07: Cộng hai số nguyên trái dấu**

* Sau khi nhập hai số nguyên trái dấu từ bàn phím/đọc từ file thành công thì tiếp tới ta chọn chức năng tính toán
* Input: Nhập số ứng với lựa chọn ( lựa chọn 1: thực hiện phép cộng )
* Output: Trả về kết quả của phép tính

**

*Phép cộng hai số nguyên trái dấu nhập từ bàn phím*

**

*Phép cộng hai số nguyên trái dấu đọc từ file*

**Test case 08: Trừ hai số nguyên dương**

* Các bước thực hiện tương tự trên, thay vì phép cộng thì ta chọn phép trừ

A screenshot of a computer

Description automatically generated

*Phép trừ hai số nguyên dương nhập từ bàn phím*

A screenshot of a computer

Description automatically generated

*Phép trừ hai số nguyên dương đọc từ file*

**Test case 09: Trừ hai số nguyên âm**

* Các bước thực hiện tương tự trên, thay vì phép cộng thì ta chọn phép trừ

A screenshot of a computer

Description automatically generated

*Phép trừ hai số nguyên âm nhập từ bàn phím*

A black screen with white lines

Description automatically generated

*Phép trừ hai số nguyên âm đọc từ file*

**Test case 010: Trừ hai số nguyên trái dấu**

* Các bước thực hiện tương tự trên, thay vì phép cộng thì ta chọn phép trừ

A screenshot of a computer

Description automatically generated

*Phép trừ hai số nguyên trái dấu nhập từ bàn phím*

A computer screen shot of a black screen

Description automatically generated

*Phép trừ hai số nguyên trái dấu đọc từ file*

**Test case 011: Nhân hai số nguyên dương**

* Các bước thực hiện tương tự trên.A computer screen shot of a black screen

  Description automatically generated

*Phép nhân hai số nguyên dương nhập từ bàn phím*

A screenshot of a computer screen

Description automatically generated

*Phép nhân hai số nguyên dương đọc từ file*

**Test case 012: Nhân hai số nguyên âm**

* Các bước thực hiện tương tự trên.

A screenshot of a computer

Description automatically generated

*Phép nhân hai số nguyên âm nhập từ bàn phím*

A screen shot of a computer

Description automatically generated

*Phép nhân hai số nguyên âm nhập từ file*

**Test case 013: Nhân hai số nguyên trái dấu**

* Các bước thực hiện tương tự trên.

A computer screen shot of a black screen

Description automatically generated

*Phép nhân hai số nguyên trái dấu nhập bàn phím*

A screenshot of a computer screen

Description automatically generated

*Phép nhân hai số nguyên trái dấu nhập từ file*

**Test case 014: Chia hai số nguyên dương**

* Các bước thực hiện tương tự trên.

A screenshot of a computer

Description automatically generated

*Phép chia hai số nguyên dương nhập từ bàn ph*

A screenshot of a computer screen

Description automatically generated

*Phép chia hai số nguyên dương nhập từ file*

**Test case 015: Chia hai số nguyên âm**

* Các bước thực hiện tương tự trên.

A screenshot of a computer

Description automatically generated

*Phép chia hai số nguyên âm nhập từ bàn phím*

A screenshot of a computer

Description automatically generated

*Phép chia hai số nguyên âm nhập từ file*

**Test case 016: Chia hai số nguyên trái dấu**

* Các bước thực hiện tương tự trên.

A screenshot of a computer

Description automatically generated

*Phép chia hai số nguyên trái dấu nhập từ bàn phím*

A black screen with white lines

Description automatically generated

*Phép chia hai số nguyên trái dấu nhập từ file*

**Test case 017: Chia một số cho 0;**

* Input: Một số nguyên khác 0 và một số nguyên bằng 0
* Output: Thông báo lỗi rồi dừng lại chương trình.

*A screenshot of a computer

Description automatically generated  
 Minh họa một số chia cho 0*

**Test case 018: In kết quả ra file**

* Sau khi thực hiện các phép toán thì người dùng có quyền lựa chọn thực hiện in kết quả ra file hay không
* Input: Nhập “Y” nếu muốn in hoặc “N” nếu không muốn in
* Output: Nếu muốn in thì nhập đường dẫn đến file, nếu không thì chương trình kết thúc

A screenshot of a computer

Description automatically generated

*Minh họa nếu lựa chọn in kết quả*

*A screenshot of a computer

Description automatically generated*

*Minh họa nếu lựa chọn không in kết quả*

**Test case 019: Lựa chọn in kết quả ra file**

* Sau khi lựa chọn nhập file thì chương trình yêu cầu nhập đường dẫn
* Input: Đường dẫn đến file (.txt) cần lưu kết quả
* Output: Thông báo thành công và kết quả được ghi trong file

A screenshot of a computer

Description automatically generated

*Màn hình hiển thị thông báo thành công*

A screenshot of a computer

Description automatically generated

*Kết quả được lưu vào file Ket\_Qua.txt*

**Test case 020: Nhập đường dẫn lưu kết quả không tồn tại hoặc sai**

* Sau khi lựa chọn nhập file thì chương trình yêu cầu nhập đường dẫn
* Input: Nhập một đường dẫn sai hoặc không tồn tại
* Output: Thông báo lỗi và kết thúc chương trình
* Mô tả kiểm thử:
* Đầu vào nhập một đường dẫn không dẫn đến file (.txt) nào cả
* Chương trình chạy không đọc được file nên báo lỗi

A screenshot of a computer

Description automatically generated

*Nhập sai đường dẫn và chương trình thông báo lỗi*

Tài liệu tham khảo

1. Vũ Thành Nam. “Bài giảng Kỹ thuật lập trình”, Khoa Toán – Tin, Đại học Bách Khoa Hà Nội ( 2024)
2. <https://nguyenvanhieu.vn/cong-tru-nhan-chia-2-so-nguyen-lon/>
3. Bjarne Stroustrup. "Programming: Principles and Practice Using C++" , 2nd Edition (2014).
4. Chat GPT.
5. Maurizio Gabbrielli, Simone Martini. "Programming Languages: Principles and Paradigms"
6. (2010)