

BÁO CÁO ĐỒ ÁN 2

1. Thông tin sinh viên thực hiện

STT	MSSV	Họ và tên	Email
1	23120023	Nguyễn Thái Bảo	23120023@student.hcmus.edu.vn ryantbnguyen27425@gmail.com

2. Đánh giá tổng quan đồ án

Điểm đề nghị: 10 / 10.

3. Liệt kê các chức năng đã thực hiện được

Số thứ tự	Tính năng	Mô tả chi tiết	Tự đánh giá mức độ hoàn thành (ví dụ như 80%)
1	Xử lý được biểu thức cộng trừ số nguyên bình thường	Hàm "infixToPostfix" có nhiệm vụ xử lý biểu thức trung tố (là các biểu thức có sẵn trong tập tin "input.txt") thành dạng hậu tố, sau đó hàm "calculatePostfix" sẽ tiến hành tính toán và trả về kết quả có kiểu "BigInt" – là cấu trúc dữ liệu tự định nghĩa để xử lý số nguyên lớn trong đồ án này.	100%

		<p>Kiểu các số nguyên bình thường cũng là một số nguyên, nên có thể dùng chung kiểu "BigInt" để xử lý và vẫn đảm bảo cho ra kết quả chính xác. Các thao tác cộng trừ số nguyên bình thường cũng dùng các phép cộng, trừ được định nghĩa để áp dụng cho kiểu "BigInt".</p> <p>Một số ghi chú cho định dạng tập tin "input.txt" và khả năng xử lý tập tin của chương trình:</p> <ol style="list-style-type: none"> 1. Các số nguyên âm có thể có dấu trừ nằm sát với số hoặc được ngăn cách với số bằng 1 khoảng trắng. Tuy nhiên, biểu thức được khuyến khích ghi theo cách 1, tức là "-5", "-(4 + 3)" để dễ phân biệt với phép trừ. Có hỗ trợ đặt số nguyên âm trong cặp dấu ngoặc đơn, ví dụ: "(-3)", "(- 3)". 2. Giữa các số và giữa các số với các dấu liên kề cần được ngăn cách bằng chính xác 1 khoảng trắng, các dấu ngoặc đơn (nếu có) cần ghi sát với số gần nhất. Ví dụ: 24 - 9 - (129 + 90 - 178), -(-5 * -3). Riêng trường hợp số âm khuyến khích ghi sát lại, ví dụ: "-5". 3. Có thể dùng dấu ":" hay dấu "/" để biểu diễn phép chia trong biểu thức. 4. Biểu thức hậu tố sẽ được in ra terminal khi chạy chương trình để tiện theo dõi và kiểm tra. 	
2	Xử lý được các biểu thức cộng trừ các số nguyên Rất Lớn	<p>Hàm "infixToPostfix" và "calculatePostfix" có nhiệm vụ xử lý các biểu thức như đã đề cập ở trên.</p> <p>Ở đây, các số nguyên rất lớn cũng được khai báo với kiểu "BigInt" được định nghĩa để xử lý các số nguyên rất lớn này, và các phép cộng, trừ được thực hiện qua</p>	100%

		<p>các toán tử cộng, trừ được định nghĩa để áp dụng cho kiểu "BigInt".</p> <p>Thuật toán cho phép cộng là làm việc với trường hợp cơ bản nhất là cộng hai số dương theo cách tính thông thường là cộng các chữ số từ phải qua trái có nhớ, từ trường hợp này suy ra cách giải quyết cho các trường hợp khác như cộng hai số âm, cộng số dương và số âm, ...</p> <p>Thuật toán cho phép trừ là làm việc với trường hợp cơ bản nhất: trừ hai số dương lớn cho số dương nhỏ theo cách tính thông thường là trừ các chữ số từ phải qua trái, nếu số bị trừ nhỏ hơn số trừ thì mượn 10 và nợ 1, từ trường hợp này suy ra cách giải quyết cho các trường hợp khác như trừ hai số âm, số dương trừ số âm, ...</p>	
3	Xử lý được các biểu thức có chứa phép nhân các số nguyên Rất Lớn	<p>Hàm "infixToPostfix" và "calculatePostfix" có nhiệm vụ xử lý các biểu thức như đã đề cập ở trên.</p> <p>Các số nguyên rất lớn được khai báo theo kiểu "BigInt".</p> <p>Ý tưởng cài đặt phép nhân trong mã nguồn là: Nhân số nguyên đầu tiên với từng chữ số của số nguyên còn lại từ phải qua trái, sau đó cộng theo phương pháp sắp hàng, tức là kết quả của số nguyên đầu tiên với từng chữ số tiếp theo sẽ nhân 10 rồi cộng kết quả cũ, tạo thành một vòng lặp, các phép nhân 1 số bên trong toán tử sẽ được thực hiện bằng vòng lặp phép cộng đã định nghĩa trước đó, sau cùng kiểm tra dấu của hai số để kết luận dấu của kết quả.</p>	100%
4	Xử lý được các	Hàm "infixToPostfix" và "calculatePostfix"	100%

	<p>biểu thức có chứa phép chia NGUYÊN các số nguyên Rất Lớn</p>	<p>có nhiệm vụ xử lý các biểu thức như đã đề cập ở trên.</p> <p>Các số nguyên rất lớn được khai báo theo kiểu "BigInt".</p> <p>Ý tưởng cài đặt phép chia trong mã nguồn là: đầu tiên đếm số chữ số trong số chia để tách ra phần đầu tiên từ trái sang phải của số bị chia có cùng số chữ số, nếu phần này bé hơn số chia thì ta lấy thêm một chữ số phía sau. Sau đó, tiến hành vòng lặp: gán thương là số đầu tiên chạy từ 9 về 0 mà khi nhân với số chia cho kết quả bé hơn hoặc bằng phần đầu tiên đã tách bên trên. Sau đó tính phần dư, và gán thương đã tính vào bên phải cùng của biến kết quả, sau đó cập nhật lại phần tách = phần dư nhân 10 + chữ số tiếp theo và thực hiện một vòng lặp mới. Sau cùng ta kiểm tra dấu của hai số nguyên để kết luận được dấu của kết quả.</p>	
5	<p>Không sử dụng các thư viện xử lý số lớn</p>	<p>Mã nguồn không sử dụng thư viện xử lý số lớn nào.</p>	100%
6	<p>Về việc sử dụng các thư viện của C++</p> <ul style="list-style-type: none"> - Nếu không sử dụng thì sẽ được cộng 1 điểm (tối đa vẫn là thang 10) - Nếu có sử dụng thì ghi rõ đã sử dụng những thư viện nào 	<p>Các thư viện của C++ đã sử dụng là:</p> <ol style="list-style-type: none"> 1. <stdio.h> 2. <iostream> 3. <string.h> 	

4. Mô tả cấu trúc dữ liệu cho số nguyên Rất Lớn

Chương trình xây dựng cấu trúc dữ liệu struct BigInt để lưu trữ và làm việc với số nguyên Rất Lớn.

Struct BigInt này tương tự như một danh sách liên kết với mỗi node là một chữ số của số nguyên Rất Lớn, lưu trữ theo đúng thứ tự bình thường từ trái sang phải.

Do đó, bên trong cấu trúc sẽ có hai con trỏ Node kiểu int (do mỗi chữ số đương nhiên là một số nguyên) là "head" và "tail" để dễ dàng quản lý và thao tác, đồng thời có thêm biến thành viên kiểu Boolean là "sign" để lưu trữ dấu của số nguyên Rất Lớn, qui ước như sau: "true" cho số dương và "false" cho số âm.

Việc sử dụng kiểu BigInt như một danh sách liên kết giúp chèn cách phần tử dễ dàng, dễ thao tác, thuận tiện lưu trữ các số nguyên Rất Lớn (hàng ngàn chữ số) vì mỗi Node chỉ là một chữ số nên không lo bị tràn số, đồng thời sau khi không sử dụng nữa có thể dễ dàng giải phóng bộ nhớ của các Node.