

ĐẠI HỌC QUỐC GIA TP. HCM
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN



BÁO CÁO ĐỒ ÁN

POLISH NOTATION

MIPS Architecture and Assembly Language

Danh sách thành viên:

Mã số sinh viên	Tên	Phân công
1712352	Chu Nguyên Đức	Làm hậu tố và kết quả.
1712357	Nguyễn Huỳnh Đức	Làm hậu tố.
1712405	Nguyễn Trường Giang	Làm báo cáo.

Môi trường lập trình:

MARS 4.0.1

Ý tưởng thiết kế và thực hiện đồ án:

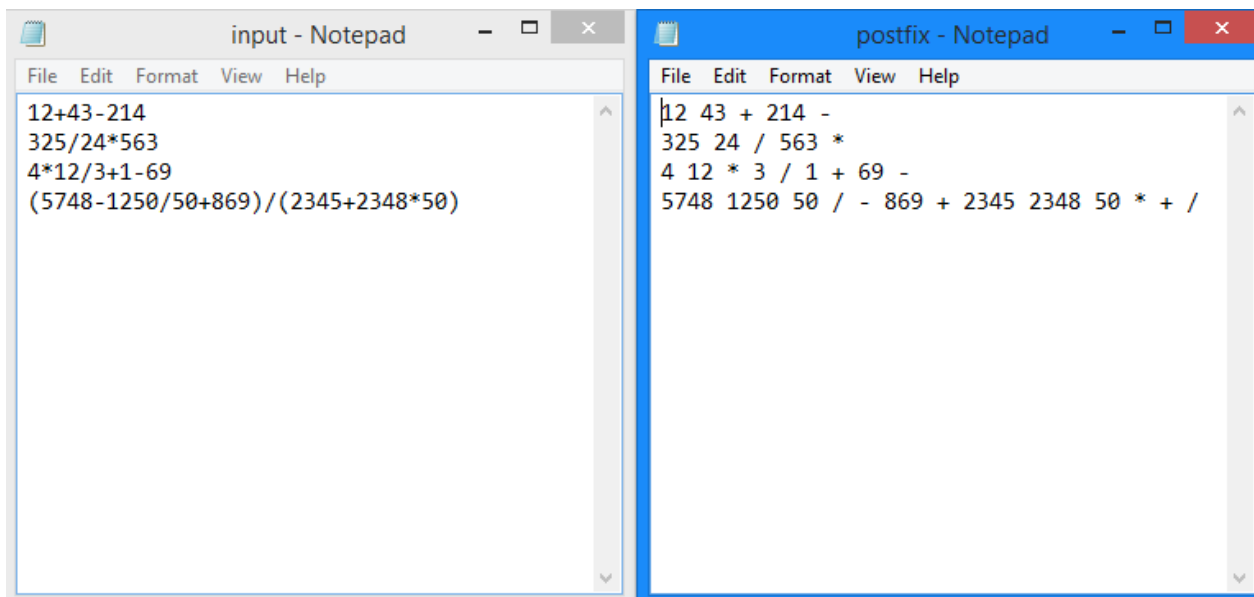
Đọc file, lưu vào buffer. Sau đó đọc từng ký tự của buffer, tiến hành chuyển từ tiền tố sang hậu tố. Gặp ký tự xuống dòng thì tiến hành tính toán, sau đó xuất output ra file. Rồi đọc ký tự tiếp theo cho đến khi hết buffer.

Các chức năng:

1) Chuyển đổi từ dạng trung tố sang hậu tố

Áp dụng thuật toán được giáo viên gợi ý.

- ✓ `hienThiStackDenKhiGapMoNgoac`: Pop và hiển thị các phần tử trong stack cho đến khi gặp ngoặc trái, xong rồi pop nhưng không hiển thị.
- ✓ `Push_Output`: Hiển thị toán hạng từ chuỗi input
- ✓ `KtraKyTu`: Kiểm tra ký tự đó thuộc toán hạng, toán tử hay là mở đóng ngoặc.
- ✓ `SoSanh_ToanTu`: Kiểm tra độ ưu tiên của toán tử để quyết định việc pop hay push stack.
- ✓ `pushStack`: Push toán tử vào stack
- ✓ `popStack`: Pop toán tử khỏi stack
- ✓ `ThemTopStackvaoChuoiOutput`: Thêm toán tử ở đỉnh stack vào chuỗi output



Dòng 1: Biểu thức có phép cộng, trừ.

Dòng 2: Biểu thức có phép nhân, chia.

Dòng 3: Biểu thức có phép cộng, trừ, nhân và chia.

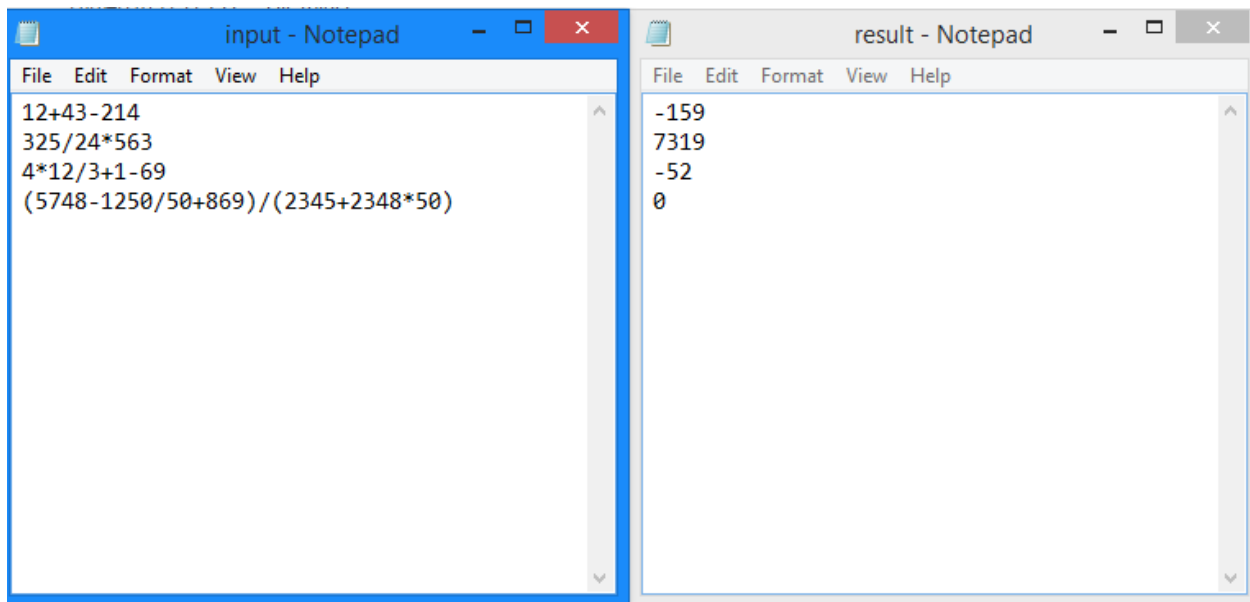
Dòng 4: Biểu thức có dấu ngoặc với các phép cộng, trừ, nhân, chia.

2) Tính kết quả dựa vào biểu thức hậu tố

Áp dụng thuật toán được giáo viên gợi ý. Sau khi ra được kết quả thì chuyển số đó thành chuỗi.

- ✓ StartCalculate: Bắt đầu thực hiện tính kết quả
- ✓ getNumberFromString: Lấy toán hạng từ biểu thức
- ✓ tinhToan2So: Thực hiện tính toán với 2 toán hạng ở đỉnh stack
- ✓ pushNumToStack: Sau khi thực hiện tính toán với 2 toán hạng thì push kết quả vào stack
- ✓ popStack_getNumber: pop stack để lấy toán hạng
- ✓ tinhCong, tinhTru, tinhNhan, tinhChia: Thực hiện các phép tính

- ✓ ContinueCalculate: Xét ký tự tiếp theo trong biểu thức hậu tố
- ✓ EndCalculate: Kết thúc tính kết quả
- ✓ Itoa: Thực hiện chuyển kết quả thành chuỗi



Dòng 1: Biểu thức có phép cộng, trừ.

Dòng 2: Biểu thức có phép nhân, chia.

Dòng 3: Biểu thức có phép cộng, trừ, nhân và chia.

Dòng 4: Biểu thức có dấu ngoặc với các phép cộng, trừ, nhân, chia.

Các chức năng đã làm được: Chuyển biểu thức trung tố thành hậu tố và tính kết quả

Mức độ hoàn thành: 70%

Nguồn tài liệu tham khảo:

- <https://github.com/MIPT-ILab/mipt-mips/wiki/MIPS-pseudo-instructions>
- <http://courses.missouristate.edu/kenvollmar/mars/help/syscallhelp.html>
- <https://stackoverflow.com/>
- <https://www.geeksforgeeks.org/>
- Slide bài giảng Kiến trúc máy tính & Hợp ngữ của cô Chung Thuý Linh, google, wiki.