**ĐẠI HỌC KHOA HỌC TỰ NHIÊN**

**KHOA CÔNG NGHỆ THÔNG TIN**

---------------🙦 🕮 🙤---------------

**đồ án POLISH NOTATION**

**kiến trúc máy tính và hợp ngữ**



**⬩ SINH VIÊN THỰC HIỆN ⬩**

Nguyễn Hữu Gia Trí - 1712254

Huỳnh Thái Anh - 1712272

Đỗ Quang Vinh - 1712207

**MỤC LỤC**

[Ý TƯỞNG THỰC HIỆN VÀ MÔI TRƯỜNG LẬP TRÌNH 3](#_Toc5223638)

[THÔNG TIN THÀNH VIÊN VÀ PHÂN CÔNG 6](#_Toc5223639)

[KIỂM THỬ CHƯƠNG TRÌNH 7](#_Toc5223640)

[MỨC ĐỘ HOÀN THIỆN 8](#_Toc5223641)

[TÀI LIỆU THAM KHẢO 8](#_Toc5223642)

# Ý TƯỞNG THỰC HIỆN VÀ MÔI TRƯỜNG LẬP TRÌNH

1. **Ý tưởng thực hiện**
   1. **Các bước giải quyết bài toán**

**Bước 1:** Đọc từ file input ra buffer, từ buffer tách thành từng dòng.

**Bước 2:** Phân tích mỗi dòng thành 2 mảng: 1 mảng data[] chứa các toán tử và toán hạng và 1 mảng isOP[] có các phần tử mang giá trị 1 nếu vị trí tương trong data[] là toán từ, mang giá trị 0 nếu vị trí tương trong data[] là toán hạng.

**Bước 3:** Từ kết quả phân tích ở bước 2 tiến hành đọc và chuyển sang dạng hậu tố (postfix) và tiền tố (prefix).

**Bước 4:** Ghi kết quả ra file.

* 1. **Thuật toán sử dụng**
* **Thuật toán chuyển từ dạng trung tố sang hậu tố:**

❖ Khởi động stack rỗng (Stack chứa toán tử)  
❖ While (không có lỗi và chưa hết biểu thức)  
o Đọc Token (Token = hằng/biến/toán tử số học /ngoặc trái/  
o Nếu Token là  
➢ Ngoặc trái:Push vào stack.  
➢ Ngoặc phải: Pop và hiển thị các phần tử của stack đến khi gặp ngoặc trái (pop ngoặc trái nhưng không hiển thị ngoặc trái).

➢ Toán tử : Nếu stack rỗng hay Token được ưu tiên hơn phần tử ở đỉnh stack thì Push vào Stack .Ngược lại (ưu tiên bằng hoặc ít ưu tiên hơn) pop và hiển thị 1 phần tử ở đỉnh stack .Lặp lại việc so sánh Token với 1 phần tử ở đỉnh stack.

➢ Toán hạng : hiển thị nó.  
❖ Khi hết biểu thức trung tố Pop và hiển thị toàn bộ stack còn lại.

* **Thuật toán chuyển từ dạng trung tố sang tiền tố:**

Bước 1 : Đảo ngược biểu thức trung tố . Lưu ý : trong khi đảo ngược thì dấu ngoặc trái ‘(‘ sẽ thành ngoặc phải ‘)’ và ngược lại .

Bước 2 : Sử dụng lại thuật toán trung tố sang hậu tố ( với biểu thức trung tố là biểu thức ở bước 1 ) và tìm được biểu thức hậu tố .

Bước 3 : Đảo ngược lại biểu thức hậu tố tìm được ta sẽ có biểu thức tiền tố.

* **Thuật toán tính giá trị biểu thức hậu tố:**

❖ Khởi động stack rỗng.  
❖ Lặp lại các bước sau đến khi hết biểu thức:  
o Đọc Token (Hằng ,biến , toán tử)  
o Nếu Token là : 4\*12/3+1-> 4 12 \* 3 / 1  
➢ Toán hạng: Push vào stack  
➢ Toán tử:  
• Pop 2 giá trị  
• Áp dụng toán tử cho 2 giá trị lấy ra.  
• Push kết quả vào stack.  
Lặp đến hết biểu thức, giá trị ở đỉnh stack là giá trị của biểu thức.

* **Độ ưu tiên của các toán tử:** ‘(’ < {‘+’,’-‘} < {‘\*’,’/’}

1. **Môi trường lập trình**
   1. **Ngôn ngữ lập trình:** MIPS
   2. **IDE:** Mars 4.5

# THÔNG TIN THÀNH VIÊN VÀ PHÂN CÔNG

1. **Thông tin thành viên**

|  |  |  |
| --- | --- | --- |
| **Tên** | **MSSV** | **Email** |
| **Nguyễn Hữu Gia Trí** | 1712254 | 1712254@student.hcmus.edu.vn |
| **Huỳnh Thái Anh** | 1712272 | 1712272@student.hcmus.edu.vn |
| **Đỗ Quang Vinh** | 1712207 | 1712207@student.hcmus.edu.vn |

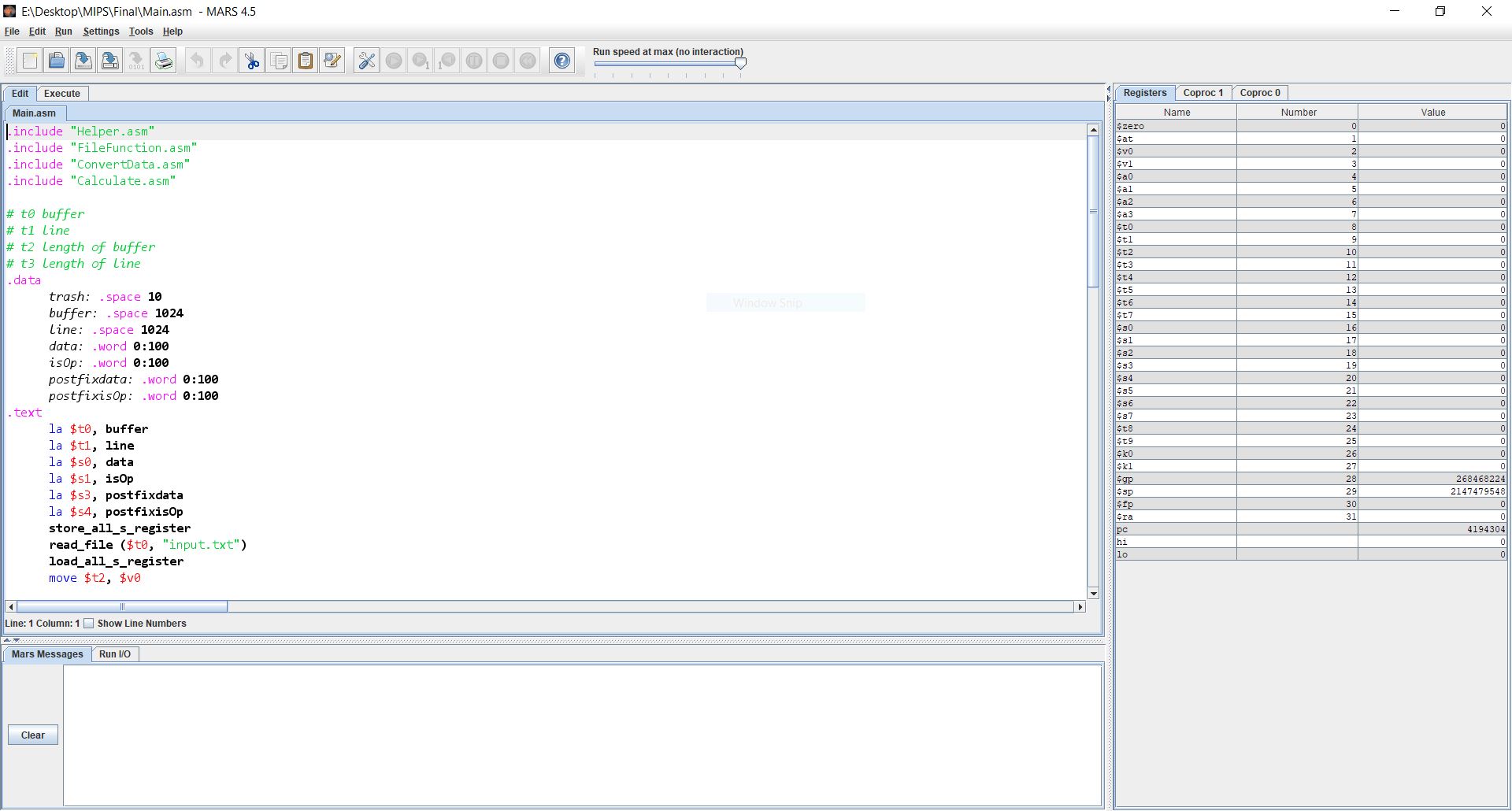
1. **Phân công**

|  |  |  |
| --- | --- | --- |
| **Tên** | **Công việc** | **Mức độ hoàn thành** |
| **Nguyễn Hữu Gia Trí** | * Tổ chức chương trình * Xử lý đọc xuất file * Tính giá trị biểu thức * Viết báo cáo | 100% |
| **Huỳnh Thái Anh** | * Chuyển biểu thức sang tiền tố * Hỗ trợ viết báo cáo | 100% |
| **Đỗ Quang Vinh** | * Chuyển biểu thức sang hậu tố * Hỗ trợ viết báo cáo | 100% |

# KIỂM THỬ CHƯƠNG TRÌNH

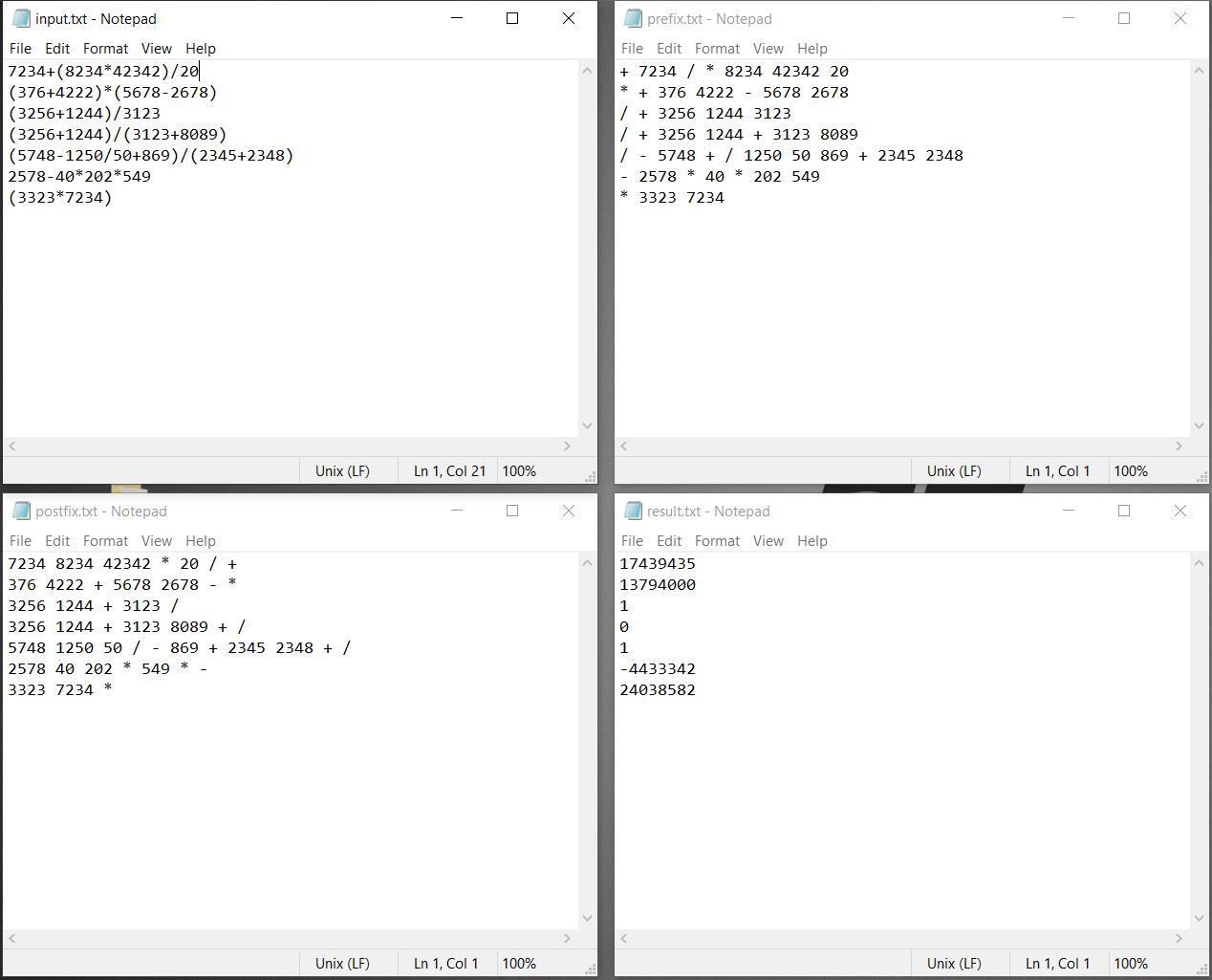
1. **Cách thực thi chương trình**

* Mở phần mềm Mars trong thư mục chứa source code.
* Chạy file Main.asm
* Các file kết quả sẽ xuất hiện trong thư mục chứa source code



1. **Đọc kết quả**

* Mỗi dòng tương ứng của tập tin Output là kết quả của phép tính toán hoặc chuyển đổi.



# MỨC ĐỘ HOÀN THIỆN

* ***Hoàn thành 100% yêu cầu của đồ án***

# TÀI LIỆU THAM KHẢO

* File hướng dẫn đồ án của giáo viên

*https://courses.fit.hcmus.edu.vn/mod/resource/view.php?id=62125*

* Github

[*https://github.com/HuyJokezy/infix-to-postfix-mips*](https://github.com/HuyJokezy/infix-to-postfix-mips)

* Youtube

<https://www.youtube.com/watch?v=L6_bDTy8g-U>