**MỤC LỤC**

[I. PHẦN MỞ ĐẦU 2](#_Toc20772)

[1. Lí do chọn đề tài 2](#_Toc31729)

[2. Mục tiêu cho đề tài 2](#_Toc22362)

[3. Phương pháp nghiên cứu 2](#_Toc13028)

[4. Giới thiệu đề tài 2](#_Toc4967)

[II. PHẦN NỘI DUNG 4](#_Toc4)

[1. Mô tả ứng dụng và Chức năng của phần mềm 4](#_Toc9547)

[2. Mô tả quá trình thực hiện 4](#_Toc8514)

[2.1. Thiết kế giao diện 4](#_Toc19368)

[2.2. Ứng dụng tính toán giá trị phép tính bằng trung tố sang hậu tố 5](#_Toc3208)

[2.3. Viết code phần xử lí 6](#_Toc30945)

[2.4. Chạy và kiểm thử 11](#_Toc6489)

[2.5. Sửa các lỗi gặp phải 13](#_Toc28697)

[III. KẾT LUẬN 14](#_Toc18766)

[IV. TÀI LIỆU THAM KHẢO 15](#_Toc21375)

**MỤC LỤC HÌNH**

[Hình 1: Máy tính Calculator trong thực tế. 3](#_Toc531329127)

[Hình 2: Giao diện cơ bản sau khi thiết kế. 5](#_Toc531329128)

[Hình 3: Hàm xử lý nút thêm số và một số kí tự. 7](#_Toc531329129)

[Hình 4: Hàm xử lí Nút “←”(Xóa một kí tự bên phải TextBox).. 7](#_Toc531329130)

[Hình 5: Chạy thử thuật toán. 9](#_Toc531329131)

[Hình 6: Hàm xét độ ưu tiên của phép tính (GetPriority). 9](#_Toc531329132)

[Hình 7: Hàm thêm ký tự số vào sh (Stack hạng) 1](#_Toc531329133)0

[Hình 8: Một phần của hàm thêm các dấu vào st (Stack tử)...............................................1](#_Toc531329134)0

[Hình 9: Mở chương trình. 1](#_Toc531329135)1

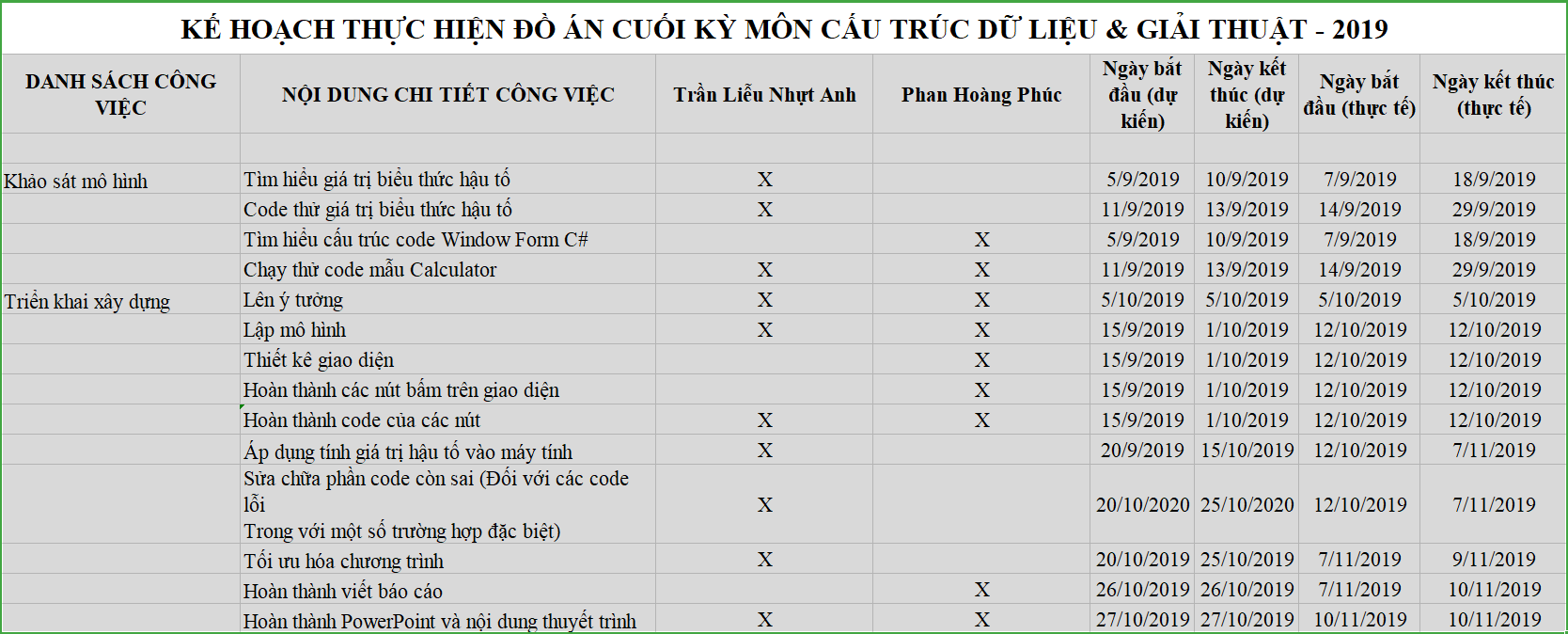
[Hình 10: Nhập phép tính đơn giản 1+1 1](#_Toc531329136)1

Hình 11: Kết quả, biểu diễn trung tố, hậu tố của 1+1........................................................12

Hình 12: Nhập biểu thức 5\*3 - (3+5).................................................................................12

Hình 13: Kết quả, biểu thức trung tố, hậu tố của biểu thức 5\*3 - (3+5)............................13

Hình 14: Hàm xử lí biến NhapLai.....................................................................................13



LỜI NÓI ĐẦU

Cấu trúc dữ liệu và giải thuật là một trong những môn học nền tảng quan trọng nhất dành cho sinh viên ngành Công nghệ Thông tin. Môn học cung cấp cho sinh viên hiểu và biết cách sử dụng các kiểu dữ liệu cơ bản thường gặp trong quá học tập và làm việc. Giúp người học biết được sự tác động lẫn nhau giữa dữ liệu và giải thuật để người sử dụng có thể giải quyết các vấn đề gặp phải trong bài toán và thực tế.

Sau quá trình học tác giả đã vận dụng kiến thức tiếp thu được để ứng dụng thực hiện đồ án môn học với đề tài: Ứng dụng tính giá trị hậu tố tạo máy tính Calculator.

# PHẦN MỞ ĐẦU

## Lí do chọn đề tài

Với đề tài “Ứng dụng tính giá trị hậu tố tạo máy tính Calculator” có thể giúp tác giả vận dụng những kiến thức học được từ môn học, đặc biệt là hiểu rõ hơn về cách tính bằng hậu tố. Ngoài ra việc tạo ứng dụng máy tính Calculator sẽ giúp mọi người có thể tính toán được các phép tính mà không cần mô hình thực tế, phục vụ cho nhu cầu tính toán của mọi người.

## Mục tiêu cho đề tài

Hoàn thành sản phẩm, có thể cho người sử dụng được các thao tác (bấm số và các dấu của phép tính). Giúp bản thân tác giả hiểu được, hiểu sâu hơn về cách tính của ứng dụng hậu tố. Có thể sử dụng thành thạo cách tính của mô hình hậu tố.

## Phương pháp nghiên cứu

Vận dụng các kiến thức đã học được từ bộ môn “Cấu trúc dữ liệu và giải thuật”, tham khảo các nguồn tài liệu, giáo trình được cung cấp. Ngoài ra còn tham khảo thêm các nguồn tài liệu khác trên mạng internet và vận dụng tất cả để thực hiện đề tài nghiên cứu một cách thành công nhất.

## Giới thiệu đề tài

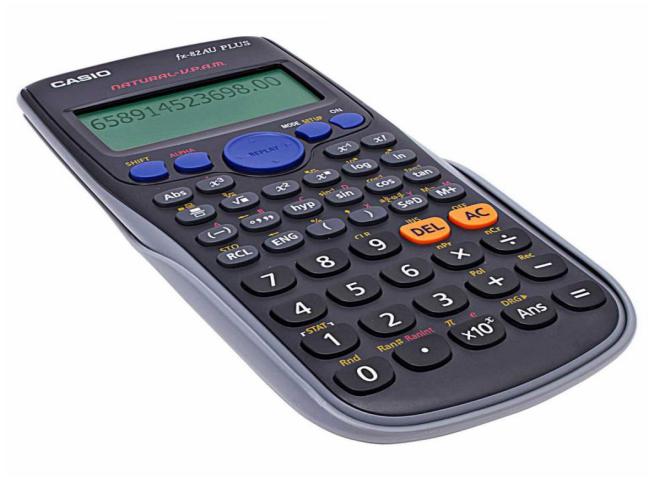
Chiếc máy tính bỏ túi là một đồ vật rất quen thuộc đối với nhiều người, từ sinh viên, học sinh cho đến các nhân viên bán hàng, kế toán v.v… Là một thiết bị giúp bạn thực hiện các quy tắc toán học như cộng trừ, nhân chia, hay cao cấp hơn như giải phương trình, tính toán ma trận. Để có thể trở thành một thiết bị nhỏ gọn mà hữu dụng với nhiều chức năng như hiện nay, chiếc máy tính bỏ túi đã phải trải qua những giai đoạn thăng trầm trong lịch sử mà mãi đến tận cuối thế kỷ 20 nó mới được sử dụng một cách rộng rãi.

Bàn tính cơ học chính là tiền thân của máy tính bỏ túi ngày nay. Công cụ tính toán số học đầu tiên được biết đến là chiếc bàn tính (Abacus) được sử dụng bởi những người Sumer và người Ai Cập vào khoảng 2000 năm trước công nguyên. Sau đó, bàn tính được sử dụng rộng rãi ở các nước Châu Á, châu Phi và nhiều vùng lãnh thổ khác chủ yếu bởi các thương nhân. Cho đến ngày nay, vẫn còn khá nhiều người sử dụng loại bàn tính cơ học của người Trung Quốc, với các hạt được xâu thành chuỗi theo chiều dọc trong một khung gỗ chữ nhật.

Đến năm 1964, Công ty Nhật Bản Sharp đã chế tạo được chiếc máy tính đầu tiên có thể tự thực hiện các phép tính toán, mặc dù chiếc máy tính có kích thước gần bằng một chiếc xe ô tô. Năm 1967, Texas Instruments giới thiệu dự án công nghệ “Cal Tech” với chiếc máy tính bỏ túi đầu tiên có kích thước nhỏ gọn, có khả năng thực hiện các phép tính đơn giản như cộng trừ, nhân chia.

Sau đó vài năm, Canon đã sử dụng công nghệ này để sản xuất những chiếc máy tính bỏ túi thương mại đầu tiên và bán rộng rãi trên thị trường với giá 400 USD. Tuy nhiên chỉ đến năm 1971, sau khi Intel ra mắt mẫu chip xử lý thương mại đầu tiên là Intel 4004, những chiếc máy tính bỏ túi mới thực sự trở nên hữu dụng với khả năng tính toán tốt hơn, kích thước nhỏ hơn cũng như giá thành hợp lý hơn.

Cho đến nay, với sự phát triển của khoa học, công nghệ những chiếc máy tính bỏ túi không chỉ đơn giản là thực hiện các phép tính cộng trừ, nhân chia. Mà bên cạnh đó chúng còn có khả năng thực hiện nhiều phép biến đổi, các hàm lượng giác và logarit, làm việc với các hằng số như pi và e, tính toán với số phức hay phân số, giải phương trình, phân tích thống kê, sác xuất hay ma trận. [4]



Hình 1: Máy tính Calculator trong thực tế.

# 

# II. PHẦN NỘI DUNG

## Mô tả ứng dụng và Chức năng của phần mềm

Mô phỏng máy tính bỏ túi thành ứng dụng để mọi người có thể tính toán mà không cần có mô hình thực tế. Ứng dụng trong tính toán, trong học tập và giảng dạy và lập trình căn bản.

## Mô tả quá trình thực hiện

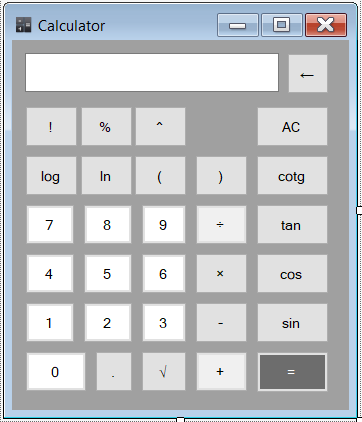
### Thiết kế giao diện

Việc thiết kế giao diện của ứng dụng là một phần rất quan trọng trong việc xây dựng ứng dụng, là bộ mặt của ứng dụng, nơi người dùng trực tiếp thao tác với ứng dụng. Tạo sự hứng thú, thu hút cho người sử dụng.

#### Thiết kế giao diện với Windows Forms Application

Kể từ lúc này chúng ta sẽ bắt đầu sử dụng Windows Forms Application để thiết kế và lập trình ứng dụng.

* Sau khi tạo project thành công, chúng ta bắt đầu điều chỉnh form với các thao tác trên thanh properties. Chúng ta sẽ thay đổi màu nền (*Backcolor*), cố định size tùy chỉnh (*MaximizeBoxTrue*), chỉnh lại mục *Name* và mục *Text*.
* Tiến hành tinh chỉnh vị trí, chỉnh thanh *TextBox* lên trên cùng của *Form* sao cho thuận mắt và dễ tương tác với người sử dụng.
* Tạo 30 nút (*Button*) Các con số: “1”, “2”, “3”, “4”, “5”, “6”, “7”, “8”, “9”, “0”, Các dấu phục vụ cho các phép tính: “+”, “-”, “x”, “/”, “√”, “(”, “)”, “%”, “^”, “ln”, “log”, “!”, “sin”, “cos”, “tan”, “cot”, “=”, “.”, dấu xóa kí tự mới vừa nhập “←”, và dấu xóa tất cả các kí tự “AC”, điều chỉnh vị trí phù hợp so với form.



Hình 2: Giao diện cơ bản sau khi thiết kế.

### Ứng dụng tính toán giá trị phép tính bằng trung tố sang hậu tố

#### Định nghĩa và cách tính của ký pháp nghịch đảo Balan

Cách trình bày biểu thức theo cách thông thường tuy tự nhiên với con người nhưng lại khá “khó chịu” đối với máy tính vì nó không thể hiện một cách tường minh quá trình tính toán để đưa ra giá trị của biểu thức. Để đơn giản hóa quá trình tính toán này, ta phải biến đổi lại biểu thức thông thường về dạng hậu tố – postfix (cách gọi ngắn của thuật ngữ ký pháp nghịch đảo Ba Lan). Để phân biệt hai dạng biểu diễn biểu thức, ta gọi cách biểu diễn biểu thức theo cách thông thường là trung tố – infix (vì toán tử nằm ở giữa hai toán hạng).

Ký pháp nghịch đảo Ba Lan được phát minh vào khoảng giữa thập kỷ 1950 bởi Charles Hamblin – một triết học gia và khoa học gia máy tính người Úc – dựa theo công trình về ký pháp Ba Lan của nhà Toán học người Ba Lan Jan Łukasiewicz. Hamblin trình bày nghiên cứu của mình tại một hội nghị khoa học vào tháng 6 năm 1957 và chính thức công bố vào năm 1962.

Từ cái tên hậu tố ta cũng đoán ra phần nào là theo cách biểu diễn này, các toán tử sẽ được đặt sau các toán hạng. Cụ thể là biểu thức trung tố: 4+5 sẽ được biểu diễn thành 4 5 +.

Quá trình tính toán giá trị của biểu thức hậu tố khá tự nhiên đối với máy tính. Ý tưởng là đọc biểu thức từ trái sang phải, nếu gặp một toán hạng (con số hoặc biến) thì push toán hạng này vào ngăn xếp; nếu gặp toán tử, lấy hai toán hạng ra khỏi ngăn xếp (stack), tính kết quả, đẩy kết quả trở lại ngăn xếp. Khi quá trình kết thúc thì con số cuối cùng còn lại trong ngăn xếp chính là giá trị của biểu thức đó. [2]

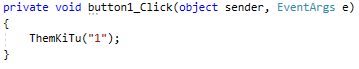
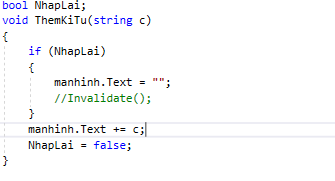
#### Các phương thức được sử dụng liên quan đến trung tố sang hậu tố (Ký pháp nghịch đảo Balan) trong đề tài:

* *Push()*: Thêm phần tử dấu vào Stack.
* *Pop():* Lấy phần tử dấu ra khỏi Stack.[3]
* *Peek()*: Lấy phần tử nhưng không xóa phần tử ấy ra khỏi Stack.

### Viết code phần xử lí

#### Nút số từ 0 đến 9 và các dấu thực hiện phép tính

Khi người chơi click vào nút từ 0 đến 9 lúc đó trên *TextBox* sẽ hiện số vừa nhập, nút các nhút như “√”, “(”, “)”, “ln”, “log”, “sin”, “cos”, “tan”, “cot”, “.” thì *TextBox* sẽ hiển thị những kí tự đại diện cho các nút ấy.

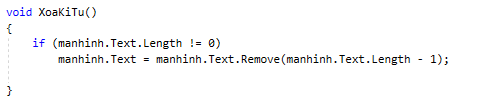


*Hình 3: Hàm xử lí nút thêm số và một số kí tự.*

Nhấn các số từ 0 đến 9 và các ký tự “√”, “(”, “)”, “ln”, “log”, “sin”, “cos”, “tan”, “cot”, “.” Thì các ký tự ấy sẽ được xuất ra *TextBox* đồng thời *NhapLai = false* nghĩa là ứng dụng cho phép ta viết tiếp vào TextBox nội dung của biểu thức.

#### Nút xóa một ký tự vừa nhập “←”.

Với nút “←” ta có hai trường hợp xảy ra, với trường hợp chưa nhập kí tự nào, và trường hợp đã nhập kí tự, với trường hợp này thì một kí tự mà ta vừa nhập sẽ được xóa đi, bao nhiêu lần bấm nút “←” thí sẽ có bấy nhiêu kí tự vừa nhập ở bên phải bị xóa đi.



Hình 4: Hàm xử lí Nút “←”(Xóa một kí tự bên phải TextBox).

* Trường hợp 1: Khi chưa nhập kí tự nào (*manhinh.Text.Length==0*): Màn hình *TextBox* vẫn được giữ nguyên *.*
* Trường hợp 2: Khi *TextBox* đã chứa kí tự (*manhinh.Text.Length!=0*): Kí tự cuối cùng bên phải sẽ biến mất bằng cách dùng lệnh *Remove().* Tức là màn hình hiện tại *manhinh.Text* sẽ đổi cách hiện thị theo dòng lệnh *manhinh.Text.Remove(manhinh.Text.Length -1)* với độ dài *Length* trên *TextBox* được giảm đi 1 đơn vị.

#### Thực hiện phép tính

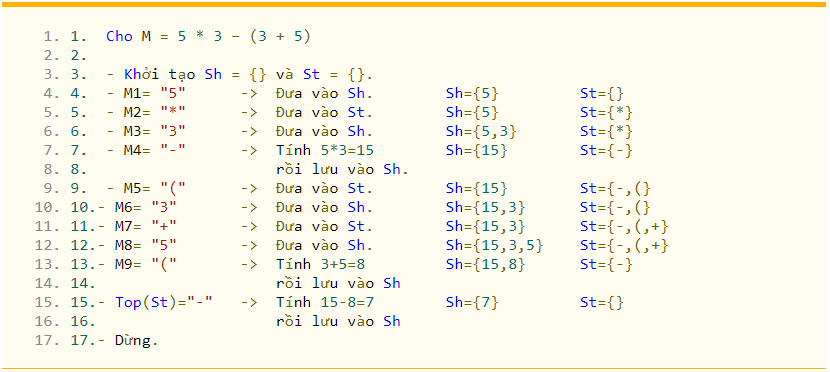
Thuật toán :

Giả sử M là một biểu thức được cho ở dạng trung tố.

Khởi tạo 2 Stack: Sh và St. Stack Sh dùng để lưu trữ toán hạng, stack St dùng để lưu trữ toán tử.

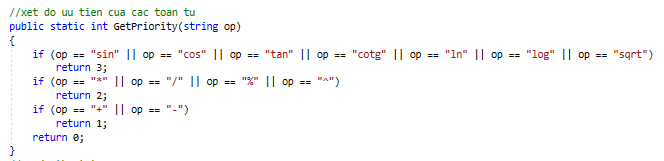
Duyệt M từ trái qua phải:

* Nếu M[i] là toán hạng thì đưa vô Sh.
* Nếu M[i] là “(“ thì đưa vô St.
* Nếu M[i] là toán tử có độ ưu tiên cao hơn toán tử hiện có trên đỉnh St thì đưa nó vào St.
* Nếu M[i] là toán tử có độ ưu tiên thấp hơn hoặc bằng toán tử hiện có trên đỉnh St thì lấy 2 toán tử trên đỉnh Sh thực hiện phép tính với toán tử là phần tử trên đỉnh của St, kết quả cất vào Sh. Sau đó cất M[i] vào St.
* Nếu M[i] là dấu “)” , nếu phần tử trên đỉnh St khác “(“ thì thực hiện phép tính với toán tử là phần tử trên đỉnh của St, kết quả cất vào Sh. Loại dấu ngoặc “(“ gặp phải đầu tiên ra khỏi St.
* Thực hiện đến khi nào St rỗng và Sh còn 1 phần tử duy nhất đó là kết quả.[2]



Hình 5: Chạy thử thuật toán

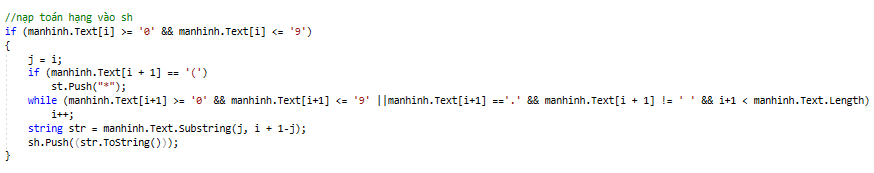
* Xét tính ưu tiên của các toán tử:



*Hình 6: Hàm xét độ ưu tiên của phép tính (GetPriority)*

Các câu lệnh *return 1, return 2, return 3* tương ứng với độ ưu tiên của các phép tính, đối với các dấu như “sin”, “cos”, “tan”, “cotg”, “ln”, “log”, “sqrt” thì sẽ được ưu tiên tính toán trước, sau khi xử lý xong hoặc không tồn tại các dấu trên thì mới tiếp tục xử lý “\*”, “/”, “%”, “^” cuối cùng là các phép tính “+”, “-”.

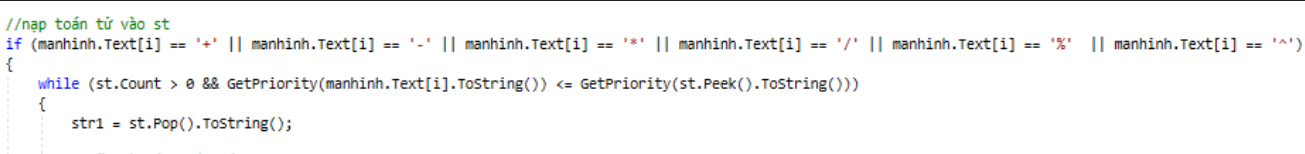
* Đưa toán hạng vào Stack:



*Hình 7: Hàm thêm ký tự số vào sh (Stack hạng)*

Nếu *manhinh.Text[i]* là một số tức là toán hạng vị trí i=0 (vị trí đầu tiên của số được nhập) được gán bằng cột mốc đánh dấu j. Nếu phần tử tiếp theo được nhập vào là ký tự “(” thì dấu được *Push* vào st (Stack tử) là dấu “\*”, khi *manhinh.Text[i+1]* tiếp tục là một con số hoặc dấu “.” và khác dấu “ ” tăng biên chạy i lên một đơn vị, sau khi kí tự tiếp theo nhập vào không phải là số nữa thì *manhinh.Text.Substing* sẽ lưu trữ phần kí tự từ cột mốc đánh dấu j đến kí tự cuối cùng của con số đó i+1-j đồng thời gán phân vừa lưu thành biến lưu trữ *str* , Sau đó, *Push* biến *str* vào sh (Stack hạng).

* Đưa toán tử vào Stack để tính toán:

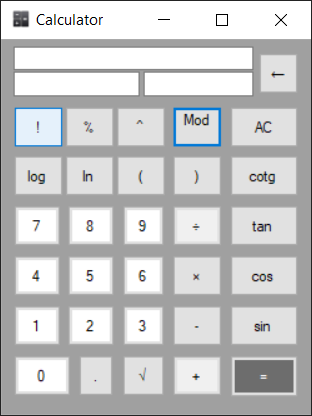


*Hình 8: Một phần của hàm thêm các dấu vào st (Stack tử)*

Nếu *manhinh.Text[i]* là các kí tự dấu thì khi số lượng ký tự có trong st (Stack tử) lớn hơn 0 đồng thời dấu trong st (Stack tử) có độ ưu tiên *GetPriority()* lớn hơn *manhinh.Text[i]* thì sẽ *Peek()* (đưa dấu đầu tiên có sẵn) trong st (Stack tử) ra tính toán sau đó đưa *manhinh.Text[i]* vào st (Stack tử).

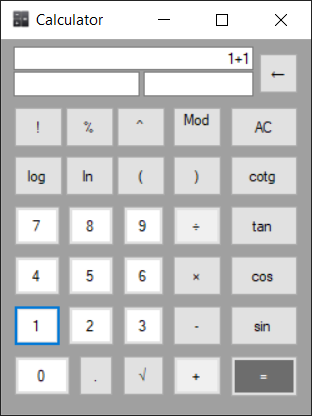
### Chạy và kiểm thử

Mở chạy thử chương trình sau đó tiến hành nhập biểu thức và tính thử.

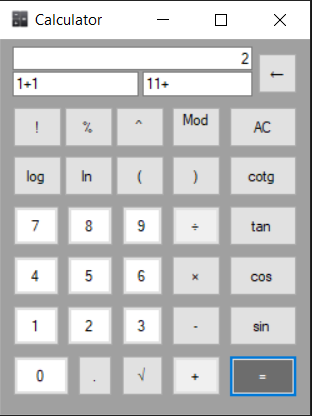


Hình 9: Mở chương trình.

* Tính toán biểu thức đơn giản 1+1:

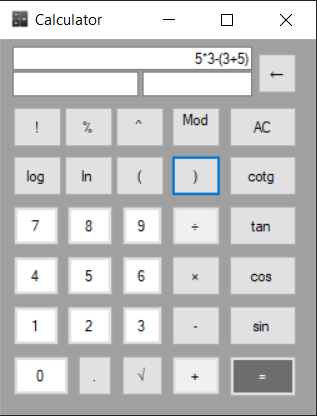


*Hình 10: Nhập phép tính đơn giản 1+1*

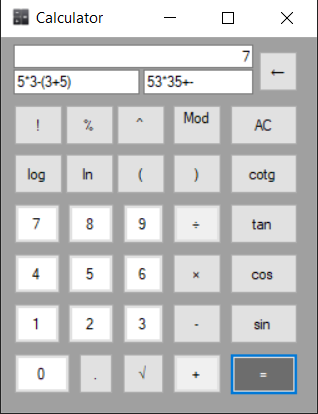


Hình 11: Kết quả, biểu diễn trung tố, hậu tố của 1+1.

* Tính biểu thức 5\*3 - (3+5):



*Hình 12: Nhập biểu thức 5\*3 - (3+5)*

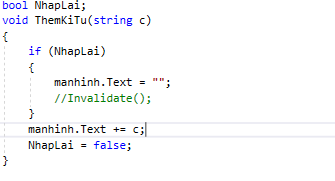


*Hình 13: Kết quả, biểu thức trung tố, hậu tố của biểu thức 5\*3 - (3+5)*

Tuy nhiên trong quá trình chạy thử đã xảy ra lỗi khi nhập một biểu thức mới bắt đầu bằng một con số thì con số ấy ghi tiếp vào bên phải kết quả nên ta đi vào phần sửa lỗi.

### Sửa các lỗi gặp phải

Lỗi đã được sửa bằng cách viết thêm biến *NhapLai* nhằm mục đích phân biệt khi nào kết thúc biểu thức và làm mới *TextBox.*



Hình 14: Hàm xử lí biến NhapLai.

# KẾT LUẬN

* Mức độ hoàn thành mục tiêu: Xây dụng được Calculator dựa trên nguyên lý tính giá trị từ trung tố sang hậu tố bằng Windows Form Applications có thể tính toán trược tiếp các phép tính, hiển thị cách tính và thứ tự ưu tiên của các dấu, đồng thời thực hiện tính toán được một số chức năng khác mà máy tính thực tế có thể làm được.
* Các khó khăn gặp phải: chưa biết cách sử dụng Winform, bị lỗi với xóa đơn vị bên phải của các phần tử, sau khi kết thúc một phép toán nếu ấn phần tử số khác sẽ bị ghi tiếp vào kết quả vừa hiện.
* Cách khắc phục: tìm hiểu và học cách sử dụng Winform từ nhiều nguồn tài liệu trên internet, đặt cờ hiệu.
* Ưu điểm của đồ án: sử dụng, khai thác hết các thuộc tính của tính toán giá trị từ trung tố sang hậu tố (ký pháp nghịch đảo Balan) và cho ra kết quả chính xác, một ít về kiểu dữ liệu Ngăn xếp (Stack), có ứng dụng trong tính toán và cả giáo dục, giao diện dễ nhìn, thân thiện, người sử dụng có thể dễ dàng thực hiện các phép tính.
* Những khuyết điểm của đồ án: so với các sản phẩm tương tự khác bên ngoài thì sản phẩm hiện tại vẫn còn chưa phát triển hoàn thiện nhất về các tính năng (có thể thêm các tính năng mới), giao diện chưa thật sự quá bắt mắt, cầu kì bằng các sản phẩm khác.
* Hướng phát triển đồ án: có thể thêm các tính năng khác như thêm một label hiển thị các kết quả đã tính hoặc nhập từ trước. Thêm các tính năng hữu dụng khác như: đổi từ thập phân sang nhị phân, giải ma trận, … Đầu tư thêm về phần xử lí đồ họa giúp tăng độ thẩm mỹ, bắt mắt thu hút người sử dụng hơn.

# TÀI LIỆU THAM KHẢO

[1]: THUẬT TOÁN TÍNH BIỂU THỨC BẰNG PHƯƠNG PHÁP CHUYỂN ĐỔI BA LAN NGƯỢC <<https://miracleandeffort.wordpress.com/2016/05/07/thuat-toan-tinh-bieu-thuc-bang-phuong-phap-chuyen-doi-ba-lan-nguoc/>>

[2]: ỨNG DỤNG STACK – BIỂU THỨC TRUNG TỐ (Infix) <<https://www.stdio.vn/articles/ung-dung-stack-bieu-thuc-trung-to-infix-470?fbclid=IwAR3527sxj87VJl20LR_B6vg6jgglS3RGXYTMMUsR-KRb_lXlmGP29xXJ-YM>>

[3]: “GIÁO TRÌNH: CẤU TRÚC DỮ LIỆU VÀ GIẢI THUẬT” của tác giả: Lê

Văn Vinh, NXB Đại học Quốc gia TP. Hồ Chí Minh.

[4]: MÁY TÍNH <<https://en.wikipedia.org/wiki/Calculator>>