**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HỒ CHÍ MINH**

**KHOA ĐÀO TẠO CHẤT LƯỢNG CAO**

**NGÀNH CÔNG NGHỆ THÔNG TIN**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_



**BÁO CÁO ĐỒ ÁN**

**CẤU TRÚC DỮ LIỆU & GIẢI THUẬT**

**Đề tài: Chương Trình Tính Biểu Thức Trung Tố Ứng Dụng Stack**

**Giảng viên: Trần Công Tú**

**SVTH : Phan Thanh Hoài**

**MSSV : 16110075**

**SVTH : Lê Trần Ngọc Đạt**

**MSSV : 17110118**

**TP. Hồ Chí Minh, tháng 12 năm 2019**

**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HỒ CHÍ MINH**

**KHOA ĐÀO TẠO CHẤT LƯỢNG CAO**

**NGÀNH CÔNG NGHỆ THÔNG TIN**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_



**BÁO CÁO ĐỒ ÁN**

**CẤU TRÚC DỮ LIỆU & GIẢI THUẬT**

**Đề tài: Chương Trình Tính Biểu Thức Trung Tố Ứng Dụng Stack**

**Giảng viên: Trần Công Tú**

**SVTH : Phan Thanh Hoài**

**MSSV : 16110075**

**SVTH : Lê Trần Ngọc Đạt**

**MSSV : 17110118**

**TP. Hồ Chí Minh, tháng 12 năm 2019**

**LỜI CẢM ƠN**

Xin được gửi lời cảm ơn chân thành đến các giảng viên hướng dẫn – thầy Trần Công Tú vì những lời khuyên, lời chỉ bảo tận tình và bổ ích đã hỗ trợ nhóm em rất nhiều trong quá trình hoàn thành đồ án này.Cảm ơn bạn bè đã tạo điều kiện thuận lợi, đóng góp ý kiến, nhận xét vô cùng bổ ích.

Xin chân thành cảm ơn sự cố gắng, nỗ lực cả tất cả các thành viên trong nhóm.

**NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN**

…………………………………………………………………………………

…………………………………………………………………………………

…………………………………………………………………………………

…………………………………………………………………………………

…………………………………………………………………………………

…………………………………………………………………………………

…………………………………………………………………………………

…………………………………………………………………………………

…………………………………………………………………………………

…………………………………………………………………………………

…………………………………………………………………………………

…………………………………………………………………………………

…………………………………………………………………………………

…………………………………………………………………………………

…………………………………………………………………………………

…………………………………………………………………………………

…………………………………………………………………………………

…………………………………………………………………………………

…………………………………………………………………………………

*Tp. Hồ Chí Minh, ngày …, tháng …, năm 2019*

**Người nhận xét**

**(Ký rõ họ tên)**

**Mục Lục**

[1 TỔNG QUAN NHÓM 7](#_Toc27224566)

[1.1 Thông tin nhóm 7](#_Toc27224567)

[1.2 Phương thức làm việc 7](#_Toc27224568)

[1.2.1 Quy trình 7](#_Toc27224569)

[1.2.2 Kế hoạch thực hiện đồ án 8](#_Toc27224570)

[2 BÁO CÁO ĐỒ ÁN 9](#_Toc27224571)

[2.1 Giới thiệu chung 9](#_Toc27224572)

[2.1.1 Tên đồ án 9](#_Toc27224573)

[2.1.2 Lý do chọn đề tài 9](#_Toc27224574)

[2.1.3 Các chức năng của chương trình 9](#_Toc27224575)

[2.1.4 Cấu trúc dữ liệu và giải thuật được dùng 9](#_Toc27224576)

[2.2 Chi tiết chương trình 10](#_Toc27224577)

[2.2.1 Thiết kế cấu trúc và giải thuật 10](#_Toc27224578)

[2.2.1.1 Nhập và kiểm tra dữ liệu đưa vào 10](#_Toc27224579)

[2.2.1.2 Cấu trúc dữ liệu Stack 12](#_Toc27224580)

[2.2.1.3 Chuyển biểu thức trung tố thành hậu tố bằng ứng dụng Stack. 14](#_Toc27224581)

[2.2.1.4 Dùng Stack để tính toán biểu thức hậu tố đã được chuyển từ biểu thức trung tố 16](#_Toc27224582)

[2.2.2 Cài đặt và kiểm thử 18](#_Toc27224583)

[3 Mức độ hoàn thành và đánh giá chương trình: 19](#_Toc27224584)

[3.1 Mức độ hoàn thành 19](#_Toc27224585)

[3.2 Đánh giá 19](#_Toc27224586)

[4 Tài liệu tham khảo: 19](#_Toc27224587)

**Mục lục hình**

[Hình 1: Quá trình chuyển đổi 14](#_Toc27139425)

[Hình 2: Kết quả thực hiện 16](#_Toc27139426)

**Mục lục bảng biểu**

[Bảng 1: Thông tin nhóm 7](#_Toc27139434)

[Bảng 2: Bảng kế hoạch thực hiện 8](#_Toc27139435)

[Bảng 3: Bảng kiểm thử chương trình 18](#_Toc27139436)

# TỔNG QUAN NHÓM

## Thông tin nhóm

|  |  |  |  |
| --- | --- | --- | --- |
| MSSV | Họ và Tên | Email | Vai trò |
|  |  |
| 17110118 | Lê Trần Ngọc Đạt | 17110118@ student.hcmute.edu.vn | Nhóm trưởng |
|  |
| 16110075 | Phan Thanh Hoài | 16110075@ student.hcmute.edu.vn | Thành viên |

*Bảng 1: Thông tin nhóm*

## Phương thức làm việc

### Quy trình

* Thống nhất đề tài đồ án.
* Phân công nhiệm vụ cho từng thành viên trong nhóm thực hiện.
* Tìm hiểu tài liệu liên quan đến đề tài.
* Lên kế hoạch, các mốc thời gian cho đồ án.
* Tiến hành các buổi họp nhóm trao đổi ý kiến, thông tin, quy trình…
* Hoàn thành đồ án, ra sản phẩm
* Đánh giá sản phẩn, hoàn thiện sản phẩm.
* Hoàn thành báo cáo đồ án.

### Kế hoạch thực hiện đồ án

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Ngày bắt đầu** | **Ngày kết thúc** | **Danh Sách Công việc** | **Nội dung chi tiết công việc** | **Hoàn thành** |
| 05/09/2019 | 10/09/2019 | Xác định đề tài | -Tìm hiểu cơ bản về yêu cầu đề tài  -Lập mô hình | X |
| 05/09/2019 | 10/09/2019 | Tìm hiểu thuật toán cho đề tài | -Thuật toán đệ quy | X |
| 10/09/2019 | 15/09/2019 | Tìm hiểu về Stack | -Stack là gì?  -Một số phép toàn trên Stack  Tìm hiểu thêm về 1 số ứng dụng của Stack | X |
| 20/09/2019 | 30/09/2019 | Tìm hiểu về “Biểu thức trung tố”. | -Thế nào là biểu thức trung tố.  -Cách giải 1 biểu thức trung tố. | X |
| 30/09/2019 | 30/10/2019 | Triển khai áp dụng thuật toán | -Ứng dụng ngăn xếp giải bài toán Biểu Thức Trung Tố | X |
| 30/10/2019 | 10/11/2019 | Triển khai xây dựng slide | Thiết kế xây dựng slide về nội dung của nhóm | X |
| 10/11/2019 | 20/10/2019 | Triển Khai viết báo cáo | Tổng hợp tất cả các nội dung đã tìm kiếm được, đánh giá cá nhân. | X |

*Bảng 2: Bảng kế hoạch thực hiện*

# BÁO CÁO ĐỒ ÁN

## Giới thiệu chung

### Tên đồ án

Xây dựng chương trình “Tính Biểu thức trung tố ứng dụng Stack”.

### Lý do chọn đề tài

Biểu thức trung tố là một biểu thức phức tạp và bao gồm nhiều toán hạng, toán tử khác nhau, yêu cầu người giải phải mất nhiều thời gian và tập trung tối đa để có thể giải một cách chính xác.

Nhiều trường hợp biểu thức đã cho không hợp lệ, dẫn đến chúng ta có thể mất nhiều thời gian vô ích để giải. Vì vậy chúng em đã chọn đề tài này để tạo nên 1 chương trình giúp giải nhanh và chính xác một “ biểu thức trung tố” đã cho trước.

### Các chức năng của chương trình

Chương trình tính biểu thức trung tố là chương trình để người dùng có thể Kiểm tra cũng như tính toán 1 biểu thức trung tố có sẵn một cách nhanh chóng và chính xác nhất.

Chương trình Ứng dụng kĩ thuật là Stack – Ngăn xếp là cấu trúc dữ liệu quan trọng, là kiến thức không thể thiếu trong Khoa học Máy tính và được ứng dụng rất nhiều trong lập trình.

Nó là kiểu dữ liệu cơ bản để giải những bài toán từ đơn giản đến phức tạp, nhiều bài toán phức tạp đã được đơn giản hóa đi rất nhiều nhờ loại cấu trúc dữ liệu này.

### Cấu trúc dữ liệu và giải thuật được dùng

Chương trình “Tính biểu thưc trung tố ứng dụng Stack” được viết với ngôn ngữ lập trình C++ và có sử dụng những thuật toán đã học của sinh viên Công Nghệ Thông Tin khoa Chất Lượng Cao của trường đại học Sư Phạm Kĩ Thuật Thành Phố Hồ Chí Minh.

Chương trình áp dụng thuật toán đệ quy để gọi lại hàm tính toán cho chương trình. Chương trình áp dụng cấu trúc dữ liệu Ngăn Xếp (Stack) để có thể giải bài toán một cách tuần tự và chính xác nhất. Ngoài ra chương trình cũng dùng một số thuật toán cơ bản của ngôn ngữ C để tính toán và xóa khoảng trắng.

## Chi tiết chương trình

### Thiết kế cấu trúc và giải thuật

#### Nhập và kiểm tra dữ liệu đưa vào

Chương trình sủ dụng hàm nhập kiểu dữ liệu String để cho người dùng nhập vào một chuỗi là các kí tự có trong một biểu thức trung tố cần tính.

Hàm nhập dữ liệu:

void Input(string& EXP)

{

cout << "Enter: ";

getline(cin, EXP);

}

Sau đó dùng hàm xóa khoảng trắng để xóa khoảng trắng của dữ liệu vào:

Hàm xóa khoảng trắng:

void deleteSpace(string& str)

{

for (int i = 0; i < str.length(); i++)

{

while (str[i] == ' ' && i < str.length()) { str.erase(i, 1); }

}

}

Hàm kiểm tra tính hợp lệ của ký tự toán nhập vào:

bool isoperator(char s)

{

if (s == '^' || s == '\*' || s == '/' || s == '+' || s == '-') return true;

return false;

}

bool isbracket(char s)

{

if (s == '(' || s == ')') return true;

return false;

}

Sau đó dữ liệu sẽ được đưa vào hàm kiểm tra (CheckExpression) để kiểm tra xem chuỗi mà người dùng nhập vào có là một biểu thức trung tố hợp lệ hay không.

bool check(string EXP)

{

for (int i = 0; i < EXP.length(); i++)

{

if (!isdigit(EXP[i]) && !isoperator(EXP[i]) &&

!isbracket(EXP[i]) && EXP[i] != '.') return false;

}

if (EXP[0] == '.' || EXP[EXP.length() - 1] == '.') return false;

int count = 0;

for (int i = 0; i < EXP.length(); i++)

{

if (isdigit(EXP[i])) { count++; }

if (EXP[i] == '.' && (!isdigit(EXP[i - 1]) || !isdigit(EXP[i + 1])))

return false;

}

if (count == 0) return false;

if (isoperator(EXP[0])) return false;

if (isoperator(EXP[EXP.length() - 1])) return false;

for (int i = 1; i < EXP.length() - 1; i++)

{

if (isoperator(EXP[i]))

{

if (!isdigit(EXP[i - 1]) && !isbracket(EXP[i - 1])) return false;

if (!isdigit(EXP[i + 1]) && !isbracket(EXP[i + 1])) return false;

}

}

int o = 0, c = 0;

for (int i = 0; i < EXP.length(); i++)

{

if (EXP[i] == '(') o++;

if (EXP[i] == ')') c++;

}

}

#### Cấu trúc dữ liệu Stack

Chương trình sử dụng cấu trúc dự liệu để xử lý các kí tự biểu thức trung tố đẻ chuyển thành hậu tố, Gồm các thao tác:

* Isempty: Kiểm tra ngăn xếp có đang rỗng hay không.
* Pop: Lấy phần tử từ trong ngăn xếp ra để xử lý
* Top: Xét giá trị vào vị trí trên cùng của ngăn xếp.
* Push : Gán giá trị mới nhập từ bên ngoài vào vị trí đầu tiên.

Hàm khởi tạo Stack và các thao tác trên Stack

node\* createNode(elementType val)

{

node\* n = new node();

n->data = val;

n->next = NULL;

return n;

}

Hàm kiểm tra Stack rỗng

bool isEmpty()

{

if (this->Top == NULL) return true;

return false;

}

Hàm xem giá trị một nút đầu Stack:

elementType top()

{

if (!isEmpty())

{

return this->Top->data;

}

}

Hàm thêm một nút kiểu dữ liệu elementType

void push(elementType val)

{

node\* n = createNode(val);

if (isEmpty())

{

this->Top = n;

}

else

{

n->next = this->Top;

this->Top = n;

}

}

Hàm xóa một nút đầu Stack

elementType pop()

{

elementType val;

if (!isEmpty())

{

if (this->Top->next == NULL)

{

val = this->Top->data;

this->Top = NULL;

}

else

{

node\* n = this->Top;

this->Top = this->Top->next;

val = n->data;

delete n;

}

}

return val;

}

#### Chuyển biểu thức trung tố thành hậu tố bằng ứng dụng Stack.

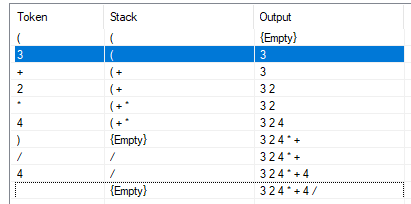
Chương trình sử dụng một Stack để lưu toán tử, và một Vector liên kết để lưu biểu thức hậu tố.

Khởi tạo Stack có dữ liệu là kiểu string có tên là ST, khởi tạo 1 Vectorn kiểu dữ liệu là string có tên là POST.

Ta bắt đầu duyệt chuỗi. Lấy giá trị kí tự tại vị trí i.

* Nếu là dấu ngoặc hoặc toán tử thì đẩy vào Stack ST.
* Nếu là toán hạng thì đẩy vào Vector liên kết.
* Chương trình sẽ đọc tiếp, nếu trước đó là dấu ngoặc “(“ thì tiếp theo sẽ là số, chương trình gặp số sẽ đẩy vào Vector, sau đó đọc tiếp gặp toán tử sẽ cho vào Stack **ST** đã lưu dấu “(“ ở Stack **ST.**
* Chương trình sẽ đọc tiếp cho đến toán tử tiếp theo và kiểm tra độ ưu tiên của toán tử thứ 2 so với toán tử thứ nhất, nếu độ ưu tiên của toán tử thứ 2 cao hơn so với độ ưu tiên của toán tử thứ nhất, thì sẽ đẩy toán tử thứ 2 sang Vector liên kết và tăng i đến vị trí tiếp theo, cứ như vậy đến khi gặp “)” thì chương trình sẽ đẩy các toán tử còn lại của Stack ST vào Vector liên kết.

VD: (3 + 2 \* 4) / 4



*Hình 1: Quá trình chuyển đổi*

Hàm xử lý số kiểu Float:

if (isdigit(infixexp[i]))

{

int j = i;

while (isdigit(infixexp[j]) || infixexp[j] == '.') j++;

string number = infixexp.substr(i, j - i);

POST.push\_back(number);

i = j - 1;

}

Hàm xử lí ngoặc:

else if (infixexp[i] == '(') ST->push(getString(infixexp[i]));

else if (infixexp[i] == ')')

{

while (!ST->isEmpty() && ST->top() != "(")

POST.push\_back(ST->pop());

ST->pop();

}

Hàm xử lý toán tử:

else if (isoperator(infixexp[i]))

{

string op = getString(infixexp[i]);

if (ST->isEmpty()) { ST->push(op); }

else

{

while (!ST->isEmpty() && rank(op) <= rank(ST->top()))

POST.push\_back(ST->pop());

ST->push(op);

}

}

Sau đó, vận dụng tất cả hàm xử lý ở trên tổng hợp lại chuyển đổi biểu thức trung tố kiểu dữ liệu String sang biểu thức hậu tố Vector<String> liên kết.

#### Dùng Stack để tính toán biểu thức hậu tố đã được chuyển từ biểu thức trung tố

Chương trình sử dụng một Stack để lưu toán hạng , và một biến để lưu kết quả.

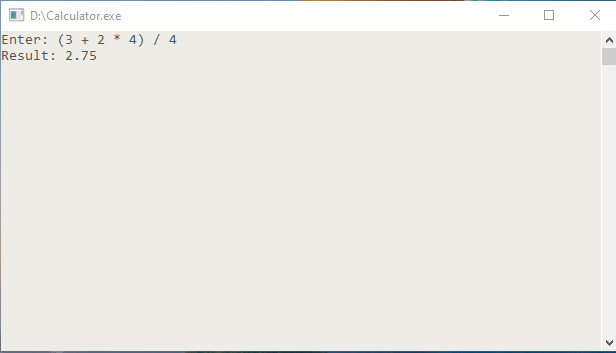
* Khởi tạo Stack có dữ liệu là kiểu string có tên là Oprand, khởi tạo 1 biến kiểu dữ liệu là double có tên là Result.
* Ta bắt đầu duyệt Vector liên kết có chứa biểu thức hậu tố. Nếu chương trình gặp 1 toán hạng thì đẩy vào Stack Oprand, sau đó nếu gặp toán tử thì lấy 2 toán hạng đầu từ Stack Oprand để thực hiện tính toán với toán tử đã bắt gặp từ Vector liên kết.
* Chương trình sẽ convert toán hạng từ kiểu String sang Double để thực hiện phép tính và trả về kết quả kiểu double, sau đó chương trình sẽ convert kết quả tính được từ kiểu dữ liệu double sang kiểu dữ liệu string và đẩy lại vào Stack Oprand, cứ như thế đến hết chương trình thì kết quả cuối cùng sẽ được lưu vào Stack dưới dạng 1 chuỗi.

Ví dụ: (3 + 2 \* 4) / 4

Chuyển sang hậu tố: (3, 2, 4, \*, +, 4, /)

* + - B1: Lấy 3, 2 , 4 vào Stack .
    - B2: Đọc phép toán nhân, và lấy 4,2 ra để thực hiện phép nhân, kết quả là 8 và đẩy vào Stack.(trong Stack còn lại (3,8)).
    - B3: Tiếp tục đọc phép toán cộng, và lấy 8,3 ra để thực hiện phép cộng, kết quả là 11 và đẩy vào Stack.(trong Stack còn lại (11)).
    - B4: Tiếp tục đọc tới số 4 thì đẩy vào Stack . trong Stack lúc này(11,4).
    - B5: Tiếp tục đọc tới phép toán chia , và lấy 4,11 ra để thực hiện phép toán chia 4/11.

Kết quả là 2.75 và đẩy vào Stack.(trong Stack bây giờ là kết quả (2.75)).



*Hình 2: Kết quả thực hiện*

Hàm thực hiện phép toán để tính kết quả từ biểu thức hậu tố:

double evaluate(vector<string> postfixexp)

{

double result;

stack\* oprand = new stack();

for (auto i = postfixexp.begin(); i != postfixexp.end(); i++)

{

if (isNumber(\*i))

{

oprand->push(\*i);

}

else

{

double num1 = stod(oprand->pop());

double num2 = stod(oprand->pop());

string op = \*i;

if (op == "+") oprand->push(to\_string(num2 + num1));

else if (op == "-") oprand->push(to\_string(num2 - num1));

else if (op == "\*") oprand->push(to\_string(num2 \* num1));

else if (op == "/") oprand->push(to\_string(num2 / num1));

else if (op == "^") oprand->push(to\_string(pow(num2, num1)));

}

}

result = stod(oprand->pop());

return result;

}

### Cài đặt và kiểm thử

|  |  |  |
| --- | --- | --- |
| **Tình huống** | **Mục đích** | **Kết quả  cách khắc phục** |
| Tình huống 1:  Nhập 2 số nguyên thực hiện các phép toán cộng trừ nhân chia đơn giản với 2 số đó. | Kiểm tra các phép toán đơn giản xem kết quả trả về có chính xác không. | Chương trình cho kết quả chính xác và nhanh chóng khi thực hiện các phép toán đơn giản đã nhập. |
| Tình huống 2:  Nhập 3 số nguyên và thực hiện các phép toán cộng trừ kết hợp với nhân chia. | Kiểm tra độ ưu tiên của các toán tử xem có đúng trình tự và cho kết quả đúng không. | Chương trình thực hiện đúng trình tự về độ ưu tiên của các toán tử và cho kết quả chính xác. |
| Tình huống 3:  Nhập nhiểu phép toán và nhiểu số hạng cùng một lúc vào trong dấu ngoặc và nhiều toán tử ở ngoài dấu ngoặc. | Kiểm tra xem chương trình có thực hiện các phép toán trong ngoặc trước sau đó mới thực hiện các phép tính ngoài dấu ngoặc và cho kết quả đúng hay không. | Chương trình thực hiện các phép toán có độ ưu tiên cao hơn ở trong ngoặc trước sau đó thực hiện các phép toán ở ngoài dấu ngoặc và cho kết quả đúng. |
| Tình huống 4:  Nhập một chuỗi kí tự không hợp lệ. | Kiểm tra xem chương trình có nhận dạng được một chương trình không hợp lệ hay không. | Chương trình xuất ra lỗi và kết thúc. |
| Tình huống 5:  Nhập vào 1 hoặc nhiều số kiểu Float và thực hiện phép toán. | Kiểm tra xem chương trình có đọc hết các kí tự của số kiểu float dưới dạng kiểu string để thực hiện phép tính đúng không. | Chương trình đã đọc chính xác các số kiểu float dưới dạng kiểu string sau đó thực hiện phép toán và cho kết quả chính xác. |

*Bảng 3: Bảng kiểm thử chương trình*

Hiện tại nhóm chưa thấy trường hợp nào mà chương trình không tính ra hoặc cho kết quả sai, nếu trong quá trình chấm thầy thấy có trường hợp nào sai xót mong thầy phản hồi để nhóm tiếp thu và rút kinh nghiệm.

# Mức độ hoàn thành và đánh giá chương trình:

## Mức độ hoàn thành

Chương trình đã thực hiện hầu hết các yêu cầu của chương trình mà thầy đưa ra, cũng như về yêu cầu của bài toán. Nhóm tự đánh giá đã hoàn thành được 100% yêu cầu.

## Đánh giá

*Ưu điểm:*

* Chương trình đã áp dụng Stack vào việc chuyển đổi biểu thức trung tố thành hậu tố cũng như tính toán biểu thức hậu tố.
* Cho ra kết quả chính xác và nhanh chóng.
* Ứng dụng mạnh mẽ ưu điểm của Cấu trúc dữ liệu Stack.

*Nhược điểm:*

* Chương trình chỉ thực hiện được các phép toán đối với các số kiểu dữ liệu Double.
* Nhiều sự dư thừa khi sử dụng Code, chưa tối ưu hóa. Dễ gây lỗi trong quá trình thực hiện.

*Hướng phát triển:*

* Chương trình thực hiện các hàm một cách riêng biệt, nên có thể phát triển sử dụng thêm một số hàm tính toán như “Căn bậc 2, hàm tích phân,…”
* Thiết kế giao diện đưa vào sử dụng thực tiễn.

# Tài liệu tham khảo:

Link 1 <https://www.youtube.com/watch?v=vq-nUF0G4fI>.

Link2: <https://www.youtube.com/watch?v=MeRb_1bddWg&list=PL2_aWCzGMAwI3W_JlcBbtYTwiQSsOTa6P&index=20>.