

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: XÂY DỰNG ỨNG DỤNG
CALCULATOR

SVTH 1 : NGUYỄN VÕ HOÀNG - 17110143
SVTH 2 : PHẠM HUỲNH THANH LÂM - 17110168

TP. Hồ Chí Minh, tháng 11 năm 2018

MỤC LỤC

PHẦN MỞ ĐẦU.....	1
1. Lý do chọn đề tài:.....	1
2. Mục đích nghiên cứu đề tài:.....	1
3. Yêu cầu :.....	1
4. Phương pháp nghiên cứu:.....	1
PHẦN NỘI DUNG.....	2
1. Chức năng Calculator.....	2
2. Nội dung chương trình.....	2
2.1 Giao diện hình máy tính calculator.....	2
2.2 Mô tả thiết kế code.....	3
2.1.1 Mô tả chức năng:.....	3
2.1.2 Chức năng xử lý biểu thức.....	4
2.1.1 Mô tả chung chuyển đổi trung tố sang hậu tố trong thuật toán:.....	9
PHẦN KẾT LUẬN.....	13
1. Kết quả đạt được.....	13
2. Kết luận.....	13
3. Bảng phân công công việc.....	15
TÀI LIỆU THAM KHẢO.....	17

DANH MỤC CÁC HÌNH

Hình 1: Giao diện của calculator.....	2
Hình 2: Hình gán giá trị toán hạng.....	4
Hình 3: Hình tạo các stack và các tham số.....	5
Hình 4: Hình nạp toán hạng vào stack có tên sh.....	5
Hình 5: Hình nạp dấu và kiểm tra dấu.....	5
Hình 6: Hình nhập các phép lượng giác và các hàm đặc biệt.....	6
Hình 7: Hình nhập toán tử vào stack có tên st.....	6
Hình 8: Hình kiểm tra phép lấy ra ở trên có thuộc các phép tính cần xét không sau đó lấy số hạng ra.....	6

Y

PHẦN MỞ ĐẦU

1. Lý do chọn đề tài:

- Lý do đề tài này được chọn vì sự thiết thực và gần gũi của máy tính đối với con người nói chung và đối với học sinh, sinh viên nói riêng.

2. Mục đích nghiên cứu đề tài:

- Mục đích của đề tài là tạo ra một giả lập của máy tính đơn giản thông qua các ngôn ngữ C đã học. Từ đó tạo ra một chương trình C giả lập áp dụng những kiến thức đã học trước đây.

3. Yêu cầu :

- Yêu cầu được đưa ra trong đề án là áp dụng phương pháp chuyển từ trung tố sang hậu tố.
- Thực hiện được các phép tính đơn giản của calculator với các kiến thức đã học của môn cấu trúc dữ liệu và những môn học khác.

4. Phương pháp nghiên cứu:

- Nghiên cứu tham khảo tra cứu thông tin và tài liệu thông qua sự cho phép của giáo viên.
- Hoàn thiện đề tài với sự hướng dẫn của giáo viên.

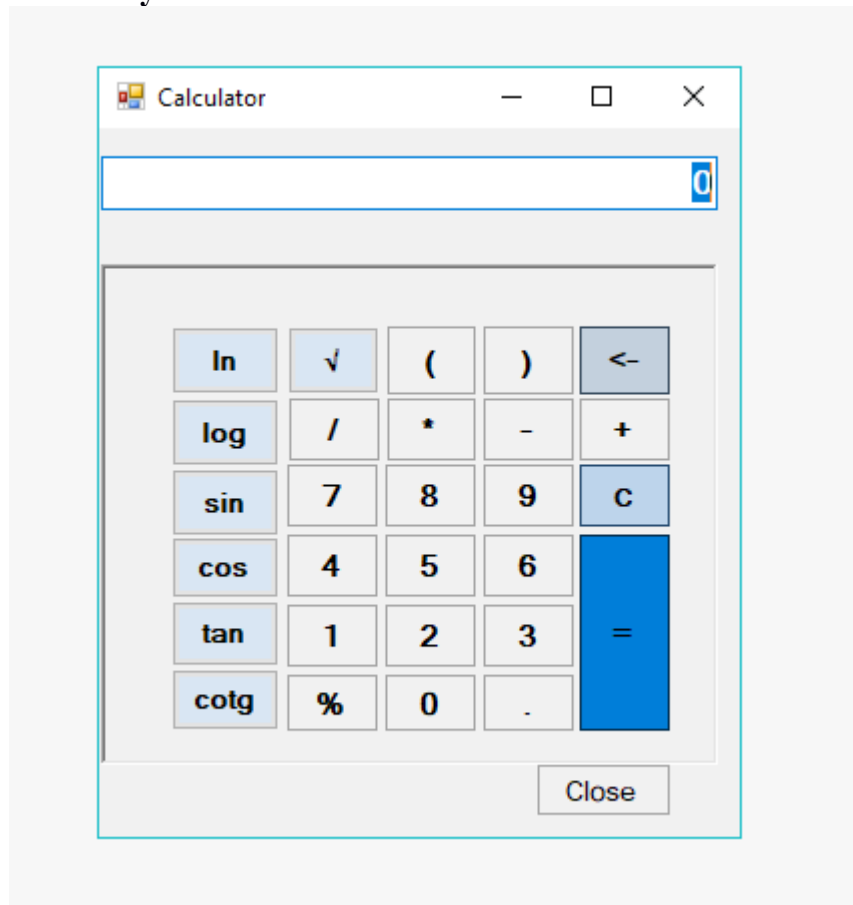
PHẦN NỘI DUNG

1. Chức năng Calculator

Gồm các chức năng tính toán cơ bản là $+$, $-$, $*$, $/$ còn bao gồm các phép toán lượng giác như \sin , \cos , \tan , \cotg , phép logarit \ln , \log và một số phép tính khác như $\sqrt{}$, $\%$. Bằng cách áp dụng trung tố Calculator có thể tạo ra chuỗi các phép tính sẽ được mô tả ở những phần dưới đây.

2. Nội dung chương trình

2.1 Giao diện hình máy tính calculator



Hình 1: Giao diện của calculator

2.2 Mô tả thiết kế code

2.1.1 Mô tả chức năng:

*Nhập toán tử và toán hạng từ nút ra màn hình textBox

Trình tự 1:

```
bool NhapLai;  
void ThemKiTu(string c)  
{  
    if ((textBox1.Text == "0"))  
        textBox1.Clear();  
    if (NhapLai)  
    {  
        textBox1.Text = "";  
        Invalidate();  
    }  
    textBox1.Text += c;  
    NhapLai = false;  
}
```

Tạo thuộc tính ThemKiTu với một chuỗi string là c, hàm nhập nhập lại bool NhapLai với các dụng xét trường hợp nhập lại của chương trình.

- Tại màn hình textBox được tạo sẵn số 0 mô phỏng với Calculator vì vậy ta cần một lệnh xóa màn hình là

```
if ((textBox1.Text == "0"))  
    textBox1.Clear();
```

Trong trường hợp cần nhập lại ta có thể sử dụng hàm Invalidate để yêu cầu control được nhập lại(hay có thể nói vẽ lại Control của window).

- Ta có thể nhập một chuỗi các ký tự ra textBox trong trường hợp không nhập lại với lệnh:

```
textBox1.Text += c;  
NhapLai = false;
```

Trình tự 2: Ta tiến hành thêm các ký tự vào các button của Calculator

- Với các trường hợp là số hạng

```
private void button0_Click(object sender, EventArgs e)  
{
```

```
ThemKiTu("0");
}
```

🚦 Với các trường hợp là toán tử

```
private void button11_Click(object sender, EventArgs e)
{
    if ((textBox1.Text == "0"))
        textBox1.Clear();
    textBox1.Text += '-';
    NhapLai = false;
}
```

🚦 Với các trường hợp các phép sin, cos, tan, cotg, sqrt, ln, log,%.Ta chỉ cần thay từ trong hàm ThemKiTu("phép").

```
private void button19_Click(object sender, EventArgs e)
{
    ThemKiTu("sin");
}
```

2.1.2 Chức năng xử lý biểu thức

○ Toán tử nhị phân và toán tử đơn phân:

Một toán tử gọi là nhị phân khi nó tác động lên 2 toán hạng, vậy cộng(+), trừ(-), nhân(*), chia(/) là toán tử nhị phân. Một toán tử đơn phân khi nó chỉ tác động lên 1 toán hạng, vậy dấu âm(-), dương(+) trước toán tử là đơn phân.

○ Trình tự chuyển đổi

➤ Một trong những điều quan trọng trước khi bắt đầu là phải tính toán được độ ưu tiên của các toán tử trong biểu thức nhập vào như sau:

```
public static int GetPriority(string op)
{
    if (op == "sin" || op == "cos" || op == "tan" || op == "cotg" || op == "ln" || op == "log" || op == "sqrt")
        return 3;
    if (op == "*" || op == "/" || op == "%" )
        return 2;
    if (op == "+" || op == "-")
        return 1;
    return 0;
}
```

Hình 2: Hình gán giá trị toán hạng

➤ Sau đó ta tạo hai hàm Stack là sh(toán hạng) và st(toán tử), tiếp khai báo double ketQua và tạo chuỗi string str1 để chứa các toán tử và vị trí các toán hạng trong và

sau lúc chuyển đổi. Stack tạm thời chứa các toán tử trong lúc chờ đưa vào str1. Theo một vòng lặp từ đầu cho đến cuối biểu thức.

```
void TinhToan()
{
    textBox1.Text += ' ';
    Stack sh = new Stack();
    Stack st = new Stack();
    int i = 0, j = 0;
    double ketQua = 0, a, b;
    string str1;
```

Hình 3: Hình tạo các stack và các tham số

❖ Như sau:

- Xét text.Box1.Text[i] xem có phải là chữ số trong khoảng 0 đến 9 hay không để tiến hành cho vào chuỗi str (chuỗi tạm để trường hợp gặp giấu ngoặc toán tử) là chuỗi được cắt ra từ textBox và cho vào stack.

```
// nạp toán hạng vào sh
if (textBox1.Text[i] >= '0' && textBox1.Text[i] <= '9')
{
    j = i;

    while (textBox1.Text[i + 1] >= '0' && textBox1.Text[i + 1] <= '9'
        || textBox1.Text[i + 1] == '.' && textBox1.Text[i + 1] != ' ' && i + 1 < textBox1.Text.Length)
        i++;
    string str = textBox1.Text.Substring(j, i + 1 - j);
    sh.Push(str.ToString());
}
```

Hình 4: Hình nạp toán hạng vào stack có tên sh

+Nếu gặp dấu“(“, push vào Stack. Ngoài ra ta còn xét trường hợp trước dấu“(“ có dấu + hay – hay không nếu có thì ta push sh là 0 vì nơi đó có st là + hoặc -.

```
// nạp dấu ( vào st
if (textBox1.Text[i] == '(')
{
    st.Push(textBox1.Text[i]);
    if (textBox1.Text[i + 1] == '-' || textBox1.Text[i + 1] == '+')
        sh.Push(0);
}
```

Hình 5: Hình nạp dấu và kiểm tra dấu

+ Xét xem phần tử trong text.Box trong có thuộc bảng chữ cái từ a đến z để kiểm tra sin, cos, tan, cotg,...và cho vào chuỗi tạm thời str con từ vị trí bắt đầu j với độ dài là i-j bằng hàm Substring().Sau đó bỏ chuỗi str vào ngăn xếp toán tử st


```

// nạp sin,cos,tan,cotg,ln,log vào toán tử st
if(textBox1.Text[i]>='a' && textBox1.Text[i]<='z')
{
    j = i;
    while (textBox1.Text[i] >= 'a' && textBox1.Text[i] <= 'z')
        i++;
    string str = textBox1.Text.Substring(j, i - j);
    st.Push((str.ToString()));
}

```

Hình 6: Hình nộ các phép lượng giác và các hàm đặc biệt
 +Nạp toán tử vào st, mức ưu tiên từ (Thấp lên cao): dấu ngoặc < Cộng,Trừ < Nhân,Chia < Dấu âm/dương(Cộng trừ đơn phân).Trong đó gồm các bước quan trọng sau:

⊙ Sau đó so sánh chuỗi được peek(lấy thông tin ra khỏi ngăn xếp) từ ngăn xếp với chuỗi text.Box so sánh sự ưu tiên của các toán tử trong textBox với toán tử đã được đưa vào trong stack trước đó.

```

// nạp toán tử vào st
if (textBox1.Text[i] == '+' || textBox1.Text[i] == '-' || textBox1.Text[i] == '*' || textBox1.Text[i] == '/' || textBox1.Text[i] == '%')
{
    while (st.Count > 0 && GetPriority(textBox1.Text[i].ToString()) <= GetPriority(st.Peek().ToString()))
    {
        str1 = st.Pop().ToString();
    }
}

```

Hình 7: Hình nộ toán tử vào stack có tên st
 ⊙Đối với các phép tính +, -, *, /: Bước đầu ta Pop các phần tử được lưu trong ngăn xếp sh vào lần lượt các phần tử mẫu tính là a và b a = double.Parse((sh.Pop().ToString())); b = double.Parse((sh.Pop().ToString()));

```

//nếu là các phép tính
if (str1 == "+" || str1 == "-" || str1 == "*" || str1 == "/" || str1 == "%")
{
    a = double.Parse((sh.Pop().ToString()));
    b = double.Parse((sh.Pop().ToString()));
    // ...
}

```

Hình 8: Hình kiểm tra phép lấy ra ở trên có thuộc các phép tính cần xét không sau đó lấy số hạng ra

⊙Sau đó hàm switch sẽ xét các trường hợp khác nhau của các phép tính nếu không có thì báo `MessageBox.Show("Không thể chia cho 0", "Thông báo");`..

- Các hàm sin, cos, tan, cotg, ln, sqrt, log chúng ta áp dụng hàm switch như trên ví dụ đây: `sh.Push(Math.Sin(double.Parse(sh.Pop().ToString())));`.

Ở đây ta xét str1 là các phép tính đặc biệt thì ta sẽ thực hiện tính hậu tố bằng cách lấy số str1(số hạng được lấy ra ở đầu) ra được kết quả ta liền đặt lại vào stack của sh.

○ Trình tự tính toán

□ Xử lý phép toán + liên tiếp và – liên tiếp trong trường hợp chúng ta sử dụng hàm if chung để xét các khả năng xuất hiện của các toán tử liên tiếp nhau như +-, --, ++, -+.

○ Đầu tiên ta có textBox1 tại vị trí i là “-” thì đặt j là 1 đánh dấu là trừ sau đó ta bắt đầu tạo ra trường hợp của dấu tiếp theo tại vị trí i+1 là gì. Trong trường hợp là “-” thì ta tiếp tục xét tiếp vì “-” với “-” là “+”

○ Sau đó ta đặt trường hợp khi $j \% 2$ không có số dư thì là + nếu có số dư là

“-”

```
if ((j) % 2 == 0)
    str1 = "+";
else
    str1 = "-";
```

rồi đưa chúng trở lại trong ngăn xếp.

○ Ngoài ra còn có các trường hợp không phải là dấu cộng trừ ta trực tiếp push nó trở lại trong st.

+Xử lý các phép toán trong ngoặc: Nếu gặp dấu “)” chúng ta pop các số ra khỏi Stack sh của nó với bước đầu tiên là xét các trường hợp toán tử và số hạng đã được nhập vào text.Box được nhập vào trước đó có nằm trong dấu ngoặc hay không → **Trường hợp trong chuỗi ký tự đã được nhập trước đó không có ký tự “(”:**

○ Đặt 1 phần tử ép kiểu double với các dữ liệu được lấy từ chuỗi trong ngăn xếp sh và chiếu st ra chuỗi str1.

○ Trong đó ta cần xét trường hợp nếu dữ liệu được pop từ st là một ký tự “(” thì khi đó chúng ta Push a trở lại sh và ngừng lại để xét trường hợp khác:

```
if (str1 == "(")
```

```

    {
        sh.Push(a);
        break;
    }

```

-Ngoài ra chúng ta còn xét các trường hợp khi ngăn xếp sh đã hết dữ liệu mà ở ngoài chuỗi str1 vẫn có các ký tự như +, - thì ta có thể đặt số b trực tiếp bằng không vì -0 hoặc +0 đều bằng 0 `if (sh.Count == 0 && (str1 == "+" || str1 == "-")) b = 0;`

-Nhưng nếu trường hợp ở ngoài chuỗi str1 là các ký tự *, / ta có thể trực tiếp khai báo lỗi vì không thể khai báo tiếp khi sh không còn dữ liệu.

```

    if (sh.Count == 0 && (str1 == "*" || str1 == "/"))
    {
        MessageBox.Show("Lỗi!");
        break;
    }

```

-Trường hợp còn lại đó chính là Stack sh còn dữ liệu và khi đó ta có thể tiến hành Pop số b từ Stack sh `b = double.Parse((sh.Pop().ToString()));`

```

//xử lí các phép toán trong dấu ngoặc
if (textBox1.Text[i] == ')')
{
    str1 = "+"; // khai báo str1 ảo(tránh stack rỗng)
    while (str1 != "(")
    {
        a = double.Parse((sh.Pop().ToString()));
        str1 = st.Pop().ToString();
        if (str1 == "(")
        {
            sh.Push(a);
            break;
        }
        if (sh.Count == 0 && (str1 == "+" || str1 == "-")) b = 0;
        else
        {
            if (sh.Count == 0 && (str1 == "*" || str1 == "/"))
            {
                MessageBox.Show("Lỗi!");
                break;
            }
            else
            {
                b = double.Parse((sh.Pop().ToString()));
                // ...
            }
        }
    }
}

```

-Tiếp đó ta xét các toán tử đơn giản trong “ (“ bằng phương pháp switch như trên.Sau kết quả của các phép tính đó được lưu trong ketQua và push trở lại trong Stach sh.

```

    if (str1 != "(")
        str1 = st.Pop().ToString();

```

+ Cuối cùng, lần lượt pop các phần tử trong Stack và push vào str1 cho tới hết. Kiểm tra coi sh còn số không và str1 còn dấu không giống cách thức như trên. Nếu sh còn số hạng thì làm như ” *Hình kiểm tra phép lấy ra ở trên có thuộc các phép tính cần xét không sau đó lấy số hạng ra*” nhưng ta đã có giá trị a nên chỉ cần lấy số hạng dưới a đặt là b.

-Thực hiện kết quả của các phép toán còn lại các phép tính còn lại bằng các phép +, -, *, / bằng lệnh switch đã được làm với chúng trước đó và đã được nói rõ ở phía trên như “ *Hình kiểm tra các trường hợp và tính toán*”. Các phép tính được thực hiện theo phép toán hậu tố số a là nằm sau còn b là số nằm đầu vì ta lấy ra theo stack nên theo mức tính ưu tiên ta lấy số b thực hiện phép tính với số a.

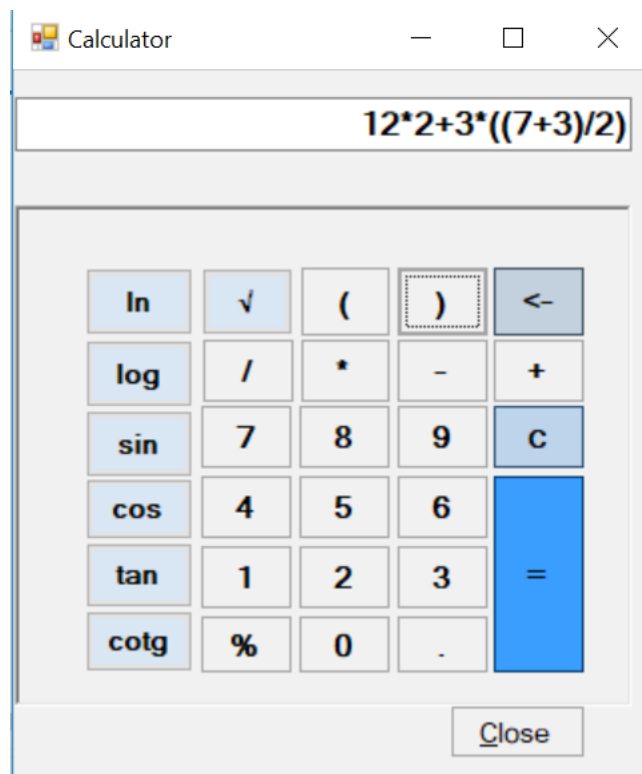
-Nếu không có phép tính trên ta xét đến các phép tính sin, cos, tan, cotg, sqrt, ln Log. Sau khi hoàn thành quá trình tính toán chúng ta đưa kết quả cuối cùng của ngăn xếp sh ra màn hình textBox1 để xuất ra kết quả của phép tính ra màn hình.

2.1.1 Mô tả chung chuyển đổi trung tố sang hậu tố trong thuật toán:

□ Tóm lại:

- Nếu là toán hạng: cho vào stack sh
- Nếu là dấu mở ngoặc “(“: cho vào stack st.
- Nếu là dấu đóng ngoặc “)”: lấy các toán tử trong stack ra và cho vào str1 cho đến khi gặp dấu mở ngoặc “(“ . (Dấu mở ngoặc cũng phải được đưa ra khỏi stack)
- Nếu là toán tử:
 - Chùng nào ở đỉnh stack là toán tử và toán tử đó có độ ưu tiên **lớn hơn hoặc bằng** toán tử hiện tại thì lấy toán tử đó ra khỏi stack và cho ra str1. Đưa toán tử hiện tại vào stack. Sau khi duyệt hết biểu thức trung tố, nếu trong stack còn phần tử thì lấy các ký tự trong đó ra và cho lần lượt vào str1.

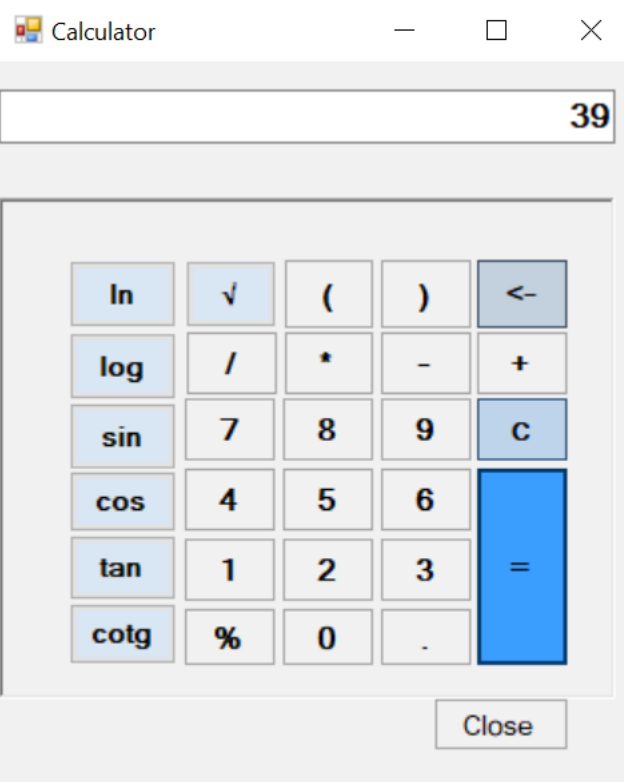
© Và sau đây là ví dụ về thuật toán trên : Chuyển biểu thức từ trung tố sang hậu tố $12*2+3*((7-3)/2)$



Và sau đây là chuỗi quá trình thực hiện biểu thức: $12*2+3*((7+3)/2)$

Ký tự	Stack st	Stack sh	String str1
12	Empty	12	12
*	*	12	12
2	*	12 2	12 2
+	+	12 2	12 2*
3	+	12 2 3	12 2* 3
*	+	12 2 3	12 2* 3
(+(12 2 3	12 2* 3
(+(12 2 3	12 2* 3
7	+(12 2 3 7	12 2* 3 7
+	+(12 2 3 7	12 2* 3 7
3	+(12 2 3 7 3	12 2* 3 7 3
)	+(12 2 3 7 3	12 2* 3 7 3+
/	+(12 2 3 7 3	12 2* 3 7 3+
2	+(12 2 3 7 3 2	12 2* 3 7 3+2
)	+	12 2 3 7 3 2	12 2* 3 7 3+2 / *
Final	Empty	12 2 3 7 3 2	12 2* 3 7 3+2 / * +

Sau đó ta có kết quả là:



⊙ Trong trường là các phép toán Sin, Cos, tan,...

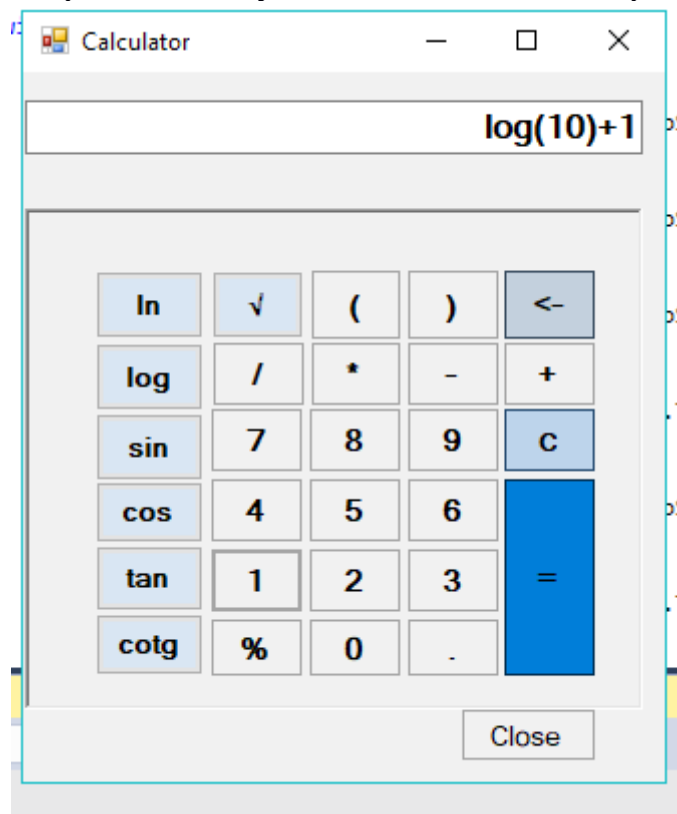
Ta chuyển về trung tố sang trung tố sang hậu tố sẽ được thực hiện có thể có trường hợp sau:

- Khi $\log(10*10)$ thì ta sẽ có kết quả là 2 và thực hiện tính chuyển hậu tố sang trung tố bình thường.

- Khi $\log(10) + 1$ thì ta thực hiện như sau:

Ký tự	Stack st	Stack sh	String str1
Log	Log	Empty	Empy
(Log (Empty	Empy
10	Log (10	10
)	Log (10	10 log
+	+	10	10 log
1		10 1	10 log 1 +
Final	Empty	10 1	10 log 1 +
Result			2

Phép tính trên máy và được tính toán ra kết quả là 2:



PHẦN KẾT LUẬN

1. Kết quả đạt được

Ta có một chương trình calculator đơn giản với các phép tính toán học. Qua bài báo cáo trên ta được biết rõ hơn các phép tính của toán học và phương pháp chuyển trung tố sang hậu tố để máy tính thực hiện và từ hậu tố chúng ta tính được kết quả của một phép toán. Chúng ta còn biết được các ứng dụng của ngăn xếp trong cấu trúc dữ liệu với việc ở đây ta sử dụng hai stack

2. Kết luận

a. Ưu điểm

- Sau khi hoàn thành đồ án này thì chúng ta tự nhận thấy đã hoàn thành tốt phần nào những yêu cầu của đề tài đã đặt ra, và cả những mục tiêu mà nhóm chúng tôi đã tự đặt ra cho bản thân. Nhưng bên cạnh đó trong quá trình làm thì còn những mục tiêu mà nhóm chúng ta tự nhận thấy là mình chưa thực hiện tốt

lầm và còn thiếu sót trong quá trình thực hiện. Và chúng ta đã sửa lại bằng việc xem lại các phép toán log, ln và sửa các phép cho chính xác hơn.

-Đồ án được sử dụng các code về phép tính và để sử dụng mà chúng ta đã được học để hoàn thành .Ngoài ra còn áp dụng được các tính chất và cấu trúc đã học trong quá trình của môn. Bằng cách đưa ra hai ngăn xếp stack là sh số hạng và st toán tử.

b. Nhược điểm

-Tuy nhiên đồ án này của chúng em tuy hoàn thành nhưng chưa hoàn thiện vì vẫn còn một số sai sót trong đồ án trong việc thực hiện các phép tính phức tạp chương trình vẫn gặp phải một số trường hợp sai số và vẫn chưa chính xác .Từ đó rút ra được những sai sót của bản thân mà chúng em cần khắc phục.Như tính hàm số lượng giác như $\sin(90)$ hay $\cos(180)$ nếu máy tính thì đều sẽ ra là 1, nhưng đối với bài chúng ta kết quả của $\sin(90)$ là 0.893996663600558 và $\cos(180)$ là -0.598460069057858 đồng thời các phép tính $\sin(4+5)-1$ cũng có vài sai sót còn được sửa

3. Bảng phân công công việc

Công việc	Nguyễn Võ Hoàng	Phạm Huỳnh Thanh Lâm
Tìm kiếm thông tin về calculator		
Thông tin của 1 phần mềm máy tính có:	X	X
Các phím số từ 0 đến 9	X	X
Các phím tính toán cộng (+), trừ(-), nhân(*), chia(/), phần trăm(%)	X	X
Các phím xóa có dạng hình (<=), phím xóa tất cả ©, phím ()	X	X
Phím bằng để in kết quả (=) , phím (- / +) để ẩn được số âm hay số dương	X	X
Các phím chứa các hàm (sin ,cos ,tan,ln,log , sqrt)	X	X
Phân tích các thông tin tìm được	X	X
Thảo luận các phần đã phân tích	X	X
Chọn lọc thông tin được về calculator về việc code		
Các chức năng của thanh Toolbox để tạo một giao diện máy tính giống của một chiếc điện thoại. Viết bằng c#	X	
Các công thức toán đơn giản và phức tạp		X
Trình tự chuyển đổi để tính	X	X
Bước 1: Tạo chuỗi Stack sh và st		
Bước 2: Các bước thực hiện		
Toán hạng		
Nếu gặp số thì đưa vào sh		
Nếu gặp dấu "(" đưa vào st		
Nếu gặp dấu ")" đưa các toán tử trong st vào cho tới khi gặp dấu "(" thì bỏ "(" đi		
Toán tử		
Kiểm tra mức độ ưu tiên của toán dấu ngoặc < cộng, trừ < nhân, chia < dấu âm/dương		
Bước 3 Cuối cùng lần lượt đưa các phần tử còn lại vào trong Stack cho tới hết		
Thiết kế giao diện		
Tạo lập một chiếc máy tính giống điện thoại	X	X
Vị trí của các con số, các hàm tính toán, các phép tính, dấu bằng , dấu ngoặc	X	X
Tùy chỉnh kích thước của mỗi ô chứa các phím có kích thước khác nhau	X	
Dùng các hình phép toán đặc biệt của word	X	
Viết code cho mỗi ô số và phép tính		
Phím ô chứa số : Hàm nhập hiện số khi ấn phím cho các ô chứa số từ 0 đến 9	X	
Chức năng của hàm nhập số từ 0 đến 9		

Phím ô chứa các dấu cộng trừ nhân chia: hàm chứa các phép tính (+, -, x, /)	X	X
Phép cộng : value1+value2 nếu ta ấn "+"	X	
Phép trừ : value1-value2 nếu ta ấn "-"	X	
Phép nhân : value1xvalue2 nếu ta ấn "x"		X
Phép chia : value1/value2 nếu ta ấn "/"		X
Viết code cho ô xóa giá trị và các hàm		
Phím ô chứa dấu (<=) khi ấn sẽ xóa con số hay hàm hoặc phép tính	X	
Phím ô chứa ký hiệu © khi ấn sẽ xóa tất cả kết quả	X	
Phím ô chứa hàm sin,cos,tan,ln,log,sqrt		X
Viết các code cho các ô còn lại		
Phím ô chứa giá trị được định sẵn giá trị	X	
Phím ô chứa giá trị tính phần trăm	X	
Hàm tính giá ra số thập phân		X
Hàm nhập phím dấu phẩy và đóng mở ngoặc		X
Xem lại và tìm kiếm các phần trong code liên quan đến môn học		
Mỗi hàm ứng dụng của môn học nào , phần nào của môn học đó	X	X
+Sửa chữa cho code phù hợp		
+Hiểu rõ phần code đó nói về gì		
Ứng dụng trong các phần cấu trúc dữ liệu(ngăn xếp)	X	X
Viết bản trình bày về đồ án		
Mục đích yêu cầu của ta khi làm về ứng dụng calculator	X	
Đánh các phần về chức năng ý nghĩa của mỗi phần ra word	X	
Bố trí các nội dung :		X
Mô tả project		X
Kết luận về các vấn đề khi làm hay mức độ của bài		X
Các biểu mẫu cho bài	X	
Thiết kế giao diện	X	
Trình tự các hàm của bài	X	
Chỉnh phong cho đồ án	X	
Tổng hợp lại kết quả và sửa lỗi sai		
Chạy lại chương trình với tất cả trường hợp mình có	X	X
Chỉnh lại các trường hợp nếu sai ở trường hợp nào(nếu có)	X	X
Chạy lại lần cuối và xem xét thời gian ,số kết quả	X	X
Kiểm tra kỹ lại trước khi nộp và thuyết trình về bài đồ án		
Kiểm tra và thuyết trình	X	X

TÀI LIỆU THAM KHẢO

[1].<http://luanvan.net.vn/luan-van/do-an-xay-dung-phan-mem-ung-dung-may-tinh-khoa-hoc-cho-dien-thoai-cam-thay-thong-minh-chay-tren-he-dieu-hanh-android-31312/>