

# **RƯỜNG 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: Tính Giá Trị Biểu Thức Trung Tố**

**SVTH : NGUYỄN HOÀNG KIM**

**MSSV : 16110132**

**SVTH : LÊ BÁ HUỲNH**

**MSSV : 16110095**

**TP. Hồ Chí Minh, tháng 12 năm 2017**

# **1.GIỚI THIỆU ĐỒ ÁN**

* 1. ***Đặt vấn đề***

Trước đây khi máy tính chưa được sử dụng rộng rãi, các nhà kế toán phải tính bằng tay hoặc trên giấy, như vậy tốn rất nhiều thời gian, dù có giàu kinh nghiệm đến mấy thì sai xót vẫn có thể xảy ra. Ngày nay chúng ta có thể tính toán dễ dàng bằng chương trình bảng tính đơn giản dễ sử dụng, có độ chính xác cao và tiết kiệm nhiều thời gian.

Các biểu thức số học vốn đã quen thuộc thân thiện với người sử dụng nhưng lại khác lạ với máy tính. Vấn đề đặt ra là làm thế nào máy tính hiểu được phép tính và trả lại kết quả phục vụ người dung.

Có nhiều phương pháp quy ước, đồ án này trình bày phương pháp xây dựng cách tính toán dựa trên ký pháp Ba Lan (Polist Notation) sử dụng ưu điểm của Stack.

* 1. ***Sơ đồ nội dung phần mềm***

*Biểu đồ 1.2 sơ đồ nội dung phần mềm*

* 1. ***Tình huống sử dụng***

Với các biểu thức toán học đơn giản (như a+b) thì bạn có thể tự làm bằng các phương pháp tách chuỗi “thủ công”. Nhưng để “giải quyết” các biểu thức có dấu ngoặc, ví dụ như (a+b)\*c + (d+e)\*f ,  thì các phương pháp tách chuỗi đơn giản đều không khả thi. Trong tình huống này, ta phải dùng đến Ký Pháp Nghịch Đảo Ba Lan (Reserve Polish Notation – RPN), một thuật toán “kinh điển” trong lĩnh vực trình biên dịch.

# **2.MÔ TẢ QUÁ TRÌNH LÀM**

***2.1 Giao diện***

***2.2 Các hàm chính***

void dangHauTo(char B[],Stack ToanTu,char C[])

{

int l=strlen(B);

int i=0,k=0;

char x;

while(i<l)

{

if(B[i]==' ')

{

if(kiemtra(B[i-1])==1)

i++;

else

{

C[k++]=B[i];

C[k]='\0';

i++;

}

}

else

{

if(kiemtra(B[i])==0)

{

C[k++]=B[i];

C[k]='\0';

i++;

}

else

{

if(B[i]=='(')

{

push(ToanTu,CreateNode(B[i]));

i++;

}

else

{

if(B[i]==')')

{

while(ToanTu.pTop->info!='(')

{

C[k++]=' ';

x=pop(ToanTu,x);

C[k++]=x;

C[k++]=' ';

C[k]='\0';

}

pop(ToanTu,x);

i++;

}

else

{

if(kiemtra(B[i])==1)

{

if(ToanTu.pTop==NULL||ToanTu.pTop->info=='(')

{

push(ToanTu,CreateNode(B[i]));

i++;

}

else

{

if(thutuUuTien(ToanTu.pTop->info)==2)

{

x=pop(ToanTu,x);

C[k++]=x;

C[k++]=' ';

C[k]='\0';

push(ToanTu,CreateNode(B[i]));

}

else

{

if(thutuUuTien(B[i])==2)

{

push(ToanTu,CreateNode(B[i]));

}

else

{

x=pop(ToanTu,x);

C[k++]=x;

C[k++]=' ';

C[k]='\0';

push(ToanTu,CreateNode(B[i]));

}

}

i++;

}

}

else

i++;

}

}

}

}

}

C[k++]=' ';

C[k]='\0';

while(ToanTu.pTop!=NULL)

{

x=pop(ToanTu,x);

C[k++]=x;

C[k++]=' ';

C[k]='\0';

}

}

void ketqua(char C[],char x[])

{

int l=strlen(C);

int i,j=timVT(C),k=0,t=0;

char y[size],z=C[j];

i=j-1;

int d=0;

while(d<=1)

{

if(C[i]!=' ')

{

x[k++]=C[i];

i--;

}

else

{

d++;

i--;

}

}

x[k]='\0';

strrev(x);

while(d<3 && i>=0)

{

if(C[i]!=' ')

{

y[t++]=C[i];

i--;

}

else

{

d++;

i--;

}

}

y[t]='\0';

strrev(y);

i=i+1;

while(i!=timVT(C))

xoaVTchuoi(C,i);

xoaVTchuoi(C,i);

TinhToan(x,y,z);

strrev(x);

ThemVitri(C,' ',i+1);

for(k=0;k<int(strlen(x));k++)

ThemVitri(C,x[k],i+1);

chuanhoa(C);

int q=0;

while(q<int(strlen(C)))

{

if(kiemtra(C[q])==1)

ketqua(C,x);

else

q++;

}

}

***2.3 Cài đặt***

Chương trình được cài đặt bằng ngôn ngữ C/C++ trên Visual C++ 6.0.

Hàm xet dùng stack và chuỗi kí tự để đưa biểu thức về dạng hậu tố.

Hàm ketqua để tách dạng hậu tố về 2 chuỗi số và 1 kí tư toán tử,rồi tính toán 2 chuỗi số cùng với kí tự toán tử đã được tách ra, dùng đệ quy để tính tiếp, kết thúc là 1 chuỗi số.

Hàm xuatkq để đưa chuỗi số đã được tính thành kq dạng số.

# **3.PHÂN CÔNG CÔNG VIỆC**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Công việc** | **Lê Bá Huỳnh** | **Nguyễn Hoàng Kim** | **Ghi chú** |
| 1 | Xử lí biểu thức | x |  |  |
| 2 | Đưa về dạng hậu tố | x |  |  |
| 3 | Tính toán |  | x |  |
| 4 | Xuất kết quả |  | x |  |

# **4.KẾT LUẬN**

***4.1 Kết quả đạt được***

Đồ án đã trình bày các bước xây dựng một biểu thức hậu tố từ một biểu thức trung tố thông thường và tính giá trị biểu thức nhờ ứng dụng của Stack.

***4.2 Khó khăn***

Nhóm chúng em gặp có lỗi khi ép chuỗi kí tự kết quả thành số thực, nhưng nhờ sự lời góp ý của thầy Tú và các bạn mà chúng em đã sửa được lỗi này. Cũng nhưng khi xử lý toán tử ‘(‘ và ‘)’ và khi tính toán số âm. Vì sử dụng ide cũ nên cũng ko thiết kế được giao diện cho dễ sử dụng, tuy có tham giảo cách dùng form bên C# nhưng chưa thể đạt được.

***4.3 Ưu điểm***

Có thể tính toán các phép đặc biệt như số dương trừ số âm (6- -9 =15), và số âm trừ số âm (-6-8=-14), và cũng như có thể tính toán được số thực.

# **MỤC LỤC**

Contents

[**1.GIỚI THIỆU ĐỒ ÁN** 2](#_Toc500833966)

[**2.MÔ TẢ QUÁ TRÌNH LÀM** 3](#_Toc500833967)

[**3.PHÂN CÔNG CÔNG VIỆC** 10](#_Toc500833968)

[**4.KẾT LUẬN** 10](#_Toc500833969)

[**MỤC LỤC** 11](#_Toc500833970)