bool operand(String d){

if(d=="+"||d=="-"||d=="\*"||d=="/"||d=="("||d==")"){

return 0;

} //operand函式 判斷是否為運算元

}

int priority(String d,bool c){

if(d=="+"||d=="-"){

return 1; //判斷運算子的優先度

}else if(d=="\*"||d=="/"){ //左括號的優先度丟進堆疊後會變最小

return 2;

}else if(d=="("&&c){

return 3;

}else{

return 0;

}

}

//將多項式字串拆解

int c=0; //計算陣列中現在放了幾個值

bool first=1; //是否為讀的第一個值

String \*data; //將多項式字串分割放進data陣列

String Data=Lab->Caption;

data=new String[Data.Length()];

for(int i=0;i<Data.Length();i++){

if(operand(Data.SubString(i+1,1))){ //如果是運算元 放進data陣列

if(first) first=0; //直到遇到運算子

data[c]+=Data.SubString(i+1,1);

}else{

if(first){

first=0; //若為第一個值直接放進data[0]

data[c++]=Data.SubString(i+1,1);

}else{

if(operand(Data.SubString(i,1))) c++; //若上個為運算元 先移到下個房間

data[c]=Data.SubString(i+1,1); //將運算子放進陣列後 移到下個房間

c++;

}

}

}

if(!(operand(Data.SubString(Data.Length(),1)))) c--; //若結尾是運算子c—，否則c多算一次

//轉換成後序表示法

String stack[10];

bool in=1; //區分”(”是在外面還是在堆疊中

int top=-1,c1=0; //top表堆疊頂端

String \*post; //c1表後序陣列中現在有幾個值

post=new String[c+1]; //後序表示法的陣列

for(int i=0;i<=c;i++){

if(data[i]==")"){ //如果遇到”)”

for(int j=top;;j--){ //將運算子輸出到post陣列直到遇到”(”

if(stack[j]!="("){

post[c1++]=stack[top--];

}else{

top--;

break;

}

}

in=1;

}else{

if(operand(data[i])){ //遇到運算元 直接丟入陣列

post[c1++]=data[i];

}else{

if(top>=0&&priority(stack[top],in)>=priority(data[i],in)){

for(int j=top;j>-1;j--){ //若堆疊頂端值的優先度較大

if(top>=0&&priority(stack[j],in)>=priority(data[i],in)){

post[c1++]=stack[top--]; //由頂端開始輸出運算子

}else{ //直到遇到優先度小於的

break;

}

}

stack[++top]=data[i]; //最後將其新增到堆疊頂端

}else{

stack[++top]=data[i]; //若堆疊頂端的優先度較小 直接疊上去

if(data[i]=="(") in=0; //左括號丟進堆疊後in=0 優先度變最小

}

}

}

}

for(int i=top;i>-1;i--){ //堆疊中剩下的值直接輸出到post

post[c1++]=stack[i];

}

c1=0;

top=-1;

//運用後序來做運算

double sum;

for(int i=0;i<=c;i++){

if(operand(post[i])){ //遇到運算元直接丟進堆疊中

stack[++top]=post[i];

}else{

if(post[i]=="+"){

sum=stack[top-1].ToDouble()+stack[top].ToDouble();

stack[--top]=sum; //遇到運算子 從堆疊找上面兩個來做運算再丟回去

}else if(post[i]=="-"){

sum=stack[top-1].ToDouble()-stack[top].ToDouble();

stack[--top]=sum;

}else if(post[i]=="\*"){

sum=stack[top-1].ToDouble()\*stack[top].ToDouble();

stack[--top]=sum;

}else{

sum=stack[top-1].ToDouble()/stack[top].ToDouble();

stack[--top]=sum;

}

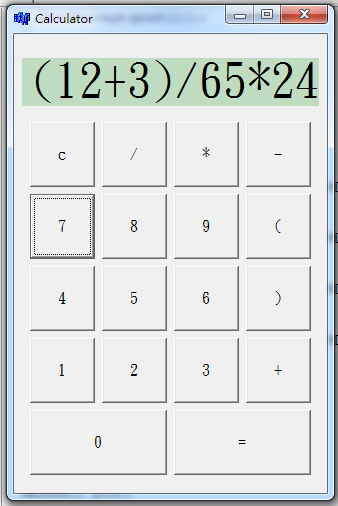
}

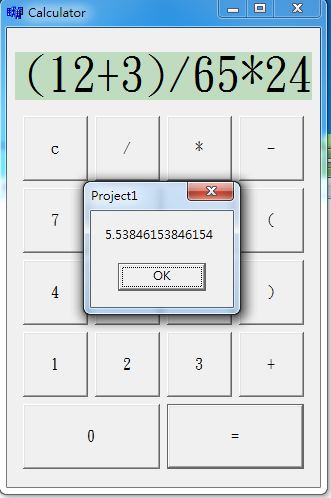
}

ShowMessage(stack[0]); //stack[0]便是最後的答案

delete[] data;

delete[] post;

輸入一串「符合規定」的多項式

按等於後，程式分為三步驟

一 將字串拆解放進陣列中

二 中序轉為後序表示法

三 利用後序表示法進行運算