Question:

Write a program that accepts an expression and evaluates it.

Hint:

For the purpose of simplicity, the following assumptions can be taken:

1. Symbols representing operations are one character long.
2. All operations are binary operations.
3. Input is in infix format.

Solution:

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include<stdio.h>

#include<conio.h>

#include<stdlib.h>

int Remove(float\*,char\*,int,int);

int check(char ,float\*,char\*,int );

void print(float\*,char\*,int);

int main()

{

char str[200],\*ops;

int number=0,c\_number,i,j;

float\*num;

clrscr();

printf("ENTER THE EXPRESSION : \n");

printf("THE EXPRESSION SHOULD BE IN INFIX FORMAT AND SHOULD NOT INCLUDE SPACES...\n");

scanf("%s",str);

i=0;

while(str[i])

{

if((int)str[i]<48||(int)str[i]>57)

number++;

i++;

}

number++;

//number stores the total number of operands

num=(float\*)malloc(sizeof(float)\*(number));

ops=(char\*)malloc(sizeof(char)\*(number-1));

for(i=0;i<number;i++)

num[i]=0;

i=j=0;

while(str[i])

{

if((int)str[i]<48||(int)str[i]>57)

ops[j++]=str[i];

i++;

}

i=j=0;

while(str[i])

{

while(((int)str[i]>=48&&(int)str[i]<=57))

num[j]=num[j]\*10+(int)str[i++]-48;

i++;j++;

}

c\_number=number;

for(i=0;i<c\_number-1;i++)

{

number=check('/',num,ops,number);

number=check('\*',num,ops,number);

number=check('-',num,ops,number);

number=check('+',num,ops,number);

}

printf("\n\nFinal Answer = %g\n",num[0]);

getch();

return 1;

}

int check(char op,float\*num,char\*ops,int number)

{

int j;

while(1)

{

j=0;

while(ops[j]!=op&&j<number)

{

j++;

if(j>number-1)

break;

}

if(j>number-1)

return number;

if(j<number-1)

{

if(op=='/')

num[j]=num[j]/num[j+1];

else if(op=='\*')

num[j]=num[j]\*num[j+1];

else if(op=='+')

num[j]=num[j]+num[j+1];

else

num[j]=num[j]-num[j+1];

number=Remove(num,ops,j,number);

continue;

}

break;

}

return number;

}

void print(float\*num,char\*ops,int number)

{

int i;

for(i=0;i<number-1;i++)

printf("%g %c ",num[i],ops[i]);

printf("%g\n",num[i]);

}

int Remove(float \*num,char \*ops,int j,int number)

{

int i;

for(i=j+1;i<number-1;i++)

{

num[i]=num[i+1];

ops[i-1]=ops[i];

}

number--;

print(num,ops,number);

return number;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/