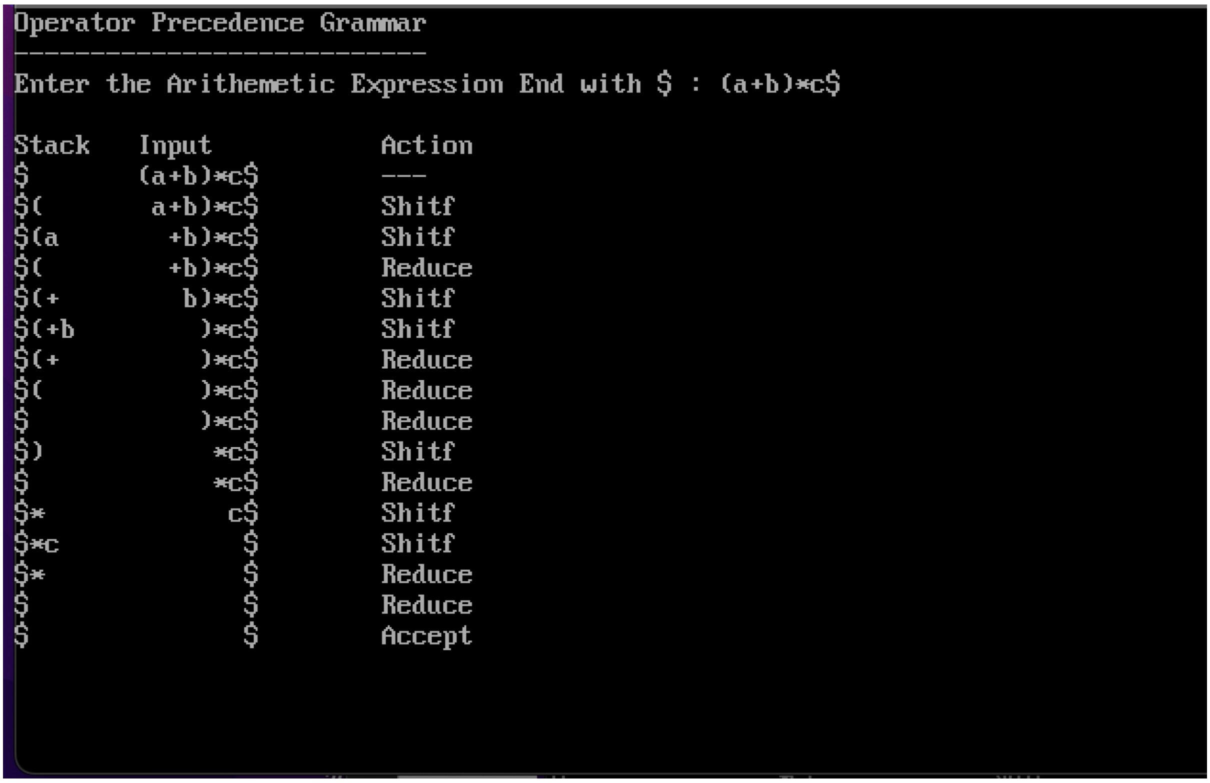
Program:

```
#include<stdio.h>
#include<conio.h>
#include<string.h>
#include<ctype.h>

char q[9][9]={
   {'>','>','<','<','<','<','>','<','>'},
   {'>','>','<','<','<','<','>','<','>'},
   {'>','>','>','>','<','<','>','<','>'},
   {'>','>','>','>','<','<','>','<','>'},
   {'>','>','<','<','<','<','>','<','>'},
   {'<','<','<','<','<','<','=','<','>'},
   {'>','>','>','>','>','E','>','E','>'},
   {'>','>','>','>','>','E','>','E','>'},
   {'<','<','<','<','<','<','<','<','A'}
};
char st[30],ip[30],qr;
int top=-1,p=0,r=0,i;

void push(char c){
   top++;
   st[top]=c;
}
int pop(){
   char a=st[top];
   st[top]=' ';
   top--;
   return a;
}

int find(char c){
   switch (c)
   {
   case '+': return 0;
   case '-': return 1;
   case '*': return 2;
   case '/': return 3;
   case '^': return 4;
   case '(': return 5;
   case ')': return 6;
   case 'a': return 7;
   case '$': return 8;
   default: return -1;
   }
}
```

```
int rel(char a,char b,char d){
   if(isalpha(a)!=0)
    a='a';
   if(isalpha(b)!=0)
    b='a';
   if(q[find(a)][find(b)]==d)
    return 1;
   else
    return 0;
}
void main(){
   clrscr();
   printf("Operator Precedence Grammar\n");
   printf("---------------------------\n");
   printf("Enter the Arithemetic Expression End with $ : ");
   gets(ip);
   push('$');
   printf("\nStack\t\tInput\t\tAction");
   printf("\n%s\t\t%s\t\t---",st,ip);
   i=-1;

   while(i){
      if(ip[p]=='$'&&st[top]=='$'){
         printf("\n%s\t\t%s\t\tAccept",st,ip);
         break;
      }
      else if(rel(st[top],ip[p],'E')){
         printf("\n%s\t\t%s\t\tReject",st,ip);
         break;
      }
      else if(rel(st[top],ip[p],'<')||rel(st[top],ip[p],'=')){
         push(ip[p]);
         ip[p]=' ';
         printf("\n%s\t\t%s\t\tShitf",st,ip);
         p++;
      }
      else if(rel(st[top],ip[p],'>')){
         while(rel(st[top],ip[p],'>')||rel(st[top],ip[p],'=')){
            qr=pop();
            printf("\n%s\t\t%s\t\tReduce",st,ip);
         }
      }
   }
   getch();
}
```

Sample Input & Output:



```
Operator Precedence Grammar
---------------------------
Enter the Arithemetic Expression End with $ : (a+b)*c$

Stack   Input           Action
$       (a+b)*c$        ---
$(       a+b)*c$        Shitf
$(a       +b)*c$        Shitf
$(        +b)*c$        Reduce
$(+        b)*c$        Shitf
$(+b        )*c$        Shitf
$(+         )*c$        Reduce
$(          )*c$        Reduce
$           )*c$        Reduce
$)           *c$        Shitf
$            *c$        Reduce
$*            c$        Shitf
$*c            $        Shitf
$*             $        Reduce
$              $        Reduce
$              $        Accept
```