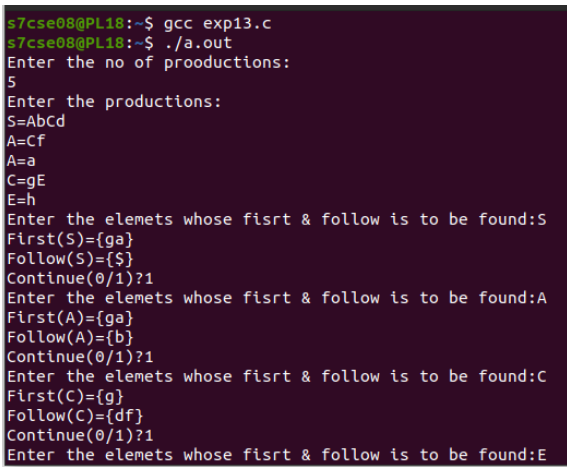
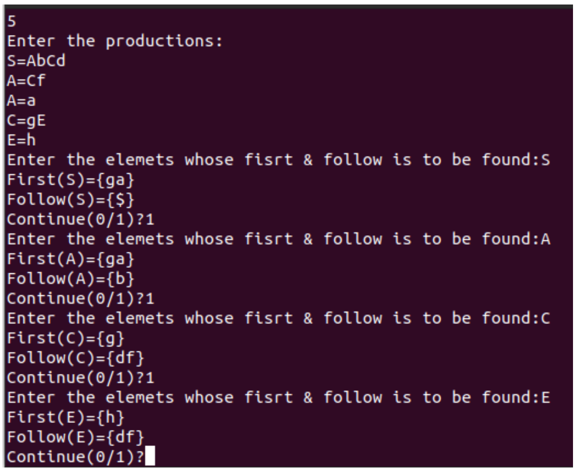
E 13

```
#include<stdio.h>
#include<math.h>
#include<string.h>
#include<ctype.h>
#include<stdlib.h>
int n,m=0,p,i=0,j=0;
char a[10][10],f[10];
void follow(char c);
void first(char c);
int main(){
int i,z;
char c,ch;
//clrscr();
printf("Enter the no of prooductions:\n");
scanf("%d",&n);
printf("Enter the productions:\n");
for(i=0;i<n;i++)
scanf("%s%c",a[i],&ch);
do{
m=0;
printf("Enter the elemets whose fisrt & follow is to be found:");
scanf("%c",&c);
first(c);
printf("First(%c)={",c);
for(i=0;i<m;i++)
printf("%c",f[i]);
printf("}\n");
strcpy(f," ");
//flushall();
m=0;
follow(c);
printf("Follow(%c)={",c);
for(i=0;i<m;i++)
printf("%c",f[i]);
printf("}\n");
printf("Continue(0/1)?");
scanf("%d%c",&z,&ch);
}while(z==1);
return(0);
}
void first(char c)
{
int k;
if(!isupper(c))
f[m++]=c;
for(k=0;k<n;k++)
{
if(a[k][0]==c)
{
if(a[k][2]=='$')
follow(a[k][0]);
else if(islower(a[k][2]))
f[m++]=a[k][2];
else first(a[k][2]);
} } }
void follow(char c)
{
if(a[0][0]==c)
f[m++]='$';
for(i=0;i<n;i++)
{
for(j=2;j<strlen(a[i]);j++)
{
if(a[i][j]==c)
{
if(a[i][j+1]!='\0')
first(a[i][j+1]);
if(a[i][j+1]=='\0' && c!=a[i][0])
follow(a[i][0]);
} } } }
```

OUTPUT



```
s7cse08@PL18:~$ gcc exp13.c
s7cse08@PL18:~$ ./a.out
Enter the no of prooductions:
5
Enter the productions:
S=AbCd
A=Cf
A=a
C=gE
E=h
Enter the elemets whose fisrt & follow is to be found:S
First(S)={ga}
Follow(S)={$}
Continue(0/1)?1
Enter the elemets whose fisrt & follow is to be found:A
First(A)={ga}
Follow(A)={b}
Continue(0/1)?1
Enter the elemets whose fisrt & follow is to be found:C
First(C)={g}
Follow(C)={df}
Continue(0/1)?1
Enter the elemets whose fisrt & follow is to be found:E
```

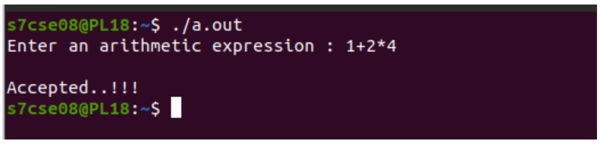


```
5
Enter the productions:
S=AbCd
A=Cf
A=a
C=gE
E=h
Enter the elemets whose fisrt & follow is to be found:S
First(S)={ga}
Follow(S)={$}
Continue(0/1)?1
Enter the elemets whose fisrt & follow is to be found:A
First(A)={ga}
Follow(A)={b}
Continue(0/1)?1
Enter the elemets whose fisrt & follow is to be found:C
First(C)={g}
Follow(C)={df}
Continue(0/1)?1
Enter the elemets whose fisrt & follow is to be found:E
First(E)={h}
Follow(E)={df}
Continue(0/1)?
```

E-14

```
#include<stdio.h>
#include<string.h>
#include<ctype.h>
char input[10];
int i, error;
void E();
void T();
void Eprime();
void Tprime();
void F();
void main() {
  i = 0;
  error = 0;
  printf("Enter an arithmetic expression : ");
  gets(input);
  E();
  if (strlen(input) == i && error == 0)
    printf("\nAccepted..!!!\n");
  else printf("\nRejected..!!!\n");
}
void E() {
  T();
  Eprime();
}
void Eprime() {
  if (input[i] == '+') {
    i++;
    T();
    Eprime();
  } }
void T() {
  F();
  Tprime();
}
void Tprime() {
  if (input[i] == '*') {
    i++;
    F();
    Tprime();
  } }
void F() {
  if (isalnum(input[i])) i++;
  else if (input[i] == '(') {
    i++;
    E();
    if (input[i] == ')')
      i++;
    else error = 1;
  } else error = 1;
}
```

OUTPUT

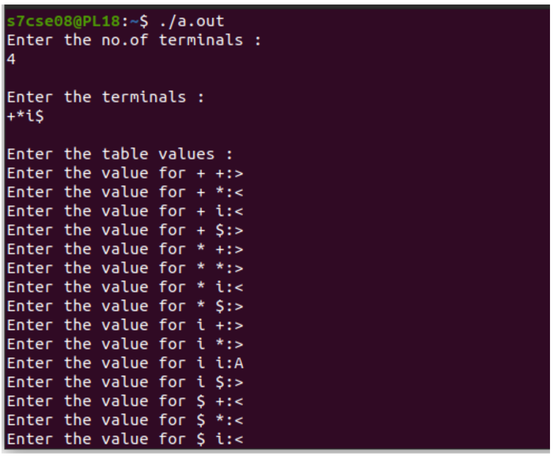


```
s7cse08@PL18:~$ ./a.out
Enter an arithmetic expression : 1+2*4

Accepted..!!!
s7cse08@PL18:~$
```

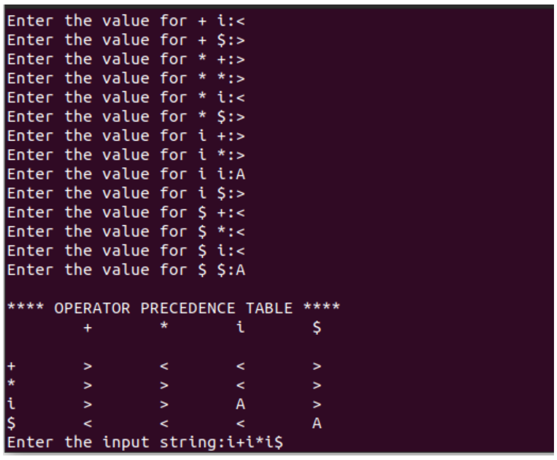
E-15

```
#include<stdio.h>
void main(){
char stack[20],ip[20],opt[10][10][1],ter[10];
int i,j,k,n,top=0,col,row;
for(i=0;i<10;i++)
{
 stack[i]=NULL;
 ip[i]=NULL;
 for(j=0;j<10;j++)
 {
 opt[i][j][1]=NULL;
 } }
printf("Enter the no.of terminals :\n");
scanf("%d",&n);
printf("\nEnter the terminals :\n");
scanf("%s",&ter);
printf("\nEnter the table values :\n");
for(i=0;i<n;i++)
{
 for(j=0;j<n;j++)
 {
 printf("Enter the value for %c %c:",ter[i],ter[j]);
 scanf("%s",opt[i][j]);
 } }
printf("\n**** OPERATOR PRECEDENCE TABLE
****\n");
for(i=0;i<n;i++)
{
printf("\t%c",ter[i]);
}
printf("\n");
for(i=0;i<n;i++){printf("\n%c",ter[i]);
for(j=0;j<n;j++){printf("\t%c",opt[i][j][0]);}}
stack[top]='$';
printf("\nEnter the input string:");
scanf("%s",ip);
i=0;
printf("\nSTACK\t\t\tINPUT
STRING\t\t\tACTION\n");
printf("\n%s\t\t\t%s\t\t\t",stack,ip);
while(i<=strlen(ip))
{
for(k=0;k<n;k++)
{
if(stack[top]==ter[k])
col=k;
if(ip[i]==ter[k])
row=k;
}
if((stack[top]=='$')&&(ip[i]=='$')){
printf("String is accepted\n");
break;}
else if((opt[col][row][0]=='<')
||(opt[col][row][0]=='='))
{ stack[++top]=opt[col][row][0];
stack[++top]=ip[i];
 printf("Shift %c",ip[i]);
 i++;
  }else{
 if(opt[col][row][0]=='>')
{
while(stack[top]!='<'){--top;}
top=top-1;
printf("Reduce");
}
else
{
printf("\nString is not accepted");
break;
} }
printf("\n");
for(k=0;k<=top;k++)
{
printf("%c",stack[k]);
}
printf("\t\t\t");
for(k=i;k<strlen(ip);k++){
printf("%c",ip[k]);
}
printf("\t\t\t");
} }
```

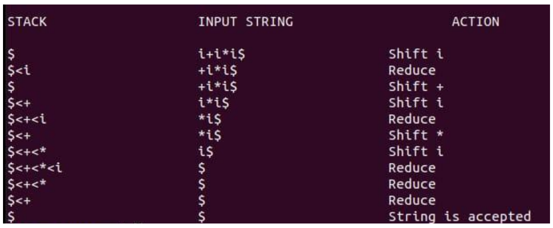
OUTPUT



```
s7cse08@PL18:~$ ./a.out
Enter the no.of terminals :
4

Enter the terminals :
+*i$

Enter the table values :
Enter the value for + +:>
Enter the value for + *:<
Enter the value for + i:<
Enter the value for + $:>
Enter the value for * +:>
Enter the value for * *:>
Enter the value for * i:<
Enter the value for * $:>
Enter the value for i +:>
Enter the value for i *:>
Enter the value for i i:A
Enter the value for i $:>
Enter the value for $ +:<
Enter the value for $ *:<
Enter the value for $ i:<
```



```
Enter the value for + i:<
Enter the value for + $:>
Enter the value for * +:>
Enter the value for * *:>
Enter the value for * i:<
Enter the value for * $:>
Enter the value for i +:>
Enter the value for i *:>
Enter the value for i i:A
Enter the value for i $:>
Enter the value for $ +:<
Enter the value for $ *:<
Enter the value for $ i:<
Enter the value for $ $:A

**** OPERATOR PRECEDENCE TABLE ****
        +       *       i       $

+       >       <       <       >
*       >       >       <       >
i       >       >       A       >
$       <       <       <       A
Enter the input string:i+i*i$
```

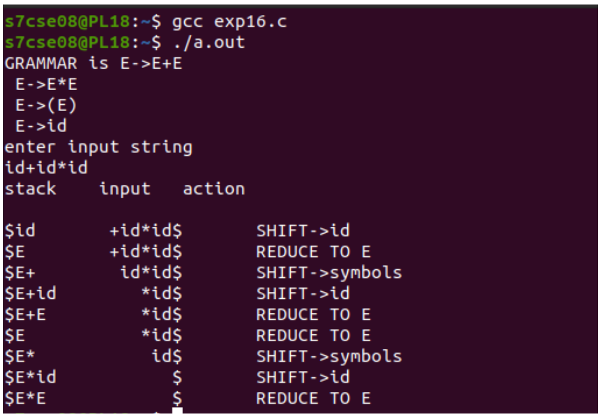


```
STACK                   INPUT STRING                    ACTION

$                       i+i*i$                  Shift i
$<i                     +i*i$                   Reduce
$                       +i*i$                   Shift +
$<+                     i*i$                    Shift i
$<+<i                   *i$                     Reduce
$<+                     *i$                     Shift *
$<+<*                   i$                      Shift i
$<+<*<i                 $                       Reduce
$<+<*                   $                       Reduce
$<+                     $                       Reduce
$                       $                       String is accepted
```

E-16

```
#include<stdio.h>
#include<string.h>
int k=0,z=0,i=0,j=0,c=0;
char a[16],ac[20],stk[15],act[10];
void check();
int main()
  {
     puts("GRAMMAR is E->E+E \n E->E*E \n
E->(E) \n E->id");
     puts("enter input string ");
     scanf("%s",a);
     c=strlen(a);
     strcpy(act,"SHIFT->");
     puts("stack \t input \t action");
     for(k=0,i=0; j<c; k++,i++,j++)
      {
        if(a[j]=='i' && a[j+1]=='d')
          {
             stk[i]=a[j];
             stk[i+1]=a[j+1];
             stk[i+2]='\0';
             a[j]=' ';
             a[j+1]=' ';
             printf("\n$%s\t%s$\t%sid",stk,a,act);
             check();
          }
        else
          {
             stk[i]=a[j];
             stk[i+1]='\0';
             a[j]=' ';

printf("\n$%s\t%s$\t%ssymbols",stk,a,act);
             check();
          }      }  }
void check()
  {
    strcpy(ac,"REDUCE TO E");
    for(z=0; z<c; z++)
      if(stk[z]=='i' && stk[z+1]=='d')
        {
          stk[z]='E';
          stk[z+1]='\0';
          printf("\n$%s\t%s$\t%s",stk,a,ac);
          j++;
        }
     for(z=0; z<c; z++)
if(stk[z]=='E' && stk[z+1]=='+' && stk[z+2]=='E')
         {
           stk[z]='E';
           stk[z+1]='\0';
           stk[z+2]='\0';
           printf("\n$%s\t%s$\t%s",stk,a,ac);
           i=i-2;
         }
     for(z=0; z<c; z++)
       if(stk[z]=='E' && stk[z+1]=='*' &&
stk[z+2]=='E')
         {
           stk[z]='E';
           stk[z+1]='\0';
           stk[z+1]='\0';
           printf("\n$%s\t%s$\t%s",stk,a,ac);
           i=i-2;
         }
     for(z=0; z<c; z++)
       if(stk[z]=='(' && stk[z+1]=='E' && stk[z+2]==')')
         {
           stk[z]='E';
           stk[z+1]='\0';
           stk[z+1]='\0';
           printf("\n$%s\t%s$\t%s",stk,a,ac);
           i=i-2;
         }  }
```

OUTPUT

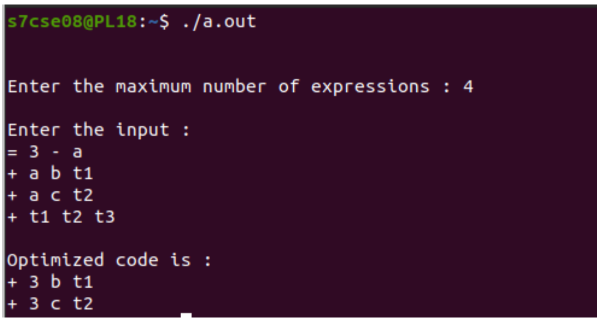


```
s7cse08@PL18:~$ gcc exp16.c
s7cse08@PL18:~$ ./a.out
GRAMMAR is E->E+E
 E->E*E
 E->(E)
 E->id
enter input string
id+id*id
stack    input   action

$id       +id*id$       SHIFT->id
$E        +id*id$       REDUCE TO E
$E+        id*id$       SHIFT->symbols
$E+id        *id$       SHIFT->id
$E+E         *id$       REDUCE TO E
$E           *id$       REDUCE TO E
$E*           id$       SHIFT->symbols
$E*id           $       SHIFT->id
$E*E            $       REDUCE TO E
```

E-17

```
#include<stdio.h>
#include<string.h>
#include<ctype.h>
void input();
void output();
void change(int p,char *res);
void constant();
struct expr{
char op[2],op1[5],op2[5],res[5];
int flag;
}arr[10];
int n;
void main(){
input();
constant();
output();
}
void input(){
int i;
printf("\n\nEnter the maximum number of expressions : ");
scanf("%d",&n);
printf("\nEnter the input : \n");
for(i=0;i<n;i++){
scanf("%s",arr[i].op);
scanf("%s",arr[i].op1);
scanf("%s",arr[i].op2);
scanf("%s",arr[i].res);
arr[i].flag=0;
} }
void constant(){
int i;
int op1,op2,res;
char op,res1[5];
for(i=0;i<n;i++){
if(isdigit(arr[i].op1[0]) && isdigit(arr[i].op2[0]) ||
strcmp(arr[i].op,"=")==0){
/*if both digits, store them in variables*/
op1=atoi(arr[i].op1);
op2=atoi(arr[i].op2);
op=arr[i].op[0];
switch(op){
case '+':
res=op1+op2;
break;
case '-':
res=op1-op2;
break;
case '*':
res=op1*op2;
break;
case '/':
res=op1/op2;
break;
case '=':
res=op1;
break;
}
sprintf(res1,"%d",res);
arr[i].flag=1;
change(i,res1);
} } }
void output(){
int i=0;
printf("\nOptimized code is : ");
for(i=0;i<n;i++){
if(!arr[i].flag)
printf("\n%s %s %s
%s",arr[i].op,arr[i].op1,arr[i].op2,arr[i].res);
} }
void change(int p,char *res){
int i;
for(i=p+1;i<n;i++){
if(strcmp(arr[p].res,arr[i].op1)==0)
strcpy(arr[i].op1,res);
else if(strcmp(arr[p].res,arr[i].op2)==0)
strcpy(arr[i].op2,res);
}
```

OUTPUT



```
s7cse08@PL18:~$ ./a.out


Enter the maximum number of expressions : 4

Enter the input :
= 3 - a
+ a b t1
+ a c t2
+ t1 t2 t3

Optimized code is :
+ 3 b t1
+ 3 c t2
```

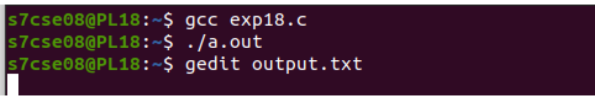
E-18

```
#include<stdio.h>
#include<string.h>
char op[2],arg1[5],arg2[5],result[5];
void main()
{
 FILE *fp1,*fp2;
 fp1=fopen("input.txt","r");
 fp2=fopen("output.txt","w");
 while(!feof(fp1))
 {
  fscanf(fp1,"%s%s%s%s",op,arg1,arg2,result);
  if(strcmp(op,"+")==0)
  {
   fprintf(fp2,"\nMOV R0,%s",arg1);
   fprintf(fp2,"\nADD R0,%s",arg2);
   fprintf(fp2,"\nMOV %s,R0",result);
  }
   if(strcmp(op,"*")==0)
  {
   fprintf(fp2,"\nMOV R0,%s",arg1);
   fprintf(fp2,"\nMUL R0,%s",arg2);
   fprintf(fp2,"\nMOV %s,R0",result);
  }
  if(strcmp(op,"-")==0)
  {
   fprintf(fp2,"\nMOV R0,%s",arg1);
   fprintf(fp2,"\nSUB R0,%s",arg2);
   fprintf(fp2,"\nMOV %s,R0",result);
  }
    if(strcmp(op,"/")==0)
  {
   fprintf(fp2,"\nMOV R0,%s",arg1);
fprintf(fp2,"\nDIV R0,%s",arg2);
   fprintf(fp2,"\nMOV %s,R0",result);
  }
if(strcmp(op,"=")==0)
  {
   fprintf(fp2,"\nMOV R0,%s",arg1);
   fprintf(fp2,"\nMOV %s,R0",result);
  }  }
  fclose(fp1);
  fclose(fp2);
 }
```

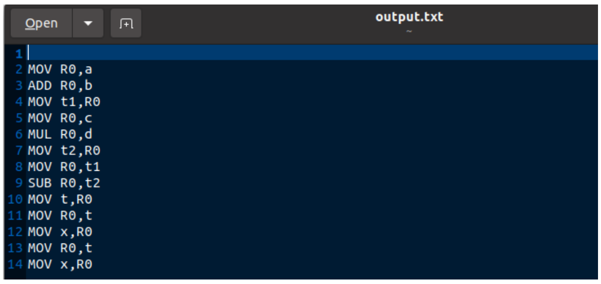
input.txt

```
+ a b t1
* c d t2
- t1 t2 t
= t ? x
```

OUTPUT



```
s7cse08@PL18:~$ gcc exp18.c
s7cse08@PL18:~$ ./a.out
s7cse08@PL18:~$ gedit output.txt
```



output.txt

```

MOV R0,a
ADD R0,b
MOV t1,R0
MOV R0,c
MUL R0,d
MOV t2,R0
MOV R0,t1
SUB R0,t2
MOV t,R0
MOV R0,t
MOV x,R0
MOV R0,t
MOV x,R0
```