

```
C:\Users\Administrator\Documents\symbol table.c - [Executing] - Embarcadero Dev-C++ 6.3
File Edit Search View Project Execute Tools AStyle Window Help
TDM-GCC 9.2.0 64-bit Release
(globals)
Project C left recursion.c left factoring.c symbol table.c Untitled4
1 #include<stdio.h>
2 #include<stdlib.h>
3 #include<string.h>
4 int cnt=0;
5 struct symtab
6 {
7     char label[20];
8     int addr;
9 }
10 sy[50];
11 void insert();
12 int search(char *);
13 void display();
14 void modify();
15 int main()
16 {
17     int ch,val;
18     char lab[10];
19     do
20     {
21         printf("\n1.insert\n2.display\n3.search\n4.modify\n5.exit\n");
22         scanf("%d",&ch);
23         switch(ch)
24         {
25             case 1:
26                 insert();
27                 break;
28             case 2:
29                 display();
30         }
31     } while(ch!=5);
32 }
```

Compiler Resources Compile Log Debug Find Results Console Close

Abort Compilation

Output Filename: C:\Users\Administrator\Documents\symbol table.exe
Output Size: 324.900390625 KiB
Compilation Time: 0.69s

Shorten compiler path

Line: 16 Col: 16 Sel: 2 0 Lines: 103 Length: 1587 Insert Done parsing in

```
C:\Users\Administrator\Documents\symbol table.c - [Executing] - Embarcadero Dev-C++ 6.3
1.insert
2.display
3.search
4.modify
5.exit
1
enter the labela
enter the address100

1.insert
2.display
3.search
4.modify
5.exit
2
a 100

1.insert
2.display
3.search
4.modify
5.exit
3
enter the labela
label is found
1.insert
2.display
3.search
4.modify
5.exit
5

Process exited after 15.72 seconds with return value 0
```

```
C:\Users\Administrator\Documents\left recursion.c - [Executing] - Embarcadero Dev-C++ 6.3
File Edit Search View Project Execute Tools AStyle Window Help
TDM-GCC 9.2.0 64-bit Release
(globals)
Project C left recursion.c
1 #include<stdio.h>
2 #include<string.h>
3 #define SIZE 10
4 int main () {
5     char non_terminal;
6     char beta,alpha;
7     int num;
8     char production[10][SIZE];
9     int index=3;
10    printf("Enter Number of Production : ");
11    scanf("%d",&num);
12    printf("Enter the grammar as E->E-A :\n");
13    for(int i=0;i<num;i++){
14        scanf("%s",production[i]);
15    }
16    for(int i=0;i<num;i++){
17        printf("\nGRAMMAR : : %s",production[i]);
18        non_terminal=production[i][0];
19        if(non_terminal==production[i][index]) {
20            alpha=production[i][index+1];
21            printf(" is left recursive.\n");
22            while(production[i][index]!=0 && production[i][index]!='|')
23                index++;
24            if(production[i][index]!=0) {
25                beta=production[i][index+1];
26                printf("Grammar without left recursion:\n");
27                printf("%c->%c%c'",non_terminal,beta,non_terminal);
28                printf("\n%c'\n->%c%c'|E\n",non_terminal,alpha,non_terminal);
29            }
30        }
31    }
32 }
```

Compiler Resources Compile Log Debug Find Results Console Close

Abort Compilation

Output Filename: C:\Users\Administrator\Documents\left recursion.exe
Output Size: 323.453125 KiB
Compilation Time: 0.64s

Shorten compiler path

Line: 32 Col: 32 Sel: 16 0 Lines: 38 Length: 1427 Insert Done parsing in 0.047 s

```
C:\Users\Administrator\Documents\left recursion.c - [Executing] - Embarcadero Dev-C++ 6.3
Enter Number of Production : 2
Enter the grammar as E->E-A :
S->(L)|a
L->L,S|S

GRAMMAR : : S->(L)|a is not left recursive.

GRAMMAR : : L->L,S|S is left recursive.
Grammar without left recursion:
L->SL'
L'->,L'|E

Process exited after 50.64 seconds with return value 0
Press any key to continue . . .
```

C:\Users\Administrator\Documents\vowels and consonants.cpp - [Executing] - Embarcadero Dev-C++ 6.3

File Edit Search View Project Execute Tools AStyle Window Help

TDM-GCC 9.2.0 64-bit Release

(globals)

Project C < cw.c x subexpression.c x TAC.c x [*] vowels and consonants.cpp x

```
1 #include <stdio.h>
2 #include <ctype.h>
3
4 int main() {
5     char sentence[1000];
6     int vowel_count = 0, consonant_count = 0;
7
8     printf("Enter a sentence: ");
9     fgets(sentence, sizeof(sentence), stdin);
10
11     for (int i = 0; sentence[i] != '\0'; i++) {
12         char ch = tolower(sentence[i]);
13
14         if (isalpha(ch)) {
15             if (ch == 'a' || ch == 'e' || ch == 'i' || ch == 'o' || ch == 'u') {
16                 vowel_count++;
17             } else {
18                 consonant_count++;
19             }
20         }
21     }
22
23     printf("Number of vowels : %d\n", vowel_count);
24     printf("Number of consonants: %d\n", consonant_count);
25
26     return 0;
27 }
28
```

Compiler (1) Resources Compile Log Debug Find Results Console Close

Abort Compilation

Output Filename: C:\Users\Administrator\Documents\vowels and consonants.exe
Output Size: 323.6533203125 KiB
Compilation Time: 0.45s

Shorten compiler path

C:\Users\Administrator\Documents\shift reduce parsing.c - [Executing] - Embarcadero Dev-C++ 6.3

```
C:\Users\Administrator x + - □ x
Enter a sentence: saveetha school of engineer
ing
Number of vowels : 12
Number of consonants: 15

-----
Process exited after 16.56 seconds with return value 0
Press any key to continue . . . |
```

C:\Users\Administrator\Documents\shift reduce parsing.c - [Executing] - Embarcadero Dev-C++ 6.3

File Edit Search View Project Execute Tools AStyle Window Help

TDM-GCC 9.2.0 64-bit Release

(globals)

Project C < shift reduce parsing.c x

```
25 void check()
26 {
27     int flag=0; temp2[0]=stack[ip_ptr]; temp2[1]='\0';
28     if(!strcmp(temp2,"a")||!strcmp(temp2,"b"))
29     {
30         stack[ip_ptr]='E'; if(!strcmp(temp2,"a"))
31         printf("\n %s\t\t%s\t\t\tE->a",stack[ip_sym]); else
32         printf("\n %s\t\t%s\t\t\tE->b",stack[ip_sym]); flag=1;
33     }
34     if(!strcmp(temp2,"+")||strcmp(temp2,"*")||!strcmp(temp2,"/"))
35     {
36         flag=1;
37     }
38     if(!strcmp(stack,"E+E")||!strcmp(stack,"E/E")||!strcmp(stack,"E*E"))
39     {
40         strcpy(stack,"E"); ip_ptr=0; if(!strcmp(stack,"E+E"))
41         printf("\n %s\t\t%s\t\t\tE->E+E",stack[ip_sym]); else
42         if(!strcmp(stack,"E/E"))
43         printf("\n %s\t\t%s\t\t\tE->E/E",stack[ip_sym]); else
44         if(!strcmp(stack,"E*E"))
45         printf("\n %s\t\t%s\t\t\tE->E*E",stack[ip_sym]); else
46         printf("\n %s\t\t%s\t\t\tE->E+E",stack[ip_sym]); flag=1;
47     }
48
49     if(!strcmp(stack,"E")&&ip_ptr==len)
50     {
51         printf("\n %s\t\t%s\t\t\tACCEPT",stack[ip_sym]); getch();
52         exit(0);
53     }
54 }
```

Compiler Resources Compile Log Debug Find Results Console Close

Abort Compilation

Output Filename: C:\Users\Administrator\Documents\shift reduce parsing.exe
Output Size: 325.1630859375 KiB
Compilation Time: 2.27s

Shorten compiler path

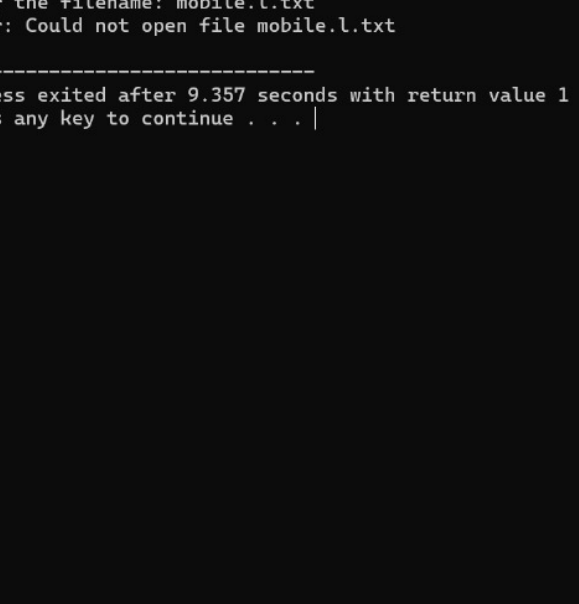
Line: 59 Col: 2 Sel: 0 Lines: 59 Length: 1936 Insert Done parsing in

```
C:\Users\Administrator\Docu x + - □ x
SHIFT REDUCE PARSER

GRAMMER
E->E+E
E->E/E
E->E*E
E->a/b
enter the input symbol: a+b

stack implementation table
stack input symbol action

$ a+b$ --
$a +b$ shift a
$E +b$ E->a
$E+ b$ shift+
$E+b $ shiftb
$E+E $ E->b
$E $ E->E+E
$E $ ACCEPT
```

The screenshot shows a Windows command prompt window with the title bar "C:\Users\Administrator\Docu...". The window contains the following text:

```
Enter the filename: mobile.l.txt
Error: Could not open file mobile.l.txt

-----
Process exited after 9.357 seconds with return value 1
Press any key to continue . . . |
```

```

Enter the string
i*(i+i)*i

STACK   INPUT   ACTION
$i      *(i+i)*i$  Shift
$E      *(i+i)*i$  Reduced: E->i
$E*     (i+i)*i$  Shift
$E*(    (i+i)*i$  Shift
$E*(i   +i)*i$  Shift
$E*(E   +i)*i$  Reduced: E->i
$E*(E+  i)*i$  Shift
$E*(E+i )*i$  Shift
$E*(E+E )*i$  Reduced: E->i
$E*(E   )*i$  Reduced: E->E+E
$E*(E)  *i$      Shift
$E*E    *i$      Reduced: E->)E(
$E      *i$      Reduced: E->E*E
$E*     i$       Shift
$E*i    $        Shift
$E*E    $        Reduced: E->i
$E      $        Reduced: E->E*E
$E$     $        Shift
$E$     $        Shift
Accepted;

-----
Process exited after 24.37 seconds with return value 10
Press any key to continue . . .

```

C:\Users\Administrator\Documents\all languages.c - [Executing] - Embarcadero Dev-C++ 6.3

File Edit Search View Project Execute Tools AStyle Window Help

TDM-GCC 9.2.0 64-bit Release

(globals)

Project C left recursion.c left factoring.c symbol table.c recursive dec

```
1 #include <stdio.h>
2 #include <string.h>
3 int isValid(char str[]) {
4     int i, aCount = 0, bCount = 0;
5     int len = strlen(str);
6     for (i = 0; i < len; i++) {
7         if (str[i] == 'a')
8             aCount++;
9         else
10            break;
11    }
12    for (; i < len; i++) {
13        if (str[i] == 'b')
14            bCount++;
15        else
16            return 0;
17    }
18    return aCount == bCount;
19 }
20 int main() {
21     char input[100];
22     printf("Enter a string: ");
23     scanf("%s", input);
24     if (isValid(input))
25         printf("The string is valid as per the grammar.\n");
26     else
27         printf("The string is NOT valid as per the grammar.\n");
28     return 0;
29 }
```

Compiler (15) Resources Compile Log Debug Find Results Console Close

Abort Compilation

Output Filename: C:\Users\Administrator\Documents\all languages.c
Output Size: 322.970703125 KiB
Compilation Time: 0.64s

Shorten compiler path

Line: 2 Col: 20 Sel: 0 Lines: 29 Length: 687 Insert Done parsing in 0.015 seconds

C:\Users\Administrator\Documents\TAC.c - [Executing] - Embarcadero Dev-C++ 6.3

File Edit Search View Project Execute Tools AStyle Window Help

(globals)

Project C cw.c subexpression.c TAC.c

```
1 #include <stdio.h>
2 #include <string.h>
3 #include <ctype.h>
4 char expr[100];
5 int tempVar = 1;
6 void newTemp(char temp[]) {
7     sprintf(temp, "%d", tempVar++);
8 }
9 int precedence(char op) {
10    if (op == '*' || op == '/') return 2;
11    if (op == '+' || op == '-') return 1;
12    return 0;
13 }
14 char operandStack[100][20];
15 char operatorStack[100];
16 int topOperand = -1, topOperator = -1;
17 void pushOperand(char *val) {
18     strcpy(operandStack[++topOperand], val);
19 }
20 void popOperand(char *val) {
21     strcpy(val, operandStack[topOperand--]);
22 }
23 void pushOperator(char op) {
24     operatorStack[++topOperator] = op;
25 }
26 char popOperator() {
27     return operatorStack[topOperator--];
28 }
29 char peekOperator() {
```

Compiler (1) Resources Compile Log Debug Find Results Console

Abort Compilation

Output Filename: C:\Users\Administrator\Documents\TAC.c
Output Size: 325.1640625 KiB
Compilation Time: 0.23s

Shorten compiler path

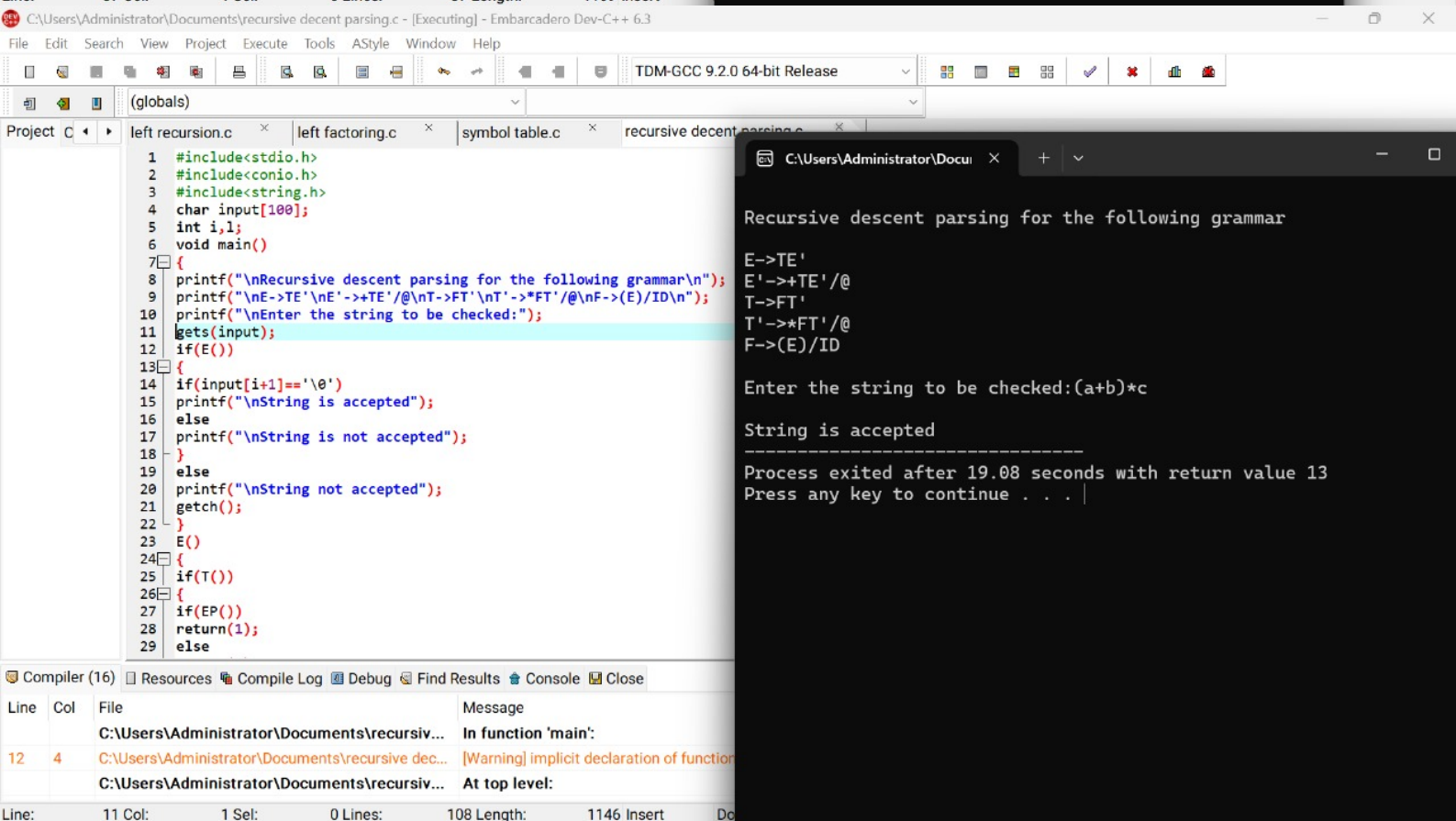
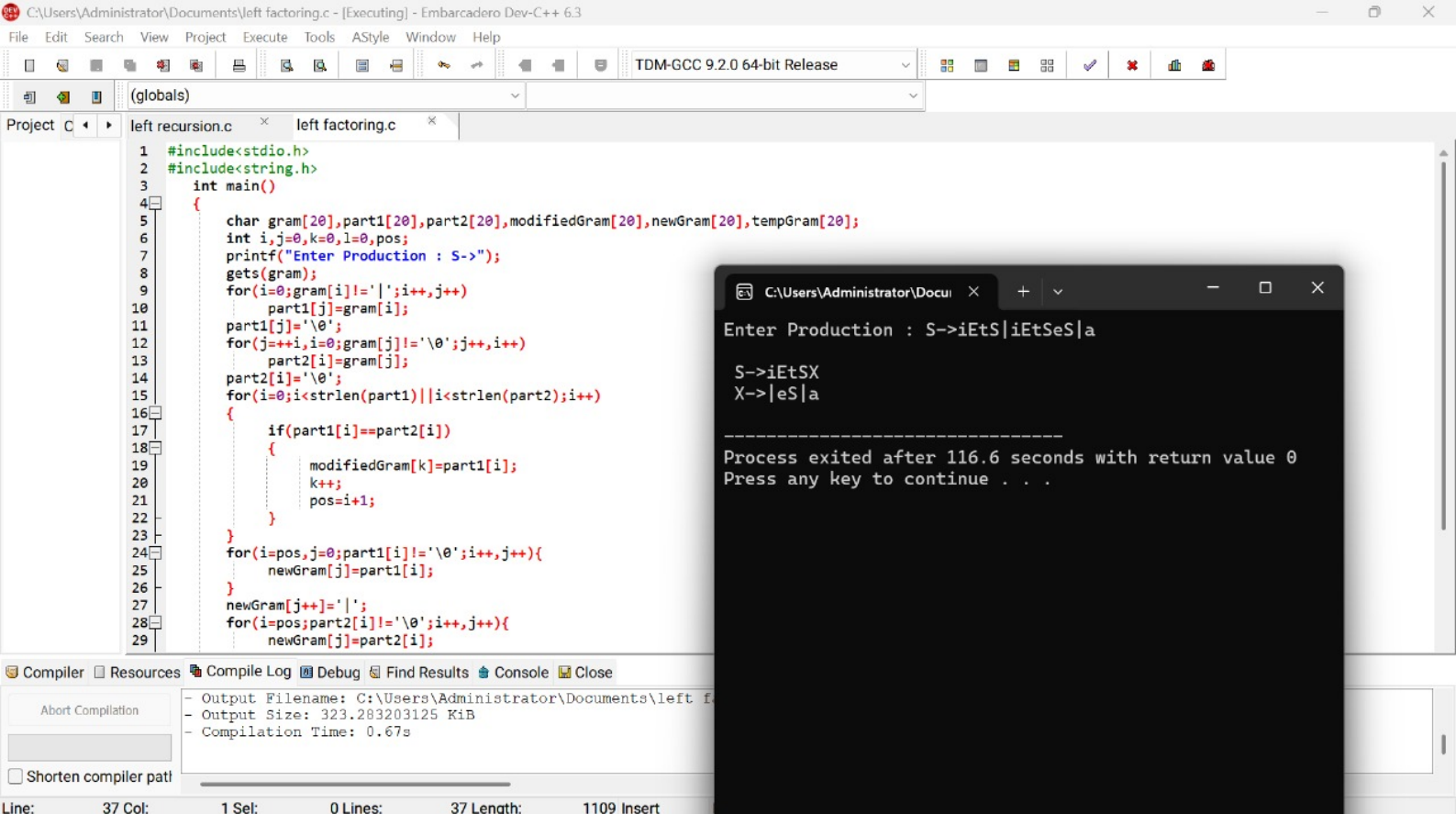
Line: 28 Col: 2 Sel: 0 Lines: 76 Length: 1846 Insert Done parsing in 0.031 seconds

Enter a string: aba
The string is NOT valid as per the grammar.

Process exited after 2.819 seconds with return value 0
Press any key to continue . . .

Enter arithmetic expression (e.g., a+b*c-d): a*b-c+d
t1 = a * b
t2 = t1 - c
t3 = t2 + d
Result stored in: t3

Process exited after 7.854 seconds with return value 0
Press any key to continue . . .




```

(globals)
Project C < cw1.c x subexpression.c x
1 #include <stdio.h>
2 #include <string.h>
3 #define MAX 100
4 typedef struct {
5     char result[10];
6     char arg1[10];
7     char op[3];
8     char arg2[10];
9 } Expression;
10 int is_common(Expression e1, Expression e2) {
11     return (strcmp(e1.arg1, e2.arg1) == 0 &&
12           strcmp(e1.arg2, e2.arg2) == 0 &&
13           strcmp(e1.op, e2.op) == 0);
14 }
15 int main() {
16     Expression expr[MAX], optimized[MAX];
17     int n, i, j, k = 0, found;
18     printf("Enter the number of expressions: ");
19     scanf("%d", &n);
20     printf("Enter expressions in the form: result = arg1 op arg2\n");
21     for (i = 0; i < n; i++) {
22         printf("Expression %d: ", i + 1);
23         scanf("%s = %s %s %s", expr[i].result, expr[i].arg1, expr[i].op, expr[i].arg2);
24     }
25     for (i = 0; i < n; i++) {
26         found = 0;
27         for (j = 0; j < k; j++) {
28             if (is_common(expr[i], optimized[j])) {
29                 printf("Optimized: %s = %s\n", expr[i].result, optimized[j].result);

```

Compiler (1) Resources Compile Log Debug Find Results Console Close

Abort Compilation

- Output Filename: C:\Users\Administrator\Documents\subexpression.exe
 - Output Size: 324.1513671875 KiB
 - Compilation Time: 0.22s

Shorten compiler path

Line: 24 Col: 6 Sel: 0 Lines: 48 Length: 1462 Insert Done parsing in 0.046 seconds

C:\Users\Administrator\Documents\identifier,operators.c - [Executing] - Embarcadero Dev-C++ 6.3

```

(globals)
Project C < identifier,operators.c x
1 #include <stdio.h>
2 #include <ctype.h>
3 #include <string.h>
4 int main()
5 {
6     int i, ic=0, m, cc=0, oc=0, j;
7     char b[30], operators[30], identifiers[30], constants[30];
8     printf("enter the string : ");
9     scanf("%[^\n]s", &b);
10    for(i=0; i<strlen(b); i++)
11    {
12        if(isspace(b[i]))
13        {
14            continue;
15        }
16        else if(isalpha(b[i]))
17        {
18            identifiers[ic] = b[i];
19            ic++;
20        }
21        else if(isdigit(b[i]))
22        {
23            m=(b[i]-'0');
24            i=i+1;
25            while(isdigit(b[i]))
26            {
27                m=m*10 + (b[i]-'0');
28                i++;
29            }

```

Compiler Resources Compile Log Debug Find Results Console Close

Abort Compilation

- Output Filename: C:\Users\Administrator\Documents\id
 - Output Size: 323.650390625 KiB
 - Compilation Time: 0.69s

Shorten compiler path

Line: 74 Col: 1 Sel: 0 Lines: 74 Length: 1389 Insert Done parsing in 0.609 seconds

```

C:\Users\Administrator x + - □ x
Enter the number of expressions: 3
Enter expressions in the form: result = arg1
op arg2
Expression 1: t1=a+b
t2=a+b
Expression 2: t3=t1
Expression 3: t3=t1+c
Kept : t1=a+b = eC
Kept : t2=a+b =
Kept : t3=t1 =

Final Optimized Code:
t1=a+b = eC
t2=a+b =
t3=t1 =

-----
Process exited after 55.01 seconds with return
value 0
Press any key to continue . . .

```

```

C:\Users\Administrator\Docu x + -
enter the string : a+7*b/4
identifiers : a b
constants : 7 4
operators : + *

-----
Process exited after 14.37 seconds with return value 0
Press any key to continue . . .

```

TDM-GCC 9.2.0 64-bit Release

(globals)

Project C

identifier,operators.c × commentbegin.c ×

```

1 #include <stdio.h>
2 #include <string.h>
3 #include <stdbool.h>
4 bool isSingleLineComment(const char *line) {
5     return (strcmp(line, "//") == 0);
6 }
7 bool isMultiLineComment(const char *line) {
8     int len = strlen(line);
9     return (len >= 4 && strstr(line, "/*") != NULL && strstr(line, "*/") != NULL);
10 }
11 int main() {
12     char line[1000];
13     printf("Enter a line: ");
14     fgets(line, sizeof(line), stdin);
15     line[strcspn(line, "\n")] = '\0';
16     if (isSingleLineComment(line)) {
17         printf("This is a single-line comment.\n");
18     }
19     else if (isMultiLineComment(line)) {
20         printf("This is a multi-line comment.\n");
21     }
22     else {
23         printf("This is not a comment.\n");
24     }
25     return 0;
26 }

```

Compiler Resources Compile Log Debug Find Results Console Close

Abort Compilation

Shorten compiler path

Output Filename: C:\Users\Administrator\Documents\commentbegin.exe
Output Size: 323.869140625 KiB
Compilation Time: 0.66s

Line: 26 Col: 2 Sel: 0 Lines: 26 Length: 740 Insert Done parsing in 0.062 seconds

C:\Users\Administrator\Docu × + - □ ×

Enter a line: //comment line
This is a single-line comment.

Process exited after 41.97 seconds with return value 0
Press any key to continue . . .

TDM-GCC 9.2.0 64-bit Release

(globals)

Project C

shift reduce parsing.c × operator precedence parsing.c × backend of the compiler.c ×

```

1 #include<stdio.h>
2 #include<conio.h>
3 #include<string.h>
4 int main()
5 {
6     int n,i,j;
7     char a[50][50];
8     printf("enter the no: intermediate code:");
9     scanf("%d",&n);
10    for(i=0;i<n;i++)
11    {
12        printf("enter the 3 address code:%d:",i+1);
13        for(j=0;j<6;j++)
14        {
15            scanf("%c",&a[i][j]);
16        }
17    }
18    printf("the generated code is:");
19    for(i=0;i<n;i++)
20    {
21        printf("\n mov %c,R%d",a[i][3],i);
22        if(a[i][4]=='-')
23        {
24            printf("\n sub %c,R%d",a[i][5],i);
25        }
26        if(a[i][4]=='+')
27        {
28            printf("\n add %c,R%d",a[i][5],i);
29        }
30    }

```

Compiler (3) Resources Compile Log Debug Find Results Console Close

Abort Compilation

Shorten compiler path

Output Filename: C:\Users\Administrator\Docume
Output Size: 323.9814453125 KiB
Compilation Time: 0.59s

Line: 16 Col: 10 Sel: 0 Lines: 42 Length: 743

C:\Users\Administrator\Docu × + - □ ×

enter the no: intermediate code:2
enter the 3 address code:1:a=b+c
enter the 3 address code:2:d=n*d
the generated code is:
mov b,R0
add c,R0
mov R0,a

mov n,R1
mul d,R1
mov R1,d

Process exited after 30.9 seconds with return value 0
Press any key to continue . . .