

OPERATOR PRECEDENCE GRAMMAR

```
#include <stdio.h>
#include <string.h>
void main()
{
    char g[20][20], c;
    int i, n, j = 2, flag = 0;
    scanf("%d", &n);
    for (i = 0; i < n; i++)
        scanf("%s", g[i]);

    for (i = 0; i < n; i++) {
        c = g[i][2];
        while (c != '\0') {
            if (g[i][3] == '+' || g[i][3] == '-' || g[i][3] == '*' || g[i][3] == '/')
                flag = 1;
            else {
                flag = 0;
                printf("Operator Precedence not satisfied");
                break;
            }

            if (c == '$') {
                flag = 0;
                printf("Operator Precedence not satisfied");
                break;
            }

            c = g[i][++j];
        }
    }

    if (flag == 1)
        printf("Grammar is operator precedence grammar");
}
```

Activities Terminal Thu 06:24

operator.c

```
1#include <stdio.h>
2#include <string.h>
3void main()
4{
5    char g[20][20], c;
6    int i, n, j = 2, flag = 0;
7    scanf("%d", &n);
8    for (i = 0; i < n; i++)
9        scanf("%s", g[i]);
10
11    for (i = 0; i < n; i++) {
12        c = g[i][2];
13        while (c != '\0') {
14            if (g[i][3] == '+' || g[i][3] == '-') {
15                flag = 1;
16            }
17            else {
18                flag = 0;
19                printf("Operator Precedence not satisfied\n");
20                break;
21            }
22            if (c == '$') {
23                flag = 0;
24                printf("Operator Precedence not satisfied\n");
25                break;
26            }
27            c = g[i][++j];
28        }
29    }
30
31    if (flag == 1)
32        printf("Grammar is operator precedence grammar\n");
33    else
34        printf("Grammar is not operator precedence grammar\n");
35}
```

glau@localhost:~

```
2
A=A+B
B=AA
Not operator grammar[glau@localhost ~]$ gedit operator.c
[glau@localhost ~]$ gcc operator.c
[glau@localhost ~]$ ./a.out
A-A/B
[glau@localhost ~]$ ./a.out
2
A-A+C
C.CC
Operator Precedence not satisfied[glau@localhost ~]$ gedit operator.c
[glau@localhost ~]$ gcc operator.c
[glau@localhost ~]$ ./a.out
2
A-A+A
B-BA
[glau@localhost ~]$ ./a.out
2
A-$
S-S+A
Operator Precedence not satisfiedGrammar is operator precedence grammar[glau@localhost ~]$ gedit operator.c
```

C Tab Width: 8 Ln 1, Col 1 INS