

$E \rightarrow E + T / T, T \rightarrow T * F / F, F \rightarrow (E) / id$ and parse the sentence
 $id + id * id$

```
#include <stdio.h>
```

```
char st[50];
```

```
char a[];
```

```
int len;
```

```
int i=0;
```

```
int top=-1;
```

```
void check() {
```

```
    int j=top;
```

```
    if(st[j] == 'T' && st[j-1] == '+' && st[j-2] == 'E')
```

```
    { if(a[i+1] == '*') {
```

```
        printf("%s\t%s\t" Do not Remove E+T to E\n", st, a);
```

```
        return;
```

```
    }
```

```
else {
```

```
    st[j] = ' ';
```

```
    st[j-1] = ' ';
```

```
    top = top-2;
```

Marks

Staff Signature

```

j = top;
printf("%s\t%s\t Reduce E+T to E\n", st, a);
check();
}

```

```

if (st[j] == 'T') {
    st[j] = 'E';
    printf("%s\t%s\t Reduce T to E\n", st, a);
    check();
}

```

```

if (st[j] == 'F' && st[j-1] == '*' && st[j-2] == 'T') {
    st[j] = '';
    st[j-1] = '';
    top = top - 2;
    j = top;
    printf("%s\t%s\t Reduce T * F to T\n", st, a);
    check();
}

```

```

if (st[j] == 'F') {
    st[j] = 'T';
    printf("%s\t%s\t Reduce F to T\n", st, a);
    check();
}

```

```

if (st[j] == 'C' && st[j-1] == 'E' && st[j-2] == 'C')
{
    st[j] = '';
    st[j-1] = '';
    st[j-2] = 'E';
    top = top - 2;
}

```



```

j = top;
printf("%s\t%s\t Reduce id to F\n", st, a);
check();
}
return;

```

```

void main() {
    printf("Enter the expression\n");
    gets(a);
    len = strlen(a);
    printf("STACK \t EXP \t ACTION\n");
    for(i = 0; i < len; i++) {
        if(a[i] == 'i' && a[i+1] == 'd') {
            st[++top] = 'i';
            st[++top] = 'd';
            a[i] = ' ';
            a[i+1] = ' ';
            i++;
            printf("%s\t%s\t SHIFT\n", st, a);
            check();
            continue;
        }
        else {
            st[++top] = a[i];
            a[i] = ' ';
            printf("%s\t%s\t SHIFT\n", st, a);
            check();
        }
    }
    return;
}

```

Output:

Enter the expression : $id + id * id$

STACK	EXP	ACTION
id	$+ id * id$	SHIFT
F	$+ id * id$	REDUCE id to F
T	$+ id * id$	REDUCE F to T
E	$+ id * id$	REDUCE T to E
E+	$id * id$	SHIFT
E+id	$* id$	SHIFT
E+ F	$* id$	REDUCE id to F
E+T	$* id$	REDUCE F to T
E+T	$* id$	DO NOT REDUCE E+T to E
E+T*	id	SHIFT
E+T*id		SHIFT
E+T*F		REDUCE id to F
E+ T		REDUCE T*F to T
E		REDUCE E+T to E