

```
// IED-001 (Prof. Dr. Silvio do Lago Pereira)
```

```
// -----  
// Exemplo 4  
// -----
```

```
#include <stdio.h>  
#include <ctype.h>  
#include <string.h>  
#include "pilha.h"
```

```
char *posfixa(char *e) {  
    static char s[256];  
    int j = 0;  
    Pilha P = pilha(256);  
    for(int i=0; e[i]; i++)  
        if( isdigit(e[i]) ) s[j++] = e[i];  
        else if( strchr("+-/",e[i]) ) empilha(e[i],P);  
        else if( e[i]=='(' ) s[j++] = desempilha(P);  
    s[j] = '\0';  
    destroip(&P);  
    return s;  
}
```

```
int main(void) {  
    char e[513];  
    printf("Infixa? ");  
    gets(e);  
    printf("Posfixa: %s\n", posfixa(e));  
    return 0;  
}
```

```
// -----  
// Exemplo 6  
// -----
```

```
int prio(char o) {  
    switch( o ) {  
        case '(': return 0;  
        case '+':  
        case '-': return 1;  
        case '*':  
        case '/': return 2;  
    }  
    return -1; // operador inválido!  
}
```

```
// -----  
// Exemplo 7  
// -----
```

```
char *posfixa(char *e) {  
    static char s[256];  
    int j = 0;  
    Pilha P = pilha(256);  
    for(int i=0; e[i]; i++)  
        if( e[i]=='(' ) empilha('(',P);  
        else if( isdigit(e[i]) ) s[j++] = e[i];  
        else if( strchr("+-/*",e[i]) ) {  
            while( !vaziap(P) && prio(topo(P))>=prio(e[i]) )  
                s[j++] = desempilha(P);  
            empilha(e[i],P);  
        }  
        else if( e[i] == ')' ) {  
            while( topo(P)!='(' )  
                s[j++] = desempilha(P);  
        }  
    }  
    s[j] = '\0';  
    destroip(&P);  
    return s;  
}
```

```
        desempilha(P);
    }
    while( !vaziap(P) )
        s[j++] = desempilha(P);
    s[j] = '\0';
    destroip(&P);
    return s;
}
```

```
// -----
// Exemplo 8
// -----
```

```
int valor(char *e) {
    Pilha P = pilha(256);
    for(int i=0; e[i]; i++)
        if( isdigit(e[i]) ) empilha(e[i]-'0',P);
        else {
            int y = desempilha(P);
            int x = desempilha(P);
            switch( e[i] ) {
                case '+': empilha(x+y,P); break;
                case '-': empilha(x-y,P); break;
                case '*': empilha(x*y,P); break;
                case '/': empilha(x/y,P); break;
            }
        }
    int z = desempilha(P);
    destroip(&P);
    return z;
}
```