

15- O Pré-Processador C

- O compilador C possui um programa chamado pré-processador: executa modificações no código baseado em instruções chamadas de DIRETIVAS.

- Uso: Todas as Diretivas do pré-processador são iniciadas com # (cerquilha).

- Podem ser colocadas em qualquer parte do programa, mas para melhor organização é bom colocar no início, antes de main() ou de uma função particular.

Diretivas mais usadas:

#define ⇒ substitui o código (MACROS) por uma sequência de caracteres.

#include ⇒ inclui o conteúdo de um arquivo em outro durante compilação.

15.1 - Definição de MACROS:

Diretiva #define

A macro mais simples é da forma:

Ex1:

```
#define SIM 1
```

equivale a definição de constantes;

a palavra SIM é substituída por 1 toda vez que é utilizada no programa.

Ex5:

```
#define PI 3.14159
#define AREA(x) 4*PI*x*x
main( )
{ float raio;
.
.
.
printf ("AREA = % .2f", AREA(raio));
}
```

Ex6:

```
#define MAX(A,B) ((A) > (B) ? (A) : (B))
```

Se no meio do programa aparecer:

```
MAX (p + q, r + s);
```

Será substituído por:

```
x = ((p + q) > (r + s) ? (p + q) : (r + s));
```

Ex7: Cuidados ao escrever macros!!

Versão 1:

```
#define soma(x, y) x + y
```

no meio do programa:

```
ans = 10*soma(3, 4);
```

Vai ser feita a seguinte substituição:

```
ans = 10*3+4; /* resulta 34 ERRADO! */
```

solução: parênteses!!!

Ex9: Seja o exemplo 6:

```
#define max(A, B) ((A) > (B) ? (A) : (B))
```

no programa:

```
x = max(i ++, j ++);
```

Será substituído por:

```
x = ((i ++ ) > (j ++ ) ? (i ++ ) : (j ++ ));
```

problema:

↓ ↓ ↓ ↓

(incr. i) (incr. j) (incr. de novo um dos dois)

Supor:

| | | |
|---|---|---|
| x | i | j |
|---|---|---|

lixo 8 5 → antes do comando

10 10 6 → depois do comando

15.3 Diretiva `#include`

- Causa a inclusão de um programa fonte em outro.
- Facilita a manipulação de grupos de `#defines` (macros) e declarações, dentre outras coisas.
- Um arquivo incluído pode ter linhas com `#include` dentro dele.

Exemplo 1: Seja um programa que usa várias macros relacionadas com cálculo de área:

```
#define PI 3.14159
#define AREA_CIRC(r) (PI*r*r)
#define AREA_RET(b, h) (b*h)
#define AREA_TRI(b, h) (b*h/2)
```

Gravar este código (todos os quatro comandos) em: `areas.h` (arquivo).

O programa que vai usar essas macros deve incluir o arquivo `areas.h`, ou seja, deve ter a diretiva:

```
#include <areas.h>
```

Usamos muitas vezes `<stdio.h>`. Dentro de `<stdio.h>` temos diversas definições, como por exemplo:

```
#define getchar( ) getc(STDIN)
#define putchar(c) putc((c), STDOUT)
```


Exemplo 2:

```
#ifndef cores          /* se cores está definida, então: */
#include "cores.h"
#define TAB 10
#else                  /* senão, */
#include "cores1.h"
#define TAB 20
#endif                /* finaliza ifndef */
```

Exemplo 3:

```
#if SYS == "IBM"      /* deixa o sistema mais portátil */
#include <ibm.h>
#endif
```

Obs.: Todo #if e #ifndef deve ser terminado com um #endif.