

# Pré-processador do C

## A diretiva #include

### Objetivo

- Permite inserir o conteúdo do ficheiro '**file**' no ficheiro que está a ser processado, exatamente no local onde ocorre a diretiva

### Formas gerais

#### #include <file>

- o ficheiro *file* é procurado nas bibliotecas standard do C, que variam de computador para computador

#### #include "file"

- o ficheiro *file* é procurado na diretoria onde se encontra o ficheiro que está a ser processado (ficheiros do programador)

## A diretiva #define

### Objetivos

- Permite criar constantes simbólicas (constantes representadas por símbolos)
- Permite criar macros (operações representadas por símbolos)

### Forma geral

```
#define identificador texto_de_substituição
```

### Descrição

- O pré-processador substitui todas as ocorrências no corpo do programa de **identificador** por **texto\_de\_substituição**, antes da compilação do ficheiro
- Na construção da macros usar o símbolo \ para mudar de linha
- O alcance de uma diretiva **#define** é desde a sua definição até ao fim do ficheiro onde está ou até à diretiva **#undef**

## Exemplo

- Construção

```
#define PI 3.141592 // definir a constante PI = 3.141592
#define Area_Circulo (x) ((PI) * (x) * (x))
#define Maior (x, y) ((x) > (y) ? (x) : (y))
```

- Utilização/chamada

```
area = Area_Circulo (4);
    // area = ((3.141592) * (4) * (4))
area = Area_Circulo (c+2);
    // area = ((3.141592) * (c+2) * (c+2))
```

## A diretiva #undef

### Objetivo

- Permite anular uma diretiva definida antes com "#define"

### Forma geral

```
#undef identificador
```

**identificador** foi definido antes com a diretiva "**#define**"

## As diretivas #ifdef, #ifndef, #else e #endif

### Objetivo

- Permite verificar se uma diretiva definida com "**#define**" existe

### Sintaxe (completa)

```
#ifdef [ifndef] identificador  
    comandos_1  
#else  
    comandos_2  
#endif
```

### Sintaxe (reduzida)

```
#ifdef [ifndef] identificador  
    comandos  
#endif
```

## Exemplo 1

```
#include <stdio.h>
#define ANO 12
int main()
{
    #ifdef ANO
        printf("ANO = %d.\n", ANO);
    #endif
    #undef ANO
    #ifndef ANO
        printf("ANO sem valor\n");
    #else
        printf("ANO = %d.\n", ANO);
    #endif
}
```

## Exemplo 2

```
#ifndef BIB2_H  
#define BIB2_H  
int soma2maioresInt (int, int, int);  
float soma2maiores (float, float, float);  
#endif // BIB2_H
```

## As diretivas #if e #elif

### Objetivo

- Permite verificar se uma dada expressão condicional é verdadeira

### Sintaxe (completa)

```
#if expressão_condicional_1  
    comandos_1  
#elif expressão_condicional_2  
    comandos_2  
#endif
```

### Sintaxe (simples)

```
#if expressão_condicional  
    comandos  
#endif
```

## Exemplo

```
#include <stdio.h>
#define ANO 8
int main()
{
    #if ANO < 5
        printf("ANO = %d menor que 5.\n", ANO);
    #elif ANO > 10
        printf("ANO = %d maior que 10.\n", ANO);
    #else
        printf("ANO = %d entre 5 e 10.\n", ANO);
    #endif
}
```

## Macros predefinidas

### Algumas destas macros

**\_\_DATE\_\_**

- string que contém a data atual

**\_\_FILE\_\_**

- string com o nome do ficheiro ".c" que contém o main (programa principal)

**\_\_LINE\_\_**

- inteiro que representa a linha com da instrução

**\_\_TIME\_\_**

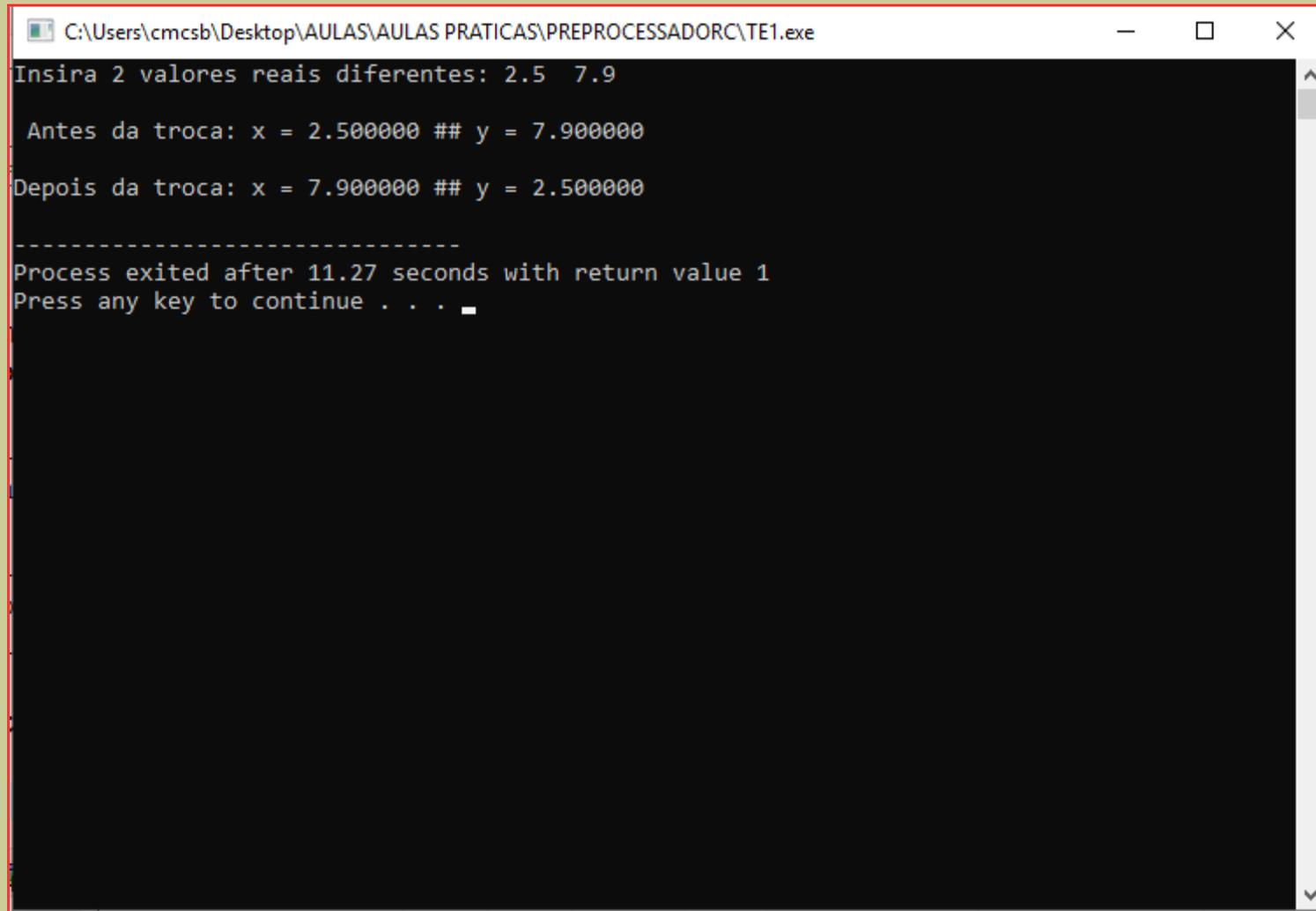
- string que contém a hora atual

## Exemplo (ex. 8 das folhas práticas – código)

```
TE1.c
1  #include <stdio.h>
2
3  #define trocar(x, y) \
4      float a = y; \
5      y = x; \
6      x = a
7
8  int main () {
9      float x, y;
10
11     printf("Insira 2 valores reais diferentes: ");
12     scanf("%f%f", &x, &y);
13
14     printf("\n Antes da troca: x = %f ## y = %f\n", x, y);
15     trocar(x, y);
16     printf("\nDepois da troca: x = %f ## y = %f\n", x, y);
17
18     return 1;
19 }
```

#define trocar(x, y) float a = y; y = x; x = a

## Exemplo (ex. 8 das folhas práticas – resultado da execução)



```
C:\Users\cmcsb\Desktop\AULAS\AULAS PRATICAS\PREPROCESSADORC\TE1.exe
Insira 2 valores reais diferentes: 2.5 7.9

Antes da troca: x = 2.500000 ## y = 7.900000

Depois da troca: x = 7.900000 ## y = 2.500000

-----
Process exited after 11.27 seconds with return value 1
Press any key to continue . . .
```