

# Tipos enumerados

Enumerações, ou tipos enumerados, são tipos de dados definidos pelo programador. Uma variável de tipo enumerado pode assumir um valor dentre um conjunto fixo de valores possíveis previamente definido.

Um tipo enumerado é visto como um caso especial de tipo inteiro. Variáveis de tipos enumerados são implementadas pelo compilador usando valores inteiros, de uma forma que é invisível ao programador.

Um exemplo simples de declaração de tipos enumerados é

```
// definição do tipo enumerado
enum Color_t {
    BLACK,
    BLUE,
    GREEN,
    RED,
    YELLOW,
    WHITE
} ;

// declaração de variável do tipo enum Color_t
enum Color_t c1;
```

A definição do tipo enumerado também pode ser feita usando "typedef"

```
typedef enum {
    BLACK,
    BLUE,
    GREEN,
    RED,
    YELLOW,
    WHITE
} Color_t ;

Color_t c1;
```

Variáveis enumeradas podem ser usadas como variáveis inteiras

```
Color_t c1;

// atribuição
c1 = GREEN ;

// uso
if (c1 == RED) { ... }

// os valores enumerados têm uma ordem definida
if (c1 >= BLUE && c1 <= YELLOW) { ... }

// variáveis enumeradas podem ser incrementadas
c1 = BLACK ;
while (c1 <= WHITE)
{
    ...
    c1++ ;
}
```

Por default, o primeiro valor de um tipo enumerado é 0 (zero), o segundo valor é 1 e assim por diante. Esses valores podem ser alterados se isso for conveniente.

```
typedef enum {
    BLACK = 1,
```

```
BLUE,  
GREEN,  
RED = 10,  
YELLOW,  
WHITE  
} Color_t ;
```

Os valores inteiros associados à enumeração são:

valor simbólico	valor 'normal'	valor com atribuição
BLACK	0	1
BLUE	1	2
GREEN	2	3
RED	3	10
YELLOW	4	11
WHITE	5	12

Tipos enumerados podem ser usados para tornar o código mais legível e também diminuir erros devidos a valores imprevistos em variáveis. Um exemplo clássico de uso de tipos enumerados é a definição de um tipo booleano em C:

```
typedef enum  
{  
    FALSE = 0,  
    false = 0,  
    TRUE = 1,  
    true = 1  
} bool_t ;  
  
bool_t flag ;
```

## Exercícios

1. Escreva um programa em C para classificar os alunos de uma determinada matéria de acordo com sua situação. O programa deve:
  - ler os dados do aluno referentes a sua nota na disciplina e sua frequência;
  - classificar o aluno pelas enumerações "Aprovado", "Exame Final" ou "Reprovado" de acordo com a nota e frequência.
2. Escreva um programa em C que recebe um número e o classifica, por meio de tipos enumerados, entre número par ou ímpar; positivo, negativo ou zero; múltiplo de 10 ou não múltiplo.
3. Escreva um programa em C que lê uma determinada sequência de caracteres e os classifique como sendo "letra", "pontuação" ou "caractere especial" e guarde os dados em uma estrutura. Para o caso de letras, o programa deve ainda classificá-las, por meio de enumerações, entre "maiúsculas" e "minúsculas"; "vogais" e "consoantes".