
LABORATÓRIO 13

UNIÕES E ENUMERAÇÕES

EXERCÍCIOS DE REVISÃO

VOCÊ DEVE ACOMPANHAR PARA OBTER INFORMAÇÕES COMPLEMENTARES

1. Considerando as declarações abaixo, responda dizendo qual o **tipo** da informação que deve estar armazenada na variável:

```
enum direcao {N, NE, E, SE, S, SW, W, NW};

union identificador
{
    char codigo[5];
    int numero;
};

struct bola
{
    identificador id; // identificação da bola
    char marca[30];   // marca da bola
    int posX;         // posição no eixo X
    int posY;         // posição no eixo Y
    direcao dir;      // orientação no plano
    float vel;        // velocidade
    float acel;       // aceleração
};

bola proVolley[10]; // declaração de variável
```

- | | |
|--|---|
| a) proVolley vetor de bola | e) proVolley[0].dir enum direcao |
| b) proVolley[4] bola | f) proVolley[4].id union identificador |
| c) proVolley[1].marca vetor de char | g) proVolley[3].id.numero int |
| d) proVolley[1].marca[2] char | h) proVolley[9].acel float |

EXERCÍCIOS DE FIXAÇÃO

VOCÊ DEVE FAZER OS EXERCÍCIOS PARA FIXAR O CONTEÚDO

1. Uma fábrica de automóveis define a cor de um carro por um número, um código de letras ou pelo nome da cor, nunca os três simultaneamente. A correspondência das principais cores é dada na tabela abaixo. Construa um registro para guardar informações sobre um carro. Um carro deve ter nome do modelo, ano de fabricação, cor (número, código ou nome) e preço.

Nome	Número	Código
Preto	13302	A33B
Azul	00828	C44E
Amarelo	11029	JB81

Em seguida construa um vetor de 10 carros inicializando os dois primeiros carros com as informações: "Vectra", 2009, "Azul", R\$58.000,00 e "Polo", 2008, "Preto", R\$45.000,00, respectivamente. Em seguida peça ao usuário para entrar com os dados de um terceiro carro e armazene-o no vetor. Finalmente o programa deve mostrar na tela os dados para os três carros.

2. Considerando o programa abaixo, modifique-o para que o tipo "mês" possa ser usado com cin e cout.

```
#include <iostream>
using namespace std;

enum mes {Jan=1,Fev,Mar,Abr,Mai,Jun,Jul,Ago,Set,Out,Nov,Dez};

int main()
{
    mes inicio, fim;

    inicio = Mar;    // inicio do semestre
    fim     = Jun;    // fim do semestre

    cout << "Digite o número do mês atual: ";
    int atual;
    cin >> atual;

    if (atual >= inicio && atual <= fim)
        cout << "Você está em período de aulas.\n";
    else
        cout << "Férias!\n";

    return 0;
}
```

EXERCÍCIOS DE APRENDIZAGEM

VOCÊ DEVE ESCREVER PROGRAMAS PARA REALMENTE APRENDER

1. Considerando as declarações abaixo, responda dizendo qual o **tipo** da informação que deve ser armazenada na variável:

```
enum trave {LEsq, LDir, CantoEsq, CantoDir, Centro};

union jogador
{
    char nome[25];
    int numero;
};

struct gol
{
    jogador jog;           // identificação do jogador
    float x,y,z;           // posição da bola nas coordenadas
    trave local;           // onde a bola entrou
    float velo;            // velocidade da bola
    float acel;            // aceleração da bola
};

gol estatistica[10];      // estatísticas para até 10 gols
```

- | | |
|--|---|
| i) estatistica vetor de gol | m) (estatistica + 9)->local não existe |
| j) estatistica[4] struct gol | n) estatistica[2].velo float |
| k) estatistica[1].jog union jogador | o) (estatistica + 1)->jog.nome[0] não existe |
| l) estatistica->jog.numero não existe | p) estatistica[6].acel float |

2. Considerando o registro gol definido abaixo, crie um programa que leia e armazene em um vetor os dados dos três últimos gols que você assistiu.

```
union jogador
{
    char nome[25];        // nome do jogador
    int numero;           // numero da camisa do jogador
};

struct gol
{
    jogador jog;          // identificação do jogador
    int hora, min;        // hora e minuto em que o gol foi marcado
};
```

Digite os dados dos 3 últimos gols:
Gol: **Bebeto 7 14:30**
Gol: **Zico 9 45:00**
Gol: **Pele 10 05:45**

3. A prefeitura de Mossoró elabora mensalmente um caderno com a programação cultural da cidade. Ela deseja ter um programa que permita construir uma lista de eventos.

O programa deve cadastrar as seguintes informações de um evento:

- Data (um registro que possui três valores inteiros – dia, mês e ano)
- Horário (um registro que possui dois valores inteiros – hora e minuto)
- Descrição (com no máximo 18 caracteres)

Os eventos devem ser registros inseridos em um vetor com no máximo 10 elementos. Peça ao usuário para entrar com as informações de 2 eventos. Em seguida exiba um resumo mostrando todos os eventos cadastrados.

```
Entre com 2 eventos:
#1
Data: 26 10 2010
Hora: 16 00
Desc: Musica ao Vivo
#2
Data: 26 10 2010
Hora: 18 00
Desc: Show de Dança
-----
Eventos Cadastrados
26/10/2010 16:00 Musica ao Vivo
26/10/2010 18:00 Show de Dança
```

4. Analise o uso de enumeração no código abaixo:

```
#include <iostream>
using namespace std;

enum mes {JAN, FEV, MAR, ABR, MAI, JUN, JUL, AGO, SET, OUT, NOV, DEZ};

int main()
{
    char meses[12][10] =
    {
        "Janeiro", "Fevereiro", "Marco", "Abril", "Maio", "Junho",
        "Julho", "Agosto", "Setembro", "Outubro", "Novembro",
        "Dezembro"
    };

    for (mes ind = JAN; ind <= DEZ; ind = mes(ind + 1))
        cout << meses[ind] << endl;

    return 0;
}
```

Refaça o programa de forma que ele exiba os dias da semana.