

Lista 10 (EXERCÍCIOS)

1) Considere a declaração abaixo:

```
typedef int * ptr;  
ptr p1, p2;
```

Qual o tipo de dado das variáveis p1 e p2?

2) Caso a questão anterior fosse alterada para:

```
typedef int* ptr;  
const ptr p1, p2;
```

Qual o significado do modificador const nessa declaração? O ponteiro é constante ou o conteúdo?

3) Modifique o exercício anterior para que: a) o conteúdo apontado seja constante, b) o ponteiro seja constante, c) o conteúdo apontado seja constante e o ponteiro seja constante. Para isto, defina novos tipos: a) const_ptr, b) ptr_const, c) const_ptr_const.

4) Qual a diferença entre os tipos das variáveis abaixo? Todas são do tipo ponteiro para caractere?

```
typedef char * string_t;  
#define string_d char *  
string_t s1, s2;  
string_d s3, s4;
```

5) Dadas as definições:

```
#define MAX_X 50  
#define MAX_Y 50  
#define MAX_BARATAS (MAX_X * MAX_Y / 3)
```

```

typedef enum Vida_e
{
    VIVA,
    MORTA
} Vida;

typedef struct Barata_t
{
    unsigned int x, y;
    Vida estaViva;
} Barata;

int Matriz[MAX_X][MAX_Y];
Barata Baratas[MAX_BARATAS];

```

e as funções `srand` e `rand` da biblioteca `stdlib.h`.

Construa um programa que:

- sorteie posições para todas as baratas dentro da matriz, sem que duas baratas ocupem o mesmo lugar;

- entre num laço no qual, uma a uma, todas as baratas andam um quadrado para uma direção aleatória (esquerda, direita, cima, baixo), sem, é claro, sair dos limites da matriz;

Obs.: cabe apenas uma barata por quadrado, i. e., célula da matriz.

- quando sobre um quadrado já estiverem passado (andaram) três baratas, tal quadrado torna-se infectado; a próxima barata que passar sobre o quadrado morre e desaparece, e o quadrado torna-se não-infectado novamente;

- o laço prossegue até que a população seja um terço da população no início do programa.

Obs.: deve ser fornecido, na saída, a posição de cada barata ainda viva e o número de iterações do laço.

6) Crie uma biblioteca consistindo de `ponto.h` e `ponto.c` Defina uma estrutura `Ponto` que represente pontos no plano Cartesiano. Escreva funções para medir a distância entre dois pontos e comparar se dois pontos são iguais.

7) Crie uma biblioteca consistindo de `retangulo.h` e `retangulo.c`. Defina uma estrutura `Retangulo` utilizando a definição anterior de `Ponto`. Considere que o retângulo definido nessa biblioteca seja paralelo aos eixos. Escreva funções para: computar a área do retângulo, determinar se um ponto está localizado dentro de um retângulo e determinar se dois retângulos estão separados ou sobrepostos.

8) Crie uma estrutura para alunos que contenha o nome do aluno, a nota da primeira avaliação, a nota da segunda avaliação, a nota da APS, a nota da lista e a condição do aluno. A condição do aluno deve ser uma enumeração que determine se ele está aprovado ou reprovado. Realize a leitura de dados de um arquivo `notas.txt` que contenha o nome do aluno e suas notas. Determine a condição de aprovado ou reprovado de cada aluno.

9) Crie uma enumeração para os dias do mês. Escreva uma função que receba um dia do mês, do tipo da enumeração criada, e o ano. Essa função deve retornar os dias do mês, considerando anos bissextos.

10) Crie enumerações de naipes, cartas e condições de vitória (venceu, perdeu, empatou). Desenvolva um jogo de truco com quatro jogadores aleatórios e salve as sequências de jogadas em um arquivo texto.

11) Escreva um programa que determine a endianness de uma plataforma 32 bits. Utilize uma união de inteiro e chars.

12) Escreva uma biblioteca para matrizes de inteiros. Defina uma estrutura `Vetor`, com um campo para o ponteiro do vetor e outro para seu tamanho. Defina uma estrutura `Matriz`, reutilizando a definição de `Vetor`. Escreva operações para alocar, desalocar, multiplicar e somar/subtrair matrizes.

13) Faça uso de pelo menos uma funcionalidade de cada biblioteca padrão da linguagem C.

14) Modifique o exercício 20 da Lista 9 para utilizar uma estrutura que contenha, pelo menos, o ponteiro do elemento alocado e o tamanho do vetor dinamicamente alocado. Caso seja necessário, defina novos tipos utilizando `typedef`.