

## Exercícios de revisão da Linguagem C

Implemente soluções em CodeBlocks para os seguintes problemas (podendo utilizar tudo visto no primeiro semestre)

- 1) Calcular o fatorial de N, onde N é inteiro. N deve ser fornecido pelo usuário.
- 2) Implementar o problema da mochila. Tendo-se uma sequência de 15 números inteiros positivos, deseja-se empacotá-los em uma mochila com tamanho 6, de forma que se coloque dentro dela preferencialmente os maiores valores, até que ela esteja cheia. Imprimir cada um dos itens em uma linha específica da tela:
  - Todos elementos a serem colocados na mochila;
  - Os elementos que entraram na mochila;
  - Os que ficaram fora da mochila;
  - Qual a soma dos que entraram na mochila;
  - Qual a soma dos elementos que não entraram na mochila.
- 3) Dado um vetor de inteiros não ordenado, faça uma função que retorne o valor mais próximo de um número fornecido pelo usuário. O vetor não pode ser ordenado.
- 4) Assumindo-se que se tenha um vetor de inteiros aleatórios com tamanho 10, implemente as seguintes funções, sem ordenar previamente o vetor:
  - Mostrar os valores do vetor;
  - encontrar o maior valor deste vetor;
  - encontrar os dois maiores valores do vetor, com apenas uma repetição do vetor;
  - Função para calcular a media dos valores do vetor;
  - Função para dizer se existem dois valores iguais no vetor.
- 5) Monte um menu que chame funções para inicializar uma matriz 4x4 de forma que:
  - os elementos da diagonal principal tenham valor 1 e os demais devem ter valor 0.
  - cada coluna possua o mesmo valor
  - cada linha possua o mesmo valor
  - cada elemento seja a soma dos índices da linha com a coluna
  - elementos acima da diagonal superior sejam 1, e o resto zero
  - elementos abaixo da diagonal superior sejam 1, e o resto zero
  - Após inicializadas, chamar função que imprime a matriz na tela de forma estruturada.
- 4) Função que diz qual linha de uma matriz 3x3 possui a maior soma.
- 6) Função para achar o maior e menor valor de uma matriz 2x3 qualquer de inteiros
- 7) Crie TADs, com possíveis hierarquias, para conter:
  - Nome de uma pessoa, endereço (número, rua, cidade, estado, país).
  - a definição de um conjunto de 3 coordenadas que definem um ponto no espaço 3D, que também possui uma cor associada, no formato R,G,B. - Ex: p(x,y,z,r,g,b)
  - Definição de uma esfera que possui centro e raio.

8) Assuma a seguinte definição:

```
int a, b;  
int *P1, *P2, **P3;  
float *P4;
```

Diga quais das sentenças são verdadeiras e quais são falsas

• a = 10	• P4 = P1
• b = &a	• P2 = P1 = &a
• P1 = a	• *P1 = 20
• a = &P1	• *P2 = *P1
• *P1 = &a	• *P3 = &P1
• &P1 = &a	• P3 = &P2
• P4 = &a	• **P3 = *P1

9) O que pode acontecer se for atribuído algum valor a um ponteiro que não tenha sido inicializado. Ex:

```
float *p;  
*p = 2000;
```

10) Em C, não se pode fazer uma função que retorne dois valores, exceto com o uso de estruturas. Uma forma de solucionar esta restrição é com o uso de ponteiros, pois a função pode receber qualquer número de variáveis por referência. Faça programa, que possuindo duas variáveis inteiras a=2 e b=3, chame a função void troca(int \*a1, int \*b1), que deve fazer a inversão dos valores de a e b, ou seja, b passa a valer o que a valia e a passa a valer o que b valia. Após a chamada da função troca(), imprimir os valores de a e b. A função troca também deve ser implementada, e deve ter tipo de retorno void.

11) Declare vetores de inteiro, char, float, double, long int com 5 posições, da seguinte forma:

```
int v[5] = {2, 5, 1, 4, 0};  
char c[5] = {'a', 'b', 'm', '4', '-'};  
float v[5] = {2.66, 0.125, 1.0, 4.99, 2.009};
```

Usando a função printf, com o argumento “%p”, e com vetores apontando para cada um dos tipos de dados, descubra quantos bytes é alocado pelo seu compilador a cada tipo de dados. Não se esqueça que para um vetor apontar para um vetor de float, ele deve ser do tipo float. Como se sabe que em um vetor as posições são contínuas, se for impresso o endereço de duas posições, pela diferença entre os dois endereços pode-se descobrir quantos bytes são alocados.