

**Exercícios para o 3º Estágio**

**Prazo de Entrega: 16/03/2014**

- 1) Faça um programa para ler Matrícula, Nome (máximo 30 caracteres) e Idade de N estudantes, sendo N informado pelo usuário (máximo 50). Em seguida, o programa deverá imprimir todos os dados lidos dos alunos com idade abaixo de 18 anos e, em seguida, imprimir todos os dados lidos dos alunos com idades maiores que a média de idades desses alunos.
- 2) Faça um programa para ler Matrícula, Nome (máximo 30 caracteres) e Idade de N estudantes, sendo N informado pelo usuário (máximo 50). Em seguida, o programa deverá imprimir os dados lidos em ordem de Matrícula, Nome ou Idade dos Alunos, de acordo com a solicitação do usuário. Para comparar cadeias de caracteres, use a função `strcmp()`. Modularize seu programa para utilizar várias funções que realizem apenas uma diferente tarefa cada.
- 3) Considerando o Programa das Figuras 5.8 e 6.9 do Livro Texto, Faça uma nova versão do programa da Fig. 6.9 que simule o lançamento de um dado vinte vezes e imprima quantas vezes cada face do dado foi lançada. Nesta nova versão continue usando o array frequência para registrar a quantidade de vezes que cada face do dado foi lançada e acrescente uma função para lançar um dado e outra para imprimir o array.
- 4) Faça um programa para ler dois conjuntos (A e B) de no máximo 10 elementos inteiros, gerar um terceiro conjunto, o conjunto diferença (A-B) e finalmente imprimir os três conjuntos. Os tamanhos de A e B serão informados pelo usuário. Utilize as funções: `le_conjunto()`, `imprime_conjunto()`, `elemento_pertence_ao_conjunto()` e `gera_o_conj_diferenca()`.
- 5) Faça um programa para corrigir provas de múltipla escolha de vários alunos. Cada prova tem 10 questões e cada questão tem seu valor informado, totalizando 10,0. O primeiro conjunto de dados a ser lido é o gabarito da prova, com cada resposta seguida de seu valor. Os outros dados serão a matrícula de cada aluno seguida de suas respectivas respostas da prova. Os dados dos alunos terminam quando o programa ler uma matrícula zero. O programa ainda:
  - a. para cada aluno, mostrará a sua matrícula e calculará e mostrará a sua nota;
  - b. calculará e mostrará a porcentagem de aprovação da turma, sabendo-se que a nota mínima é 7,0.
- 6) Faça um programa para ler duas matrizes de inteiros e imprimir as matrizes lidas e informar se a primeira é ou não a transposta da segunda. A dimensão de cada matriz deve ser informada pelo usuário (máximo 10 x 10)
- 7) Faça um programa para ler uma matriz de números inteiros 7 x7 e imprimir o elemento do meio de cada linha da matriz, após essas linha serem ordenadas crescentemente.
- 8) Faça um programa que leia uma matriz de dimensão M x N, Imprima a matriz lida e os totais de elementos positivos, negativos e nulos dessa matriz. Os valores de M e N devem ser informados pelo usuário (Máximo 10) e o programa deve usar as seguintes funções: `le_matriz()`, `imprime_matriz()`, `calcula_os_três_totais()` e `imprime_os_tres_totais()`.

Atenção! Você vai precisar de passagem de parâmetro por referência, além do array bidimensional.

9) Escreva um programa que leia N inteiros de 6 dígitos representando horários no formato HHMMSS e, se o horário for válido, imprime os horários no formato digital, HH:MM:SS . O valor de N é informado pelo usuário e o programa deverá usar as seguintes funções:

- *extrai\_componentes()* que recebe HHMMSS e retorna HH, MM e SS;
- *eh\_horario\_valido()* que recebe HH, MM e SS e retorna verdadeiro se esses valores formam um horário válido e falso no caso contrário e
- *imprime\_horario\_digital()* que recebe HH, MM, e SS e imprime HH:MM:SS

Atenção! Você vai precisar de passagem de parâmetro por referência.

10) Faça um programa que leia um array de N elementos distintos (no máximo 10), imprima o array lido, o maior e o menor elemento desse array. O programa deve usar uma função que retorne o endereço do maior elemento e outra que retorne o endereço do menor elemento do array. A função main() recebe estes endereços e imprime os elementos que estão armazenados nestes endereços. Não use indexação, use apontadores.

11) Faça um programa que leia um array de N elementos (no máximo 10), imprima o array lido, o array ordenado e a mediana (elemento que fica no meio do array após ordenação. No caso da quantidade de elementos ser par, a mediana é a média dos dois elementos do meio). O programa deve usar funções para ler array, imprimir array, ordenar array e trocar elementos do array durante a ordenação. Não use índices para percorrer o array nem para calcular a posição da mediana. Não use indexação, use apontadores.

12) Faça um programa que leia uma *string* (máximo 30 caracteres), com letras minúsculas e/ou maiúsculas, aleatoriamente. Em seguida, mostra a *string* lida e transforme as letras iniciais de cada palavra nessa *string* em letras maiúsculas e as demais em minúsculas. O programa torna a imprimir a *string*, após a transformação. O programa deve usar uma função para transformar a *string*. Não use índices para percorrer a *string*, use apontadores.

13) Faça um programa que leia e imprima uma *string* (máximo 30 caracteres) e determine se ela é ou não um palíndromo. O Programa não diferenciará minúsculas de maiúsculas. Não use índices para percorrer a *string*, use apontadores.

### **ATENÇÃO!!!!**

Em breve avisaremos os cinco exercícios que valerão nota.

*SUCESSO!*