

**CURSO:** Sistemas de Informação  
**DISCIPLINA:** Estrutura, Pesquisa e Ordenação de Dados  
**PROFESSOR (A):** Frederico Brito Fernandes  
**SEMESTRE/ANO:** 2010.2  
**TURMA:** Noite  
**CARGA HORÁRIA:** 80 h/a

### **Exercícios-Desafio 1:** Estruturas de controle, Operadores relacionais e lógicos, uso de funções de biblioteca

**ED 1.1)** Escreva um programa em C que solicita ao usuário que introduza uma nota, cujo valor pode variar entre 0.0 e 10.0, e imprime o conceito referente a esta nota de acordo com a tabela a seguir:

#### **Nota Conceito**

Entre 9.0 (inclusive) e 10.0 (inclusive) A  
Entre 8.0 (inclusive) e 9.0 B  
Entre 7.0 (inclusive) e 8.0 C  
Entre 6.0 (inclusive) e 7.0 D  
Entre 5.0 (inclusive) e 6.0 E  
Menor do que 5.0 F

O programa deve ainda apresentar mensagens de erro correspondentes a entradas fora do intervalo de valores permitido. Em qualquer outro aspecto, você pode assumir que as entradas são corretas.

**ED 1.2)** Escreva um programa em C que solicita ao usuário para introduzir uma série de valores inteiros positivos e lê estes números até que o usuário introduz o valor 0. Então, o programa deve apresentar o menor, o maior e a média dos valores introduzidos (sem levar em consideração 0). Caso o usuário introduza um número negativo, o programa deve informá-lo de que o valor não é válido e não deve levar este valor em consideração.

Exemplo de interação com o programa (**negrito** corresponde a entrada do usuário):

```
[Apresentação do programa]
Introduza uma série de números inteiros positivos terminando a série
com zero.
Introduza o próximo número: 5
Introduza o próximo número: -2
-2 não é um valor válido.
Introduza o próximo número: 1
Introduza o próximo número: 6
Introduza o próximo número: 0
Menor valor introduzido: 1
Maior valor introduzido: 6
Média dos valores introduzidos: 4.0
```

**ED 1.3)** Escreva um programa em C que solicita ao usuário para introduzir n valores inteiros, lê estes números e apresenta o menor, o maior e a média dos valores introduzidos. O valor n deve ser o primeiro dado introduzido pelo usuário. Exemplo de interação com o programa (**negrito** corresponde a entrada do usuário):

```
[Apresentação do programa]
Quantos números você irá introduzir? 3
Introduza o próximo número: 5
Introduza o próximo número: -2
Introduza o próximo número: 0
Menor valor introduzido: -2
Maior valor introduzido: 5
Média dos valores introduzidos: 1.0
```

**ED 1.4)** Escreva um programa em C que brinque de adivinhar números com um usuário. O jogo consiste em gerar um número aleatório entre 1 e 100 e solicitar ao usuário que tente adivinhar o número gerado. De acordo com a resposta do usuário, o programa deve informar o novo intervalo que o número está. Cada partida termina quando o usuário acerta o número, faz o máximo de 5 tentativas ou introduz um número negativo. Números não negativos fora do intervalo de 1 a 100 não devem ser levados em consideração. Ao final de cada partida, informe ao usuário se ele acertou o número gerado ou qual era este número. Permita que o usuário jogue quantas partidas ele desejar e, antes de encerrar o programa, informe ao usuário quantas partidas foram jogadas e quantas vezes ele acertou o número.

Exemplo de interação do programa (**negrito** corresponde a entrada do usuário), supondo que o número aleatório seja 78:

```
Vou pensar num numero entre 1 e 100.
Tente adivinhar este numero em no máximo 5 tentativas.
Se você digitar um numero negativo, estará desistindo da
partida.
*** Nova partida ***
Dê seu chute entre 1 e 100: 5
Infelizmente você errou.
Dê seu chute entre 5 e 100: 54
Infelizmente você errou.
Dê seu chute entre 54 e 100: 85
Infelizmente você errou.
Dê seu chute entre 54 e 85: 75
Infelizmente você errou.
Dê seu chute entre 75 e 85: 78
Partida encerrada. Parabéns!!! Você acertou!!!
Deseja jogar uma nova partida? s
*** Nova partida ***
Dê seu chute entre 1 e 100: 12
Infelizmente você errou.
Dê seu chute entre 12 e 100: 10
Numero invalido!!! O numero deve estar entre 12 e 100.
Dê seu chute entre 12 e 100: 21
...
Partida encerrada. Infelizmente, você não acertou. Meu número era 78.
Deseja jogar uma nova partida ('n' ou 'N' encerra o jogo)? n
```

```
*** Resumo do jogo ***
*** Numero de partidas jogadas: 2
*** Numero de vezes que você acertou: 1
Obrigado por ter jogado comigo. Digite qualquer tecla para sair do
programa.
```

### Sugestões:

1. A função `rand()` (`#include <stdlib.h>`) gera números inteiros aleatórios entre 0 e o valor dado pela constante `RAND_MAX`. Para gerar números aleatórios entre `N` e `M`, utilize a fórmula: `rand()%M + N`.
2. Para não gerar os mesmos números aleatórios a cada execução do programa, a função `rand()` precisa ser *alimentada* com um valor arbitrário que não seja o mesmo em duas execuções do programa. Uma técnica comum de fazer isto é usar o valor retornado pela função `time()` (`#include <time.h>`) que corresponde ao número de segundos decorridos desde 00:00:00 de 1º de janeiro de 1970. Para usar esta função declare uma variável `t` do tipo `time_t` (definido em `<time.h>`) e utilize a função `srand()` (`#include <stdlib.h>`) para alimentar o gerador de números aleatórios como na instrução: `srand((unsigned) time(&t));`. [O *casting* `(unsigned)` é necessário aqui porque a função `time()` retorna um valor do tipo `time_t`.]
3. É muito trabalhoso testar o programa com números no intervalo entre 1 e 100, porque a probabilidade de acerto é muito pequena. Portanto, teste seu programa com números entre 1 5 e, após estar satisfeito com os resultados, retorne aos valores originais.

**Observação:** Na construção dos programas devem ser observadas rigorosamente as normas contidas no documento *Orientações Para Escrita de Programas* .