

## Algoritmos e Programação: Fundamentos

**PROVA DO GRAU B - GABARITO**

Resolver as questões com consulta local somente ao material apresentado em aula, ao repositório de exemplos da turma e aos códigos-fonte dos exercícios resolvidos pelo próprio aluno. Nos primeiros 15 minutos da prova será permitido baixar este material. Não é permitido consultar outros colegas, a Internet, e-mail, drivers virtuais na nuvem, redes sociais ou telefone celular. Cada questão deve ter seu próprio arquivo-fonte, identificado com o número da questão (1.c, 2.c, etc) em um diretório chamado **ProvaGrauA-NomeSobrenome**. O arquivo contendo o diretório compactado (**.zip**) deve ser enviado pelo Moodle até as 22h15. Os nomes das variáveis devem ser autoexplicativos.

- (0.5 pt)** Implemente uma função **sortear** que receba o intervalo de valores inteiros **início** e **fim** como parâmetro, sorteie e retorne um número dentro do intervalo (considerando intervalo fechado [início, fim]).

Exemplo de uso:

```
int nroSorteado = sortear(1,10);
```

## Uma possível solução

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

// Cabeçalho da função sortear
int sortear(int inicio, int fim);

int main()
{
    // Inicializa o gerador de números aleatórios
    srand(time(NULL));

    // Exemplo de uso da função sortear
    int numero = sortear(1, 10);
    printf("Número sorteado: %d\n", numero);

    return 0;
}

// Implementação da função sortear
int sortear(int inicio, int fim)
{
```

```
    return inicio + rand() % (fim - inicio + 1);  
}
```

## Explicação

### Cabeçalho da Função

O cabeçalho da função sortear é escrito como:

```
int sortear(int inicio, int fim);
```

#### 1. Tipo de Retorno: int

- Indica que a função retorna um número inteiro.

#### 2. Parâmetros:

- int inicio: Define o início do intervalo.
- int fim: Define o final do intervalo.

Os parâmetros devem ser fornecidos na ordem correta e com os tipos correspondentes. Ao chamar a função, **os valores fornecidos “substituem” os parâmetros definidos** (são atribuídos às variáveis temporárias `inicio` e `fim`).

*Exemplo:*

```
int numero = sortear(1, 10);
```

Aqui, 1 substitui `inicio` e 10 substitui `fim`.

### Passagem de Parâmetros

A função `sortear` recebe dois parâmetros inteiros: `inicio` e `fim`. Esses parâmetros definem o intervalo fechado de onde o número aleatório será sorteado.

*Quando a função é chamada, os valores são passados para os parâmetros e usados dentro do cálculo (nossa “fórmula mágica” dos sorteios):*

```
inicio + rand() % (fim - inicio + 1);
```

### Retorno da Função

O retorno da função se dá pelo comando `return` seguido da variável que foi usada para fazer o cálculo referente à lógica da função ou, como neste caso é um cálculo simples, podemos já mandar diretamente a expressão matemática que faz o sorteio.

- 
2. **(4.0 pts)** Implemente um programa que, dado uma matriz 5x5 preenchida com valores entre 0 e 4, faça a contagem de quantos valores de cada um dos números possíveis esta matriz contém.

Exemplo de matriz:

0	4	3	2	1
1	0	2	4	0
3	2	4	4	0
1	2	3	4	0
0	0	0	1	2

Retornaria a seguinte contagem:

```
Quantidade de valores 0: 8
Quantidade de valores 1: 4
Quantidade de valores 2: 5
Quantidade de valores 3: 3
Quantidade de valores 4: 5
```

## Uma possível solução

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

int main()
{
    srand(time(0));
    int mat[5][5];

    // Preenchendo a matriz com valores entre 0 e 4 e exibindo ela
    for (int i = 0; i < 5; i++) // percorre as linhas
    {
        for (int j = 0; j < 5; j++) // percorre as colunas
        {
            mat[i][j] = rand() % 5; // sorteia um nro entre 0 e 4
            printf("%d ", mat[i][j]);
        }
        printf("\n");
    }

    // Array de contadores
    int conts[5]; // para armazenar a quantidade dos valores

    // Inicializar cada contador com zero
    for (int i = 0; i < 5; i++)
    {
        conts[i] = 0;
    }
```

```

/* OBS. Essa parte de fazer a contagem já poderia ser feita no Loop anterior, mas pensando numa construção de código que futuramente possa ser modularizado (por exemplo, virar uma função), vou fazer em separado */

// Percorrer a matriz para fazer a contagem dos valores
for (int i = 0; i < 5; i++) // percorre as linhas
{
    for (int j = 0; j < 5; j++) // percorre as colunas
    {
        conts[mat[i][j]]++; // o array de contadores no índice correspondente ao valor (0 a 4) é incrementado
    }
}

// Exibindo os resultados
for (int i = 0; i < 5; i++)
{
    printf("Quantidade de valores %d: %d\n", i, conts[i]);
}
return 0;
}

```

## Explicação

### Passo a Passo

1. **Preenchimento e Exibição da Matriz:**
  - a) A matriz `mat[5][5]` é preenchida com números aleatórios entre 0 e 4, usando `rand() % 5`.
  - b) O conteúdo da matriz é exibido no console imediatamente após o preenchimento.
2. **Uso do Vetor de Contadores (`conts[5]`):**
  - a) Criamos um array `conts` para armazenar a contagem de cada valor possível (0 a 4).
  - b) Cada índice de `conts` mapeia diretamente o valor correspondente da matriz: por exemplo, `conts[0]` conta quantas vezes o número 0 aparece na matriz.
3. **Contagem dos Valores:**
  - a) Percorremos a matriz e, para cada elemento `mat[i][j]`, incrementamos o contador correspondente em `conts[mat[i][j]]`.
4. **Exibição dos Resultados:**
  - a) O vetor `conts` é usado para exibir a contagem final de cada número.

### Conceito Importante:

5. **Mapeamento de Índices:**
  - a) O valor de cada elemento da matriz é utilizado como índice no vetor `conts`.
  - b) Essa técnica é eficiente para armazenar contagens, evitando a necessidade de várias condições (`if` ou `switch`).

- 
3. **(4.0 pts) Correção de provas objetivas:** faça um programa que permita cadastrar um vetor com o gabarito de uma prova objetiva contendo 10 questões, cujos resultados podem ser de 'a' a 'f'. Depois disso, permita ao usuário a opção de entrar com as respostas dadas por um estudante.

**Exemplo de gabarito:**

a	f	c	d	d	a	e	e	b	a
---	---	---	---	---	---	---	---	---	---

O programa deve:

- Contabilizar o número de respostas certas e dar a pontuação final, de 0 a 10.
- Durante o processo de correção, imprimir na tela quais questões foram acertadas e quais o estudante errou. Neste caso, deve-se imprimir a resposta correta.

**Exemplo:**

Respostas do estudante:

a	f	a	d	f	a	e	e	c	a
---	---	---	---	---	---	---	---	---	---

Saída do programa:

```
Questão 1: resposta correta
Questão 2: resposta correta
Questão 3: resposta incorreta. A resposta correta é c.
Questão 4: resposta correta
Questão 5: resposta incorreta. A resposta correta é d.
Questão 6: resposta correta
Questão 7: resposta correta
Questão 8: resposta correta
Questão 9: resposta incorreta. A resposta correta é b.
Questão 10: resposta correta
A pontuação do estudante é 7/10.
```

## Uma possível solução

```
#include <stdio.h>
#include <string.h>

#define TAM 10 // Define o tamanho do gabarito e das respostas

int main()
{
    char gabarito[TAM], respostas[TAM];
```

```

int pontuacao = 0;

// Entrada do gabarito
printf("Digite o gabarito (10 questões, de 'a' a 'f'):\n");
for (int i = 0; i < TAM; i++)
{
    printf("Questão %d: ", i + 1);
    scanf(" %c", &gabarito[i]);
}

// Entrada das respostas do estudante
printf("\nDigite as respostas do estudante (10 questões, de 'a' a 'f'):\n");
for (int i = 0; i < TAM; i++)
{
    printf("Questão %d: ", i + 1);
    scanf(" %c", &respostas[i]);
}

// Correção das respostas
printf("\nCorreção:\n");
for (int i = 0; i < TAM; i++)
{
    if (respostas[i] == gabarito[i])
    {
        printf("Questão %d: resposta correta\n", i + 1);
        pontuacao++;
    }
    else
    {
        printf("Questão %d: resposta incorreta. A resposta correta é %c.\n", i + 1, gabarito[i]);
    }
}

// Exibição da pontuação final
printf("\nA pontuação do estudante é %d/%d.\n", pontuacao, TAM);

return 0;
}

```

## Explicação

### Correção das Respostas

- O programa percorre os arrays **gabarito** e **respostas** ao mesmo tempo, comparando cada elemento.
- Para cada questão:

- Se o valor do índice no array **respostas** for igual ao índice correspondente no array **gabarito**, a resposta é correta e a pontuação é incrementada.
- Caso contrário, é exibida a resposta correta para aquela questão.

- 
4. **(1.5 pts) Programa “Cebolinha”:** ler uma *string* (array de char) e converter todos os r's em l's.

**Exemplo:**

Entrada:

O rato roeu a roupa do rei de Roma

Saída:

O lato loeu a loupa do lei de Loma

**OBS:** para ler uma string com espaços, podemos usar o comando fgets():

```
char myString[100];
printf("Digite uma palavra ou frase: ");
fgets(myString, sizeof(myString), stdin);
printf("String digitada: %s", myString);
```

## Uma possível solução

```
#include <stdio.h>
#include <string.h>

// Função para substituir todos os 'r' por 'l' e 'R' por 'L'
void substituirRPorL(char frase[])
{
    int tam = strlen(frase);
    for (int i = 0; i < tam; i++) // Percorre a string até o tamanho calculado
    {
        if (frase[i] == 'r') // Substitui 'r' minúsculo por 'l'
        {
            frase[i] = 'l';
        }
        else if (frase[i] == 'R') // Substitui 'R' maiúsculo por 'L'
        {
            frase[i] = 'L';
        }
    }
}

int main()
```

```
{  
    char frase[100]; // Declara a string para armazenar a frase  
  
    printf("Digite uma frase: ");  
    fgets(frase, sizeof(frase), stdin); // Lê a string com espaços  
  
    substituirRPorL(frase); // Chama a função para substituir os caracteres  
  
    printf("Frase alterada: %s\n", frase); // Exibe a string alterada  
  
    return 0;  
}
```

## Explicação

### Função substituirRPorL

- **Objetivo:** Percorre cada caractere da string e substitui os caracteres 'r' e 'R' pelos respectivos 'l' e 'L'.
- Utiliza a função `strlen(frase)` para calcular o tamanho da string uma única vez.

### Passagem de Array por Referência

- Strings (arrays de `char`) são passadas por referência em C, permitindo que as alterações feitas dentro da função `substituirRPorL` impactem diretamente a string original no `main`.

**BOA PROVA!** 😊

“A ciência é feita de erros, mas são erros que é bom cometer, porque nos levam pouco a pouco à verdade.” — Jules Verne, *Viagem ao Centro da Terra*