

# Atividade Prática 2 — Técnicas de Programação Gabarito

Prof. Me. Fernando Esquírio & Prof. Me. Matheus Fernandes

## Exercício 01

```
#include <stdio.h>
#include <stdlib.h>
typedef struct func
{
    int id;
    char nome[50];
    char cpf[11];
    int dia;
    int mes;
    int ano;
} tFuncionario;

int validarCPF(char *cpf);
int validarData(int dd, int mm, int aa);

int main(){
    tFuncionario dados[5];
    int registro = 0, escolha = 0, letras = 1;
    char continua = 's';
    do{
        printf("Escolha uma opcao:\n");
        printf("1 - Cadastrar funcionarios\n");
        printf("2 - Listar funcionarios\n");
        printf("3 - Sair\n");
        printf("Escolha uma opcao: ");
        scanf("%d", &escolha);
        fflush(stdin);
        switch(escolha){
            case 1:
                do{
                    system("cls");
                    dados[registro].id = registro+1;
                    //Leitura dos dados
                    do{
                        letras = 1;
                        printf("\nDigite o nome do funcionario: ");
                        gets(dados[registro].nome);
                        fflush(stdin);
                        for(int i = 0; (dados[registro].nome[i] != '\0' && i < 50); i++){
                            if(!(dados[registro].nome[i] >= 'A' && dados[registro].nome[i] <= 'Z' ||
                                dados[registro].nome[i] >= 'a' && dados[registro].nome[i] <= 'z' ||
                                dados[registro].nome[i] == ' '))
                                letras = 0;
                            if (i == 49)
                                dados[registro].nome[i] = '\0';
                        }
                    }while(!letras);
                }do{
                    printf("\nDigite o CPF: ");
                    scanf("%c%c%c%c%c%c%c%c%c%c%c", &dados[registro].cpf[0], &dados[registro].cpf[1],
                        &dados[registro].cpf[2], &dados[registro].cpf[3], &dados[registro].cpf[4], &dados[registro].cpf[5],
                        &dados[registro].cpf[6], &dados[registro].cpf[7], &dados[registro].cpf[8], &dados[registro].cpf[9],
                        &dados[registro].cpf[10]);
                    fflush(stdin);
                    if(!validarCPF(dados[registro].cpf))
                        printf("CPF Invalido!\n\n");
                }while(!validarCPF(dados[registro].cpf));
                do{
                    printf("\nDigite a data de nascimento: ");
                    scanf("%i%c%i%c%i", &dados[registro].dia, &dados[registro].mes, &dados[registro].ano);
                    fflush(stdin);
                    if(!validarData(dados[registro].dia, dados[registro].mes, dados[registro].ano))
                        printf("Data Invalida!!\n\n");
                }while(!validarData(dados[registro].dia, dados[registro].mes, dados[registro].ano));
```

```

        registro++;
        printf("\n\nDeseja cadastrar mais funcionarios? s-sim ou n-nao\n");
        scanf("%c", &continua);
        fflush(stdin);
    }while(continua == 's');
    system("cls");
    break;
case 2:
    system("cls");
    if(registro != 0){
        printf("\nID\tNome\t\tCPF\t\tData de nascimento\n");
        for(int i = 0; i < registro; i++){
            //Imprime os dados
            printf("%d\t", dados[i].id);
            printf("%s\t", dados[i].nome);
            for(int j = 0; j <=10; j++){
                printf("%c", dados[i].cpf[j]);
                if(j==2||j==5)
                    putchar('.');
                if(j==8)
                    putchar('-');
            }
            printf("\t\t%i/%i/%i\n\n", dados[i].dia, dados[i].mes, dados[i].ano);
        }
    }
    else{
        printf("Nao existe registros!\n\n");
    }
    system("Pause");
    break;
case 3:
    printf("\n\nObrigado a todos!!\n\n");
    break;
}
}while(escolha == 1 || escolha == 2);

return 0;
} //Fim da main()

//Retorno: [1] - Se for válido | [0] - Se for inválido
int validarCPF(char *cpf)
{
    int i, j, digito1 = 0, digito2 = 0;
    if(strlen(cpf) != 11)
        return 0;
    else if((strcmp(cpf,"00000000000") == 0) || (strcmp(cpf,"11111111111") == 0) ||
    (strcmp(cpf,"22222222222") == 0) ||
    (strcmp(cpf,"33333333333") == 0) || (strcmp(cpf,"44444444444") == 0) ||
    (strcmp(cpf,"55555555555") == 0) ||
    (strcmp(cpf,"66666666666") == 0) || (strcmp(cpf,"77777777777") == 0) ||
    (strcmp(cpf,"88888888888") == 0) ||
    (strcmp(cpf,"99999999999") == 0))
        return 0; //se o CPF tiver todos os números iguais ele é inválido.
    else
    {
        ///digito 1-----
        for(i = 0, j = 10; i < strlen(cpf)-2; i++, j--) ///multiplica os números de 10 a 2 e soma os
        resultados dentro de digito1
            digito1 += (cpf[i]-48) * j;
        digito1 %= 11;
        if(digito1 < 2)
            digito1 = 0;
        else
            digito1 = 11 - digito1;
        if((cpf[9]-48) != digito1)
            return 0; //se o digito 1 não for o mesmo que o da validação CPF é inválido
        else
            ///digito 2-----
            {
                for(i = 0, j = 11; i < strlen(cpf)-1; i++, j--) ///multiplica os números de 11 a 2 e soma os
                resultados dentro de digito2
                    digito2 += (cpf[i]-48) * j;
            }
    }
}

```

```

        digito2 %= 11;
        if(digito2 < 2)
            digito2 = 0;
        else
            digito2 = 11 - digito2;
        if((cpf[10]-48) != digito2)
            return 0; ///se o digito 2 não for o mesmo que o da validação CPF é inválido
    }
}
return 1;
}

int validarData(int dd, int mm, int aa){
    if(mm < 1 || mm > 12){
        return 0;
    }
    else{
        if(mm == 1 || mm == 3 || mm == 5 || mm == 7 || mm == 8 || mm == 10 || mm == 12){
            if(dd < 1 || dd > 31){
                return 0;
            }
        }
        else if(mm == 2){
            if(dd < 1 || dd > 28){
                return 0;
            }
        }
        else{
            if(dd < 1 || dd > 30){
                return 0;
            }
        }
    }
}
if(aa < 1950 || aa > 2002){
    return 0;
}
return 1;
}

```

## Exercício 02

```

#include <stdio.h>
#include <stdlib.h>
int main(){
    char texto[200], alfabeto[26], parada = '1';
    int tam = 0, cont=0;
    printf("Escreva seu texto, digite ~ para parar:\n");
    do{
        parada = fgetc(stdin);

        if(parada == '~'){
            texto[tam] = '\0';
            tam++;
        }
        else if (parada == 13){
            putchar('\n');
        }
        else{
            texto[tam] = parada;
            tam++;
        }
    }while(parada != '~');
    for(int i=0; i<26; i++)
        alfabeto[i]=0;

    while(texto[cont]!='\0'){
        if(texto[cont]>='A'&& texto[cont]<='Z' || texto[cont]>='a'&& texto[cont]<='z')
            alfabeto[(texto[cont] | ' ') - 'a']++;
        cont++;
    }//Fim do while(texto[cont]!='\0')
}

```

```

//Imprimi as ocorrências
printf("\n\nOcorrências:\n");
printf("Letras:\t\t");
for(char i='a'; i<='z'; i++)
    printf(" %2c", i);

printf("\nN Vezes:\t");
for(int i=0; i<26; i++)
    printf(" %2d", alfabeto[i]);

putchar('\n');
return 0;
} //Fim da função main()

```

## Exercício 02 – Solução do Murillo e do João Vitor

```

// Joao Vitor Andrade de Araujo - RA: 21377112
// Murillo Monteiro Pedroso - RA:21393496

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

typedef struct txt
{
    int id;
    char titulo[200];
    char texto[2000];
    int tamanho_titulo;
    int tamanho_texto;
    int letras_titulo[26];
    int letras_texto[26];
} tdiario;

int main()
{
    // Inicialização de variáveis de controle e variáveis acumuladoras
    tdiario dados[50];
    dados[0].id = -1;
    int x = 0;
    int i = 0;
    int j = 0;
    int resposta = 0;
    int contador_dois = 0;
    int controle_minusculo = 97;
    int controle_maiusculo = 65;

    /*
        Upgrades realizados:
        - Menu de interação.
        - Usuário fazer busca de textos já feitos.
        - Contagem de caracteres.
        - Separação entre título e texto e contagem de caracteres separada entre eles com seus respectivos limites.
        - Não distinguir letras maiúsculas e minúsculas.
        - Uso de tecla especial para encerrar o texto, ou seja, o "enter" não encerra.
        - Verificação de opção errada do usuário.
    */

    printf("\nEmpresa do Malvado Doofenshmirtz!!!\n\n");
    do
    {
        resposta = 0;
        do // Início do programa, pega a primeira opção do usuário, ou seja, define o primeiro contato usuário x programa
        {
            resposta = 0;
            printf("\n\n\t\t01a Dr. Doofenshmirtz!\n\n\t\t0 que deseja fazer?\n\t\t1 - Criar um novo plano\n\t\t2 - Visualizar seus planos antigos\n\t\t3 - Sair\n");

```

```

printf("Numero da opcao: ");
fflush(stdin);
scanf(" %i", &resposta);
setbuf(stdin, NULL);

if (resposta < 1 || resposta > 3)
{
    printf("Dr. Doofenschmirtz, opcao invalida, digite novamente:\n");
}
}
while (resposta < 1 || resposta > 3); //repete enquanto resposta for invalida

switch (resposta)
{
case 1:

    dados[x].id = x;

    for (i = 0; i < 50; i++) //inicializa os vetores do texto e demarca os espaços para que não
haja interrupção na contagem
    {
        dados[x].titulo[i] = 0;
    }
    printf("De um titulo para o seu plano.\nQuando desejar encerrar aperte '^' + espaco, por fim
aperte enter para parar (Limite de 20 caracteres):\n");
    fflush(stdin);
    scanf(" %20[^^]s", &dados[x].titulo);
    setbuf(stdin, NULL);
    dados[x].tamanho_titulo = strlen(dados[x].titulo); //mede o tamanho do texto
    printf("\n\tQuantidade de caracteres usados no titulo: %i/20\n\n", dados[x].tamanho_titulo);

    for (i = 0; i < 26; i++) //inicializa todo o vetor para evitar lixo de memoria
    {
        dados[x].letras_titulo[i] = 0; //zera cada letra
    }
    /*
    Nesse segundo bloco é basicamente como o bloco acima, há os mesmos funcionamentos
    porém, no entanto. Os limites de caracteres são maiores como pedido no exercício.
    */

    printf("Digite seu texto. Quando desejar encerrar aperte '^' + espaco, por fim aperte enter
para parar (Limite de 200 caracteres):\n");
    fflush(stdin);
    scanf(" %200[^^]s", &dados[x].texto);
    setbuf(stdin, NULL);
    dados[x].tamanho_texto = strlen(dados[x].texto); //mede o tamanho do texto
    printf("\n\tQuantidade caracteres usados no texto: %i/200\n\n", dados[x].tamanho_texto);

    for (i = 0; i < 26; i++) //inicializa todo o vetor para evitar lixo de memoria
    {
        dados[x].letras_texto[i] = 0; //zera cada letra
    }

    for (j = 0; j < dados[x].tamanho_titulo; j++) //for para percorrer cada caracter do texto
    {
        if (dados[x].titulo[j] == '\n')
        {
            dados[x].titulo[j] = ' ';
        }
        /* Nessas duas variáveis são armazenadas os valores da tabela
        ASCII, dessa forma há um início para os laços de repetições e
        checagem "saberem" o que está sendo implementado.*/
        controle_minusculo = 97;
        controle_maiusculo = 65;
        for (i = 0; i < 26; i++) //for para comparar cada letra com todas as letras do alfabeto
        {
            if (dados[x].titulo[j] == controle_minusculo || dados[x].titulo[j] ==
controle_maiusculo) //compara a letra com cada letra do alfabeto
            {
                dados[x].letras_titulo[i]++; //caso a letra corresponda ele soma 1 no vetor de
letras
            }
        }
    }
}

```

```

        controle_minusculo++; //Vai pra prox letra minuscula
        controle_maiusculo++; //Vai pra prox letra maiuscula
        /* No final as duas variáveis impressas de uma vez
        fazendo que não haja distinção de maiúsculas e minúsculas.*/
    }
}

for (j = 0; j < dados[x].tamanho_texto; j++) //for para percorrer cada caracter do texto
{
    if (dados[x].texto[j] == '\n')
    {
        dados[x].texto[j] = ' ';
    }
    /*
    Nessas duas variáveis são armazenadas os valores da tabela
    ASCII, dessa forma há um início para os laços de repetições e
    checagem "saberem" o que está sendo implementado.
    */
    controle_minusculo = 97;
    controle_maiusculo = 65;
    for (i = 0; i < 26; i++) //for para comparar cada letra com todas as letras do alfabeto
    {
        if (dados[x].texto[j] == controle_minusculo || dados[x].texto[j] ==
controle_maiusculo) //compara a letra com cada letra do alfabeto
        {
            dados[x].letras_texto[i]++; //caso a letra corresponda ele soma 1 no vetor de
letras
        }
        controle_minusculo++; //Vai pra prox letra minuscula
        controle_maiusculo++; //Vai pra prox letra maiuscula
        /*
        No final as duas variáveis impressas de uma vez
        fazendo que não haja distinção de maiúsculas e minúsculas.
        */
    }
}

controle_maiusculo = 65;
/*
Nos laços de repetições abaixo são para que haja uma contagem de vetor e checagem
de quantas vezes a letra digitada passou para que haja um total e seja mostrado ao
usuário.
*/
printf("Letras:          ");
for (i = 0; i < 26; i++)
{
    printf("    %c ", controle_maiusculo);
    controle_maiusculo++;
}
printf("\n\n");
printf("N vezes no titulo: ");

for (i = 0; i < 26; i++)
{
    printf("    %i  ", dados[x].letras_titulo[i]);
}
printf("\n");
printf("N vezes no texto:  ");

for (i = 0; i < 26; i++)
{
    printf("    %i  ", dados[x].letras_texto[i]);
}
x++;
break;

case 2:
    if (dados[0].id < 0) //verifica se existem funcionarios cadastrados
    {
        printf("\nAinda nao existem planos malignos Dr. Doofenshmirtz!\n");
    }
    else

```

```

        {
            printf("    ID\t        Titulo\t        Descricao\n");
            for (i = 0; i < x; i++)
            {
                printf("\n    %i", dados[i].id);
                printf("        %s ", dados[i].titulo);

                for (j = 0; j < dados[i].tamanho_texto; j++)
                {
                    if (j == 30 || j == 60 || j == 90 || j == 120 || j == 150 || j == 180)
                    {
                        printf("\n\n\t\t\t\t\t");
                    }
                    printf("%c", dados[i].texto[j]);
                    if (j == dados[i].tamanho_texto - 1)
                    {
                        printf("        (%i/200)", dados[i].tamanho_texto);
                    }
                }
                printf("\n-----");
            }
            break;
        case 3:
            printf("\nVoce saiu\n");
            printf("Boa sorte sr. Malvado Doofenshmirtz!");
            break;

        default:
            break;
    }

}
while (resposta != 3);

return 0;
}

```