



**CENTRO UNIVERSITÁRIO INTERNACIONAL UNINTER**  
**ESCOLA SUPERIOR POLITÉCNICA**  
**BACHARELADO EM ENGENHARIA DA COMPUTAÇÃO**  
**DISCIPLINA DE LINGUAGEM DE PROGRAMAÇÃO**

## **ATIVIDADE PRÁTICA**

**RAPHAELL MACIEL DE SOUSA – RU: 1234567**  
**PROF. WINSTON SEN LUN FUNG**

**CAJAZEIRAS – PARAÍBA**  
**2020**

## 1 EXERCÍCIO

ENUNCIADO: Vamos supor que várias pedras do jogo de xadrez estão no tabuleiro. Construa um programa, em linguagem C, que:

- Solicite ao usuário que informe qual peça deseja colocar em cada posição do tabuleiro.
- Mostre o tabuleiro informado pelo usuário na tela.
- Determine a qual a quantidade de cada tipo de peça no tabuleiro.

```
/*
Descrição: Solução do exercício 1 da atividade prática da disciplina de linguagem de
programação
Author: Raphael Maciel de Sousa
Data: 22/10/2020
*/

#include <stdio.h>
#include <stdlib.h>
#include <locale.h> //biblioteca para definição de idioma de saída de dados
#include <ctype.h> // contém funções e macros para manipulação de caracteres

//Definição de variáveis
int tabuleiro[8][8];
char player[50];
bool control_out = 0;

int peoes = 0;
int cavalos = 0;
int torres = 0;
int bispos = 0;
int reis = 0;
int rainhas = 0;

void print_menu();
void read_user_choices(char option);
void reset_tabuleiro();
void read_tabuleiro();
void show_tabuleiro();

int main()
{
    setlocale(LC_ALL, "Portuguese"); //Define a lingua portuguesa para saída de da-
dos
    printf("Olá, vamos iniciar...\n");
    printf("Esse é o nosso xadrez 1.0\n");

    reset_tabuleiro(); //Inicialmente, vamos zerar todas as posições do tabuleiro

    while(control_out == 0)
    {
        printf("\n\nEscolha uma das seguintes opções\n\n");
        print_menu(); //chamada da função de apresentação do menu

        char option = getchar();

        /* Ignora o resto da linha */
        while (getchar() != '\n');
```

```

        printf("\n\nSua escolha foi [%c], vamos nessa!\n", toupper(option));

        read_user_choices(toupper(option)); //chama a função que seleciona o que
o usuário deseja realizar
    }

    system("Pause"); //Não é necessário na versão 2019 do visual Studio
    return 0;
}

//Função para apresentação do menu
void print_menu()
{
    printf("[N] - Inserir nome do jogador\n");
    printf("[I] - Inserir peças no tabuleiro\n");
    printf("[M] - Mostra tabuleiro\n");
    printf("[R] - Reset tabuleiro\n");
    printf("[Q] - Quantidade de peças\n");
    printf("[S] - Sair\n\n");
}

//Função de reset do tabuleiro, escreve zero em todas as posições da matriz
void reset_tabuleiro()
{
    for (int i = 0; i < 8; i++)
    {
        for (int j = 0; j < 8; j++)
        {
            tabuleiro[i][j] = 0;
        }
    }
    peoes = 0;
    cavalos = 0;
    torres = 0;
    bispos = 0;
    reis = 0;
    rainhas = 0;
}

//Solicita do usuário o posicionamento das peças do xadrez
void read_tabuleiro()
{
    printf("\nOk, vamos preencher o tabuleiro com as posições das peças...\n\n");
    for (int i = 0; i < 8; i++)
    {
        printf("\n*****linha %d *****\n\n", i + 1);

        for (int j = 0; j < 8; j++)
        {
            printf("Input [%d][%d]: ", i+1, j+1);
            int input;
            scanf_s("%d", &input);
            while (getchar() != '\n');
            tabuleiro[i][j] = input;

            //Condições para contagem da quantidade de peças
            if (input == 1)
                peoes = peoes + 1;
            if (input == 2)
                cavalos = cavalos + 1;
            if (input == 3)

```

```

        torres = torres + 1;
    if (input == 4)
        bispos = bispos + 1;
    if (input == 5)
        reis = reis + 1;
    if (input == 6)
        rainhas = rainhas + 1;
    }
}

// mostrando a matriz na tela
printf("\n\nO tabuleiro foi organizado da seguinte forma: \n\n");
for (int i = 0; i < 8; i++)
{
    for (int j = 0; j < 8; j++)
    {
        printf("%d \t", tabuleiro[i][j]);
    }
    printf("\n\n");
}

//Esta função exibe no prompt o tabuleiro
void show_tabuleiro()
{
    for (int i = 0; i < 8; i++)
    {
        for (int j = 0; j < 8; j++)
        {
            printf("%d \t", tabuleiro[i][j]);
        }
        printf("\n\n");
    }
}

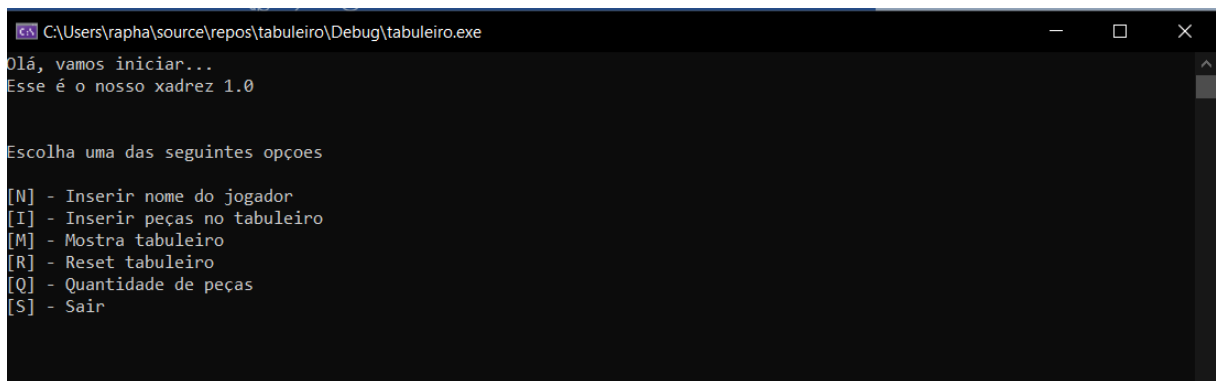
//Função de seleção de tarefas escolhidas pelo usuário
void read_user_choices(char option)
{
    switch (option)
    {
        case 'N':
            printf("\nInsira o nome do Jogador\n\n");
            scanf_s("%s", &player, 50);
            while (getchar() != '\n'); // Ignora o resto da linha
            printf("\nSeja Bem vindo %s\n\n", player);
            break;
        case 'I':
            printf("\nOk, vamos inserir as peças do tabuleiro...");
            printf("\n1 - Peão \t4 - Bispos \t0 - Ausência de Peças");
            printf("\n2 - Cavalos \t5 - Reis");
            printf("\n3 - Torres \t6 - Rainhas\n\n");
            read_tabuleiro();
            break;
        case 'M':
            printf("\n\nOk, vamos ver como está a distribuição de peças no tabu-
leiro\n\n");
            show_tabuleiro();
            break;
        case 'R':
            reset_tabuleiro();
            break;
        case 'Q':
    
```

```

        printf("\n\nA quantidade de peças escolhidas é:\n\n");
        printf("%d Peões\n", peoes);
        printf("%d Cavalos\n", cavalos);
        printf("%d Torres\n", torres);
        printf("%d Bispos\n", bispos);
        printf("%d Reis\n", reis);
        printf("%d Rainhas\n\n", rainhas);
        break;
    case 'S':
        control_out = 1;
        printf("\n\nAté a próxima!\n\n");
        break;
    default: printf("\nOpção Inválida!\n\n");
            break;
}
}
}

```

Imagem do código funcionando no seu computador:



```

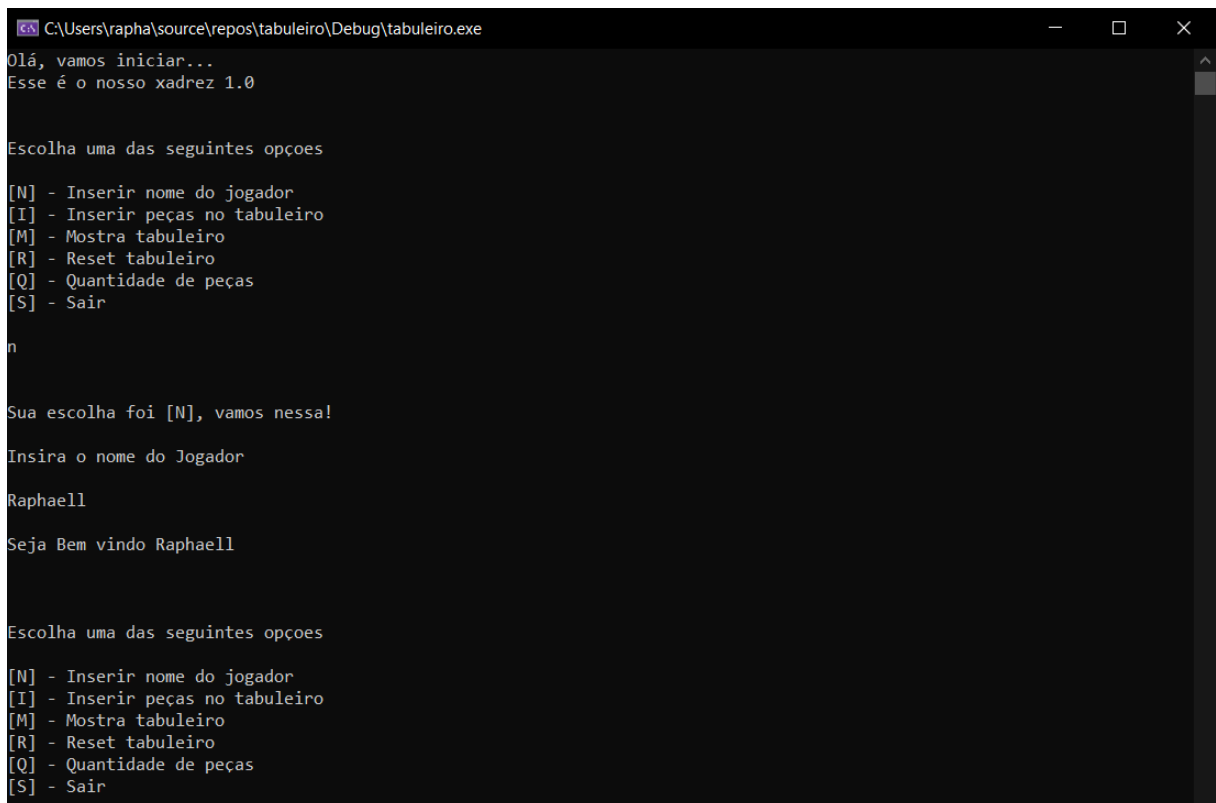
C:\Users\rapha\source\repos\tabuleiro\Debug\tabuleiro.exe
Olá, vamos iniciar...
Esse é o nosso xadrez 1.0

Escolha uma das seguintes opções

[N] - Inserir nome do jogador
[I] - Inserir peças no tabuleiro
[M] - Mostra tabuleiro
[R] - Reset tabuleiro
[Q] - Quantidade de peças
[S] - Sair

```

Figura 1: prompt com a execução do código desenvolvido. Menu de opções.



```

C:\Users\rapha\source\repos\tabuleiro\Debug\tabuleiro.exe
Olá, vamos iniciar...
Esse é o nosso xadrez 1.0

Escolha uma das seguintes opções

[N] - Inserir nome do jogador
[I] - Inserir peças no tabuleiro
[M] - Mostra tabuleiro
[R] - Reset tabuleiro
[Q] - Quantidade de peças
[S] - Sair
n

Sua escolha foi [N], vamos nessa!

Insira o nome do Jogador

Raphaell

Seja Bem vindo Raphaell

Escolha uma das seguintes opções

[N] - Inserir nome do jogador
[I] - Inserir peças no tabuleiro
[M] - Mostra tabuleiro
[R] - Reset tabuleiro
[Q] - Quantidade de peças
[S] - Sair

```

Figura 2: prompt com a execução do código desenvolvido. Inserindo o nome do jogador.

```
C:\Users\rappa\source\repos\tabuleiro\Debug\tabuleiro.exe

[N] - Inserir nome do jogador
[I] - Inserir peças no tabuleiro
[M] - Mostra tabuleiro
[R] - Reset tabuleiro
[Q] - Quantidade de peças
[S] - Sair

i

Sua escolha foi [I], vamos nessa!

Ok, vamos inserir as peças do tabuleiro...
1 - Peao      4 - Bispos      0 - Ausência de Peças
2 - Cavalos   5 - Reis
3 - Torres    6 - Rainhas

Ok, vamos preencher o tabuleiro com as posições das peças...

*****linha 1 *****
Input [1][1]: 1
Input [1][2]: 2
Input [1][3]: 3
Input [1][4]: 1
Input [1][5]: 2
Input [1][6]: 0
Input [1][7]: 1
Input [1][8]: 1

*****linha 2 *****
Input [2][1]:
```

Figura 3: prompt com a execução do código desenvolvido. Inserindo as peças no tabuleiro.

```
C:\Users\rappa\source\repos\tabuleiro\Debug\tabuleiro.exe

Input [8][5]: 3
Input [8][6]: 0
Input [8][7]: 0
Input [8][8]: 0

O tabuleiro foi organizado da seguinte forma:

1      2      3      1      2      0      1      1
1      2      1      0      0      0      0      0
5      4      6      1      2      1      0      0
0      0      1      1      2      0      3      5
0      0      0      0      0      0      0      0
1      1      1      5      4      1      0      0
0      0      0      0      0      0      0      0
1      1      1      2      3      0      0      0

Escolha uma das seguintes opções

[N] - Inserir nome do jogador
[I] - Inserir peças no tabuleiro
[M] - Mostra tabuleiro
[R] - Reset tabuleiro
[Q] - Quantidade de peças
[S] - Sair
```

Figura 4: prompt com a execução do código desenvolvido. Apresentação do tabuleiro.

```
C:\Users\rapha\source\repos\tabuleiro\Debug\tabuleiro.exe
Escolha uma das seguintes opcoes

[N] - Inserir nome do jogador
[I] - Inserir peças no tabuleiro
[M] - Mostra tabuleiro
[R] - Reset tabuleiro
[Q] - Quantidade de peças
[S] - Sair

q

Sua escolha foi [Q], vamos nessa!

A quantidade de peças escolhidas é:

17 Peoes
6 Cavalos
3 Torres
2 Bispos
3 Reis
1 Rainhas

Escolha uma das seguintes opcoes

[N] - Inserir nome do jogador
[I] - Inserir peças no tabuleiro
[M] - Mostra tabuleiro
[R] - Reset tabuleiro
[Q] - Quantidade de peças
[S] - Sair
```

Figura 5: prompt com a execução do código desenvolvido. Exibição da quantidade de peças.

```
C:\Users\rapha\source\repos\tabuleiro\Debug\tabuleiro.exe
[I] - Inserir peças no tabuleiro
[M] - Mostra tabuleiro
[R] - Reset tabuleiro
[Q] - Quantidade de peças
[S] - Sair

r

Sua escolha foi [R], vamos nessa!

Escolha uma das seguintes opcoes

[N] - Inserir nome do jogador
[I] - Inserir peças no tabuleiro
[M] - Mostra tabuleiro
[R] - Reset tabuleiro
[Q] - Quantidade de peças
[S] - Sair

q

Sua escolha foi [Q], vamos nessa!

A quantidade de peças escolhidas é:

0 Peoes
0 Cavalos
0 Torres
0 Bispos
0 Reis
0 Rainhas
```

Figura 6: prompt com a execução do código desenvolvido. Resetando o tabuleiro.

```
C:\Users\rapha\source\repos\tabuleiro\Debug\tabuleiro.exe
A quantidade de peças escolhidas é:
0 Peoes
0 Cavalos
0 Torres
0 Bispos
0 Reis
0 Rainhas

Escolha uma das seguintes opções
[N] - Inserir nome do jogador
[I] - Inserir peças no tabuleiro
[M] - Mostra tabuleiro
[R] - Reset tabuleiro
[Q] - Quantidade de peças
[S] - Sair

S

Sua escolha foi [S], vamos nessa!

Até a próxima!

Press any key to continue . . .
```

Figura 7: prompt com a execução do código desenvolvido. Saindo da aplicação.

## 2 EXERCÍCIO

ENUNCIADO: Escreva um programa, em LINGUAGEM C, que:

a) Solicite ao usuário a entrada de um registro que armazene na memória o Nome do funcionário e o seu salário.

b) Utilizando o salário informado calcule o imposto de renda devido e mostre na tela, conforme a tabela.

```
/*
Local: Cajazeiras - PB
Data: 24/10/2020
Autor: Raphael Maciel de Sousa
Descrição: programa para cálculo do imposto.
Solução do exercício 2 da atividade prática
*/

#include <stdio.h>
#include <stdlib.h>
#include <locale.h> //biblioteca para definição de idioma de saída de dados

struct cadastro_de_funcionarios /*Criando a struct*/
{
    char nome[50];
    float salario;
}; struct cadastro_de_funcionarios cadastro; /*variavel tipo strut
para refe-
rência ao
cadas-
tro_de_funcionarios*/
```



```

/*Declaração de funções*/
void print_table();
void calculo_faixa_salarial(float salario);
void calculo_do_imposto(float salario);

int *faixa;
int faixa_salarial;

int main()
{
    setlocale(LC_ALL, "Portuguese"); //Define a lingua portuguesa para saída de da

    printf("Olá, por favor, insira seu nome: \n\n");
    fflush(stdin);
    fgets(cadastro.nome, 50, stdin); //Leitura do nome

    printf("\n\nSeja bem vindo %s\n\n", cadastro.nome);
    while (1)
    {
        faixa = &faixa_salarial;

        printf("Vamos calcular o seu imposto, por favor, informe seu salários:
\n\n");
        scanf_s("%f", &cadastro.salario);

        /*Chamada de função para vizualização de dados no prompt*/
        print_table();

        /*Chamada de função para a definição da faixa salarial*/
        calculo_faixa_salarial(cadastro.salario);

        printf("Sua faixa salarial é: %d\n\n", *faixa);

        /*Chamada de função para o cálculo imposto devido*/
        calculo_do_imposto(cadastro.salario);
    }

    system("Pause");
    return 0;
}

void print_table()
{
    printf("\n*****\n");
    printf("Base de Cálculo Mensal em R$ \tAliquota em %\tFaixa\n\n");
    printf("Até 1.637,11 \t\t\t -\t\t1\n");
    printf("De 1637,12 até 2.456,50 \t 7,5\t\t2\n");
    printf("De 2.453,51 até 3.271,38 \t 15,0\t\t3\n");
    printf("De 3.271,39 até 4.087,65 \t 22,5\t\t4\n");
    printf("Acima de 4.087,65 \t\t 27,5\t\t5\n");
    printf("*****\n");
    printf("\n\n");
}

void calculo_faixa_salarial(float salario)
{
    if (salario <= 1.63711e3f)
        faixa_salarial = 1;
    if (salario > 1.63711e3f && salario <= 2.45350e3f)
        faixa_salarial = 2;
    if (salario > 2.45350e3f && salario <= 3.27138e3f)

```

```

        faixa_salarial = 3;
    if (salario > 3.27138e3f && salario <= 4.08765e3f)
        faixa_salarial = 4;
    if (salario > 4.08765e3f)
        faixa_salarial = 5;
}

void calculo_do_imposto(float salario)
{
    float imposto_devido = 0;
    if (*faixa == 1)
    {
        printf("\nVocê está isento de impostos!\n");
    }
    if (*faixa == 2)
    {
        imposto_devido = salario * 0.075;
        float liquido = salario - imposto_devido;
        printf("O salário bruto é de: R$ %.2f\n", salario);
        printf("Total de impostos retidos: R$ %.2f\n", imposto_devido);
        printf("O salário líquido é de: R$ %.2f\n", liquido);
    }
    if (*faixa == 3)
    {
        imposto_devido = salario * 0.15;
        float liquido = salario - imposto_devido;
        printf("O salário bruto é de: R$ %.2f\n", salario);
        printf("Total de impostos retidos: R$ %.2f\n", imposto_devido);
        printf("O salário líquido é de: R$ %.2f\n", liquido);
    }
    if (*faixa == 4)
    {
        imposto_devido = salario * 0.225;
        float liquido = salario - imposto_devido;
        printf("O salário bruto é de: R$ %.2f\n", salario);
        printf("Total de impostos retidos: R$ %.2f\n", imposto_devido);
        printf("O salário líquido é de: R$ %.2f\n", liquido);
    }
    if (*faixa == 5)
    {
        imposto_devido = salario * 0.275;
        float liquido = salario - imposto_devido;
        printf("O salário bruto é de: R$ %.2f\n", salario);
        printf("Total de impostos retidos: R$ %.2f\n", imposto_devido);
        printf("O salário líquido é de: R$ %.2f\n", liquido);
    }
    printf("\n\n");
}

```

```

C:\Users\rapha\source\repos\imposto\Debug\imposto.exe
Olá, por favor, insira seu nome:
Raphaell

Seja bem vindo Raphaell

Vamos calcular o seu imposto, por favor, informe seu salários:
3600

```

Figura 8: prompt com a execução do código desenvolvido. Entrada de dados.

```
CA:\Users\rapha\source\repos\imposto\Debug\imposto.exe
*****
Base de Cálculo Mensal em R$      Aliquota em %      Faixa
Até 1.637,11                      -                  1
De 1637,12 até 2.456,50           7,5                2
De 2.453,51 até 3.271,38          15,0               3
De 3.271,39 até 4.087,65          22,5               4
Acima de 4.087,65                 27,5               5
*****
Sua faixa salarial é: 4
O salário bruto é de: R$ 3600,00
Total de impostos retidos: R$ 810,00
O salário líquido é de: R$ 2790,00
```

Figura 9: prompt com a execução do código desenvolvido. Saída de dados, cálculo do imposto e salário líquido.

### 3 EXERCÍCIO

ENUNCIADO: Faça um programa, em linguagem C, que lê 5 frases, informadas pelo o usuário, de, no máximo, 50 caracteres cada uma e armazene-as em um arquivo. Mas, antes de gravar cada frase no arquivo, é necessário converter todas as suas letras para maiúsculas. O nome do arquivo será fornecido, via teclado, pelo usuário. A seguir, feche o arquivo. Reabra o arquivo, para leitura, exibindo na tela todas as frases convertidas.

```
#include <stdio.h>
#include <stdlib.h>
#include <locale.h> //biblioteca para definição de idioma de saída de dados
#include <ctype.h>

FILE* stream;
//using namespace std;
char fileName[10];

char frase1[50];
char frase2[50];
char frase3[50];
char frase4[50];
char frase5[50];

char FRASE1[50];
char FRASE2[50];
char FRASE3[50];
char FRASE4[50];
char FRASE5[50];

int i = 0;
char line[50];

int main()
{
    //string filename;
    setlocale(LC_ALL, "Portuguese"); //Define a lingua portuguesa para saída de da-
dos
    printf("Insira o nome do arquivo\n\n");
    fgets(fileName, 10, stdin); //Leitura do nome
    fflush(stdin);
    while (getchar() != '\n'); // Ignora o resto da linha
```

```

printf("Insira a primeira frase\n");
fgets(frase1, 50, stdin); //Leitura do nome
fflush(stdin);

// Loop para converter as letras em maiúsculas
while (frase1[i]) {
    FRASE1[i] = toupper(frase1[i]);
    //printf("%c", toupper(FRASE1[i]));
    i++;
}
i = 0;

printf("Insira a segunda frase\n");
fgets(frase2, 50, stdin); //Leitura do nome
fflush(stdin);

// Loop para converter as letras em maiúsculas
while (frase2[i]) {
    FRASE2[i] = toupper(frase2[i]);
    //printf("%c", toupper(FRASE1[i]));
    i++;
}
i = 0;

printf("Insira a terceira frase\n");
fgets(frase3, 50, stdin); //Leitura do nome
fflush(stdin);

// Loop para converter as letras em maiúsculas
while (frase3[i]) {
    FRASE3[i] = toupper(frase3[i]);
    //printf("%c", toupper(FRASE1[i]));
    i++;
}
i = 0;

printf("Insira a quarta frase\n");
fgets(frase4, 50, stdin); //Leitura do nome
fflush(stdin);

// Loop para converter as letras em maiúsculas
while (frase4[i]) {
    FRASE4[i] = toupper(frase4[i]);
    //printf("%c", toupper(FRASE1[i]));
    i++;
}
i = 0;

printf("Insira a quinta frase\n");
fgets(frase5, 50, stdin); //Leitura do nome
fflush(stdin);

// Loop para converter as letras em maiúsculas
while (frase5[i]) {
    FRASE5[i] = toupper(frase5[i]);
    //printf("%c", toupper(FRASE1[i]));
    i++;
}
i = 0;

//Cria um arquivo no modo escrita com o nome informado pelo usuário

```

```

fopen_s(&stream, fileName, "w+");

if (stream)
{
    fputs(FRASE1, stream);
    fputs(FRASE2, stream);
    fputs(FRASE3, stream);
    fputs(FRASE4, stream);
    fputs(FRASE5, stream);

    fclose(stream);
}

//Abre o arquivo criado no modo de leitura
fopen_s(&stream, fileName, "r");
if (stream)
{
    //Leitura dos dados do arquivo
    char str[50];
    fgets(str, 50, stream);
    printf("%s\n", str);

    char str2[50];
    fgets(str2, 50, stream);
    printf("%s\n", str2);

    char str3[50];
    fgets(str3, 50, stream);
    printf("%s\n", str3);

    char str4[50];
    fgets(str4, 50, stream);
    printf("%s\n", str4);

    char str5[50];
    fgets(str5, 50, stream);
    printf("%s\n", str5);

    fclose(stream);
}

return 0;
}

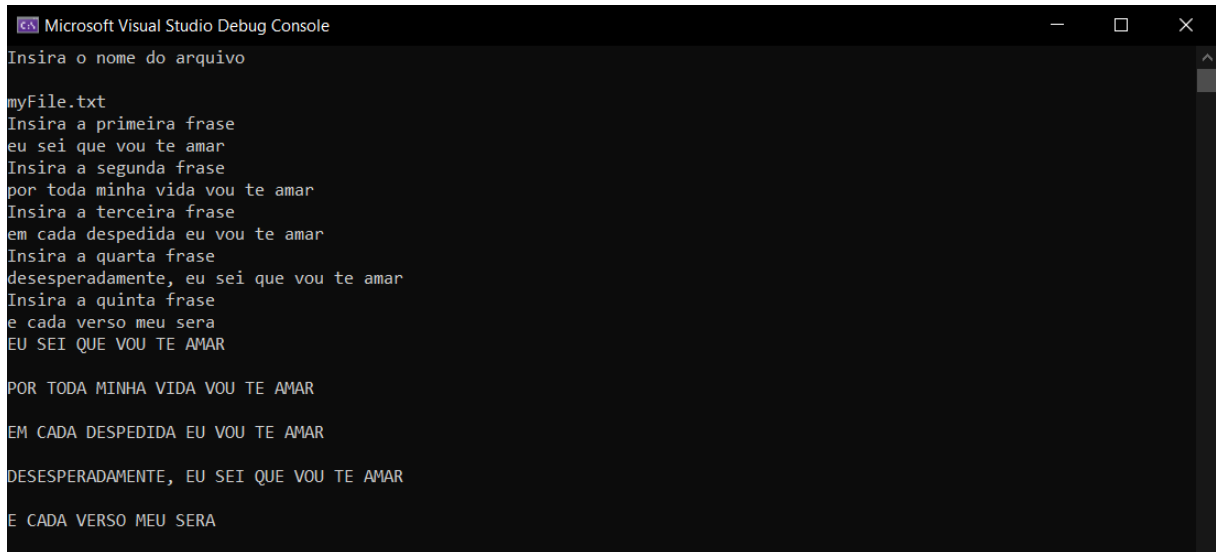
```



Figura 10: prompt com a execução do código desenvolvido. Inserindo o nome do arquivo para gravação dos dados.

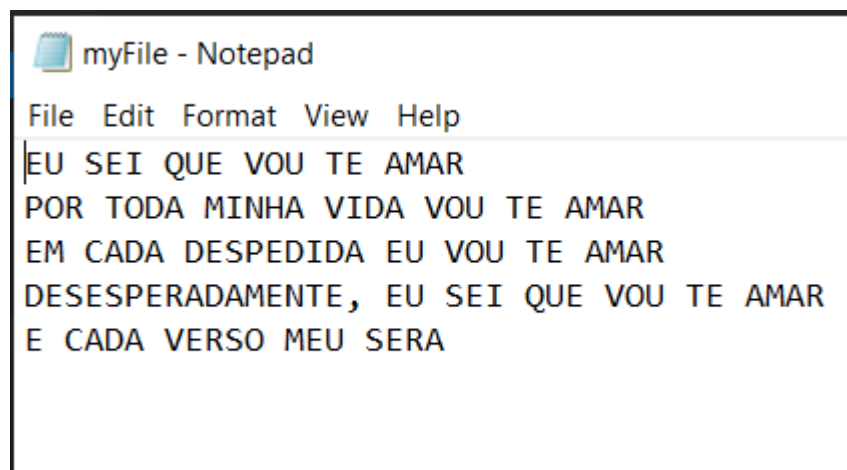


Figura 11: prompt com a execução do código desenvolvido. Inserindo o nome do arquivo para gravação dos dados.



```
Microsoft Visual Studio Debug Console
Insira o nome do arquivo
myFile.txt
Insira a primeira frase
eu sei que vou te amar
Insira a segunda frase
por toda minha vida vou te amar
Insira a terceira frase
em cada despedida eu vou te amar
Insira a quarta frase
desesperadamente, eu sei que vou te amar
Insira a quinta frase
e cada verso meu sera
EU SEI QUE VOU TE AMAR
POR TODA MINHA VIDA VOU TE AMAR
EM CADA DESPEDIDA EU VOU TE AMAR
DESESPERADAMENTE, EU SEI QUE VOU TE AMAR
E CADA VERSO MEU SERA
```

Figura 12: prompt com a execução do código desenvolvido. Inserindo os dados, convertendo o texto para maiúsculo, gravando em um arquivo .txt, realizando a leitura e apresentando para o usuário.



```
myFile - Notepad
File Edit Format View Help
EU SEI QUE VOU TE AMAR
POR TODA MINHA VIDA VOU TE AMAR
EM CADA DESPEDIDA EU VOU TE AMAR
DESESPERADAMENTE, EU SEI QUE VOU TE AMAR
E CADA VERSO MEU SERA
```

Figura 13: prompt com a execução do código desenvolvido. Arquivo .txt criado pelo usuário.

## 4 EXERCÍCIO

ENUNCIADO: Faça um programa, em linguagem C, para calcular a área e o perímetro de um hexágono. O programa deve implementar uma função chamada `calc_hexa` que calcula a área e o perímetro de um hexágono regular de lado `L`. O programa deve solicitar ao usuário o lado do polígono, calcular e imprimir a área e o perímetro do polígono. O programa termina quando for digitado um valor negativo qualquer para o lado. A função deve obedecer ao seguinte protótipo:

**`void calc_hexa(float L, float *area, float *perimetro);`**

```

#include <stdio.h>
#include <stdlib.h>
#include <locale.h> //biblioteca para definição de idioma de saída de dados
#include <math.h> /* pow */

float L, area, perimetro;

void calc_hexa(float L, float* area, float* perimetro);

int main()
{
    setlocale(LC_ALL, "Portuguese"); //Define a lingua portuguesa para saída de da-
dos
    printf("Por favor, informe o tamanho do lado L [em metros]:\n\n");
    scanf_s("%f", &L);

    /*Início do loop para chamada de função e entrada de dados*/
    while (L > 0) {
        calc_hexa(L, &area, &perimetro);
        printf("\nA area e perímetro do hexagono com lado %.2f [metros] é %.2f
[metros quadrados] e %.2f [metros], respectivamente\n\n", L, area, perimetro);

        printf("Por favor, informe o tamanho do lado L [em metros]:\n\n");
        scanf_s("%f", &L);
    }
    return 0;
}

/*Função de cálculo da área e perímetro do hexágono*/
void calc_hexa(float L, float* area, float* perimetro)
{
    *area = (3.0 * pow(L, L) * sqrt(3.0)) * 0.5;
    *perimetro = 6 * L;
}

```

```

C:\Users\rapha\source\repos\hexagono\Debug\hexagono.exe
Por favor, informe o tamanho do lado L [em metros]:
2
A area e perímetro do hexagono com lado 2,00 [metros] é 10,39 [metros quadrados] e 12,00 [metros], respectivamente
Por favor, informe o tamanho do lado L [em metros]:
10
A area e perímetro do hexagono com lado 10,00 [metros] é 25980762112,00 [metros quadrados] e 60,00 [metros], respectiv
amente
Por favor, informe o tamanho do lado L [em metros]:

```

Figura 14: prompt com a execução do código desenvolvido. Cálculo da área e perímetro do hexágono

## 5 EXERCÍCIO

ENUNCIADO: Faça um programa, em linguagem C, que:

- a) Crie uma struct chamada ponto2d que tenha como atributos os pontos x,y.
- b) Crie duas estruturas do tipo ponto2d chamadas ponto\_inicial e ponto\_final.
- c) Mostre um menu com as seguintes opções e implemente-as:

[1] - Digitar os valores do primeiro ponto

[2] - Digitar os valores do segundo ponto

[3] - Mostrar a distância entre os pontos

[4] - Sair

```
#include <stdio.h>
#include <math.h> /* pow and sqrt */
#include <locale.h> //biblioteca para definição de idioma de saída de dados

struct ponto2d {
    int x;
    int y;
};
void read_point(struct ponto2d* p);
float distancia(struct ponto2d ponto_inicial, struct ponto2d ponto_final);

void print_menu();
void read_user_choices(int option);

bool control_out = 0;
struct ponto2d ponto_inicial, ponto_final;

int main()
{
    setlocale(LC_ALL, "Portuguese"); //Define a lingua portuguesa para saída de dados
    int input;

    while (control_out == 0)
    {
        print_menu();
        scanf_s("%d", &input);
        while (getchar() != '\n');
        read_user_choices(input);
    }

    return 0;
}

//Função para apresentação do menu
void print_menu()
{
    printf("[1] - Digitar os valores do primeiro ponto\n");
    printf("[2] - Digitar os valores do segundo ponto\n");
    printf("[3] - Mostrar a distância entre os pontos\n");
    printf("[4] - Sair\n\n");
}

//Função de seleção de tarefas escolhidas pelo usuário
```



```

void read_user_choices(int option)
{
    switch (option)
    {
        case 1:
            printf("\nOpção 1\n");
            printf("\nInforme as coordenadas do ponto inicial:\n");
            read_point(&ponto_inicial);
            break;
        case 2:
            printf("\nOpção 2\n");
            printf("\nInforme as coordenadas do ponto final:\n");
            read_point(&ponto_final);
            break;
        case 3:
            printf("\nOpção 3\n");
            printf("\n\nDistancia entre os pontos: %.2f\n\n", distancia(ponto_inicial, ponto_final));
            break;
        case 4:
            printf("\nOpção 4\n");
            control_out = 1;
            break;
        default: printf("\nOpção Inválida!\n");
            break;
    }
}

void read_point(struct ponto2d* p)
{
    int x, y;
    scanf_s("%d%d", &x, &y);
    p->x = x;
    p->y = y;
}

float distancia(struct ponto2d ponto_inicial, struct ponto2d ponto_final)
{
    float term1, term2;
    term1 = pow((ponto_inicial.x - ponto_final.x), 2); /* Funcao pow(x,y) retorna
x^y */
    term2 = pow((ponto_inicial.y - ponto_final.y), 2);
    return sqrt(term1 + term2); /* Funcao sqrt(x) retorna a raiz quadrada de x */
}

```



```

C:\Users\rappa\source\repos\distancia\Debug\distancia.exe
[1] - Digitar os valores do primeiro ponto
[2] - Digitar os valores do segundo ponto
[3] - Mostrar a distância entre os pontos
[4] - Sair

```

Figura 15: prompt com a execução do código desenvolvido. Menu do usuário.

```
C:\Users\rapha\source\repos\distancia\Debug\distancia.exe
[1] - Digitar os valores do primeiro ponto
[2] - Digitar os valores do segundo ponto
[3] - Mostrar a distância entre os pontos
[4] - Sair
1
Opção 1
Informe as coodenadas do ponto inicial:
```

Figura 16: prompt com a execução do código desenvolvido. Informando primeira coordenada (x,y).

```
C:\Users\rapha\source\repos\distancia\Debug\distancia.exe
[1] - Digitar os valores do primeiro ponto
[2] - Digitar os valores do segundo ponto
[3] - Mostrar a distância entre os pontos
[4] - Sair
1
Opção 1
Informe as coodenadas do ponto inicial:
1
4
[1] - Digitar os valores do primeiro ponto
[2] - Digitar os valores do segundo ponto
[3] - Mostrar a distância entre os pontos
[4] - Sair
2
Opção 2
Informe as coodenadas do ponto final:
```

Figura 16: prompt com a execução do código desenvolvido. Informando segunda coordenada (x,y).

```
C:\Users\rapha\source\repos\distancia\Debug\distancia.exe
[1] - Digitar os valores do primeiro ponto
[2] - Digitar os valores do segundo ponto
[3] - Mostrar a distância entre os pontos
[4] - Sair

2

Opção 2

Informe as coodenadas do ponto final:
5
4

[1] - Digitar os valores do primeiro ponto
[2] - Digitar os valores do segundo ponto
[3] - Mostrar a distância entre os pontos
[4] - Sair

3

Opção 3

Distancia entre os pontos: 4,00

[1] - Digitar os valores do primeiro ponto
[2] - Digitar os valores do segundo ponto
[3] - Mostrar a distância entre os pontos
[4] - Sair
```

Figura 17: prompt com a execução do código desenvolvido. Cálculo da distância.

```
[1] - Digitar os valores do primeiro ponto
[2] - Digitar os valores do segundo ponto
[3] - Mostrar a distância entre os pontos
[4] - Sair

4

Opção 4

C:\Users\rapha\source\repos\distancia\Debug\distancia.exe (pr
To automatically close the console when debugging stops, enab
le when debugging stops.
Press any key to close this window . . .
```

Figura 17: prompt com a execução do código desenvolvido. Opção sair.