

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

# EXERCÍCIO

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

a) Solicite ao usuário que informe qual peça deseja colocar em cada posição do tabuleiro.

b) Mostre o tabuleiro informado pelo usuário na tela.

c) 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: Raphaell 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 dados

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 tabuleiro\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:

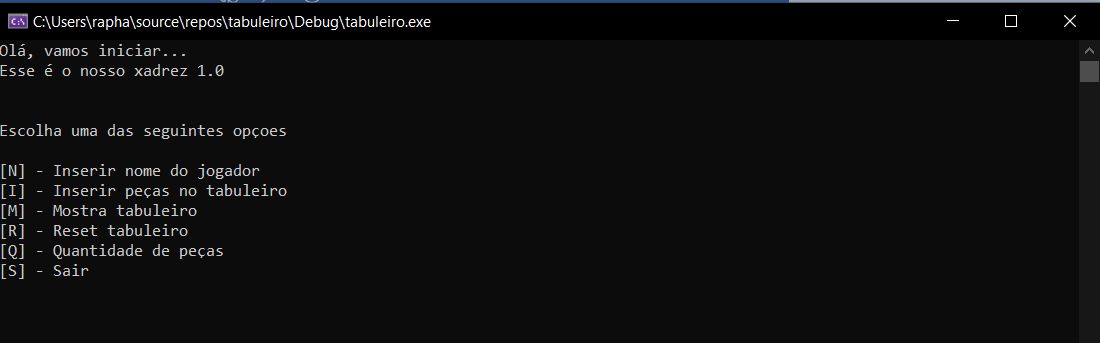


Figura 1: prompt com a execução do código desenvolvido. Menu de opções.

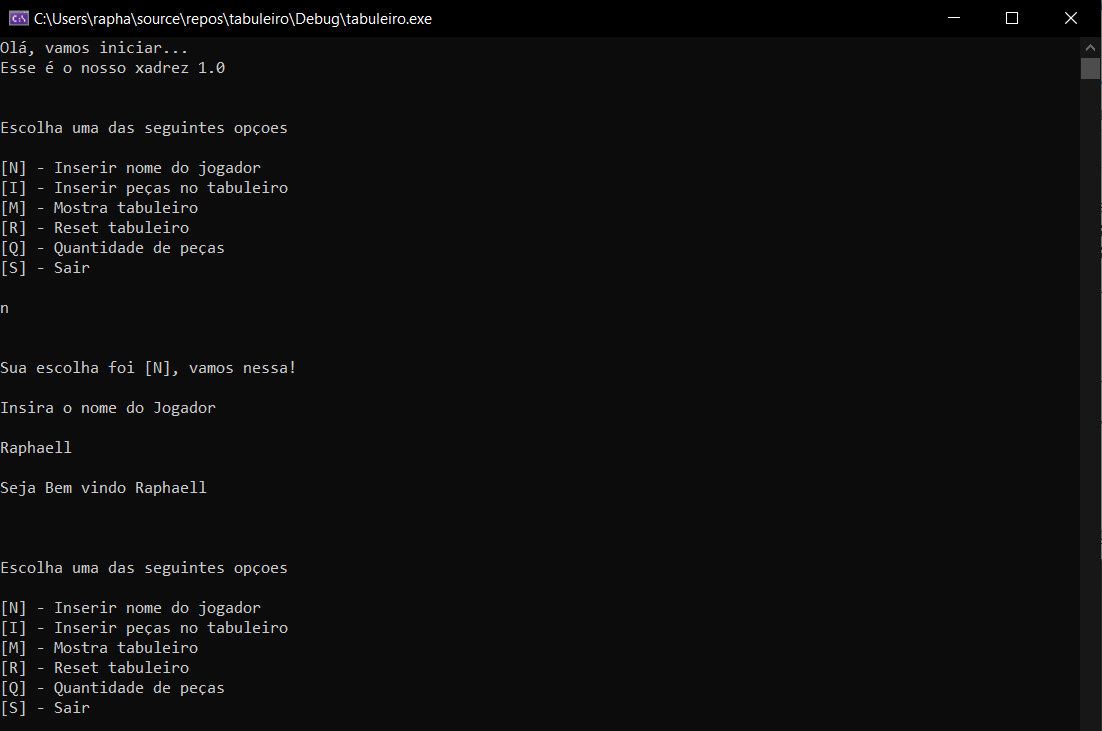


Figura 2: prompt com a execução do código desenvolvido. Inserindo o nome do jogador.

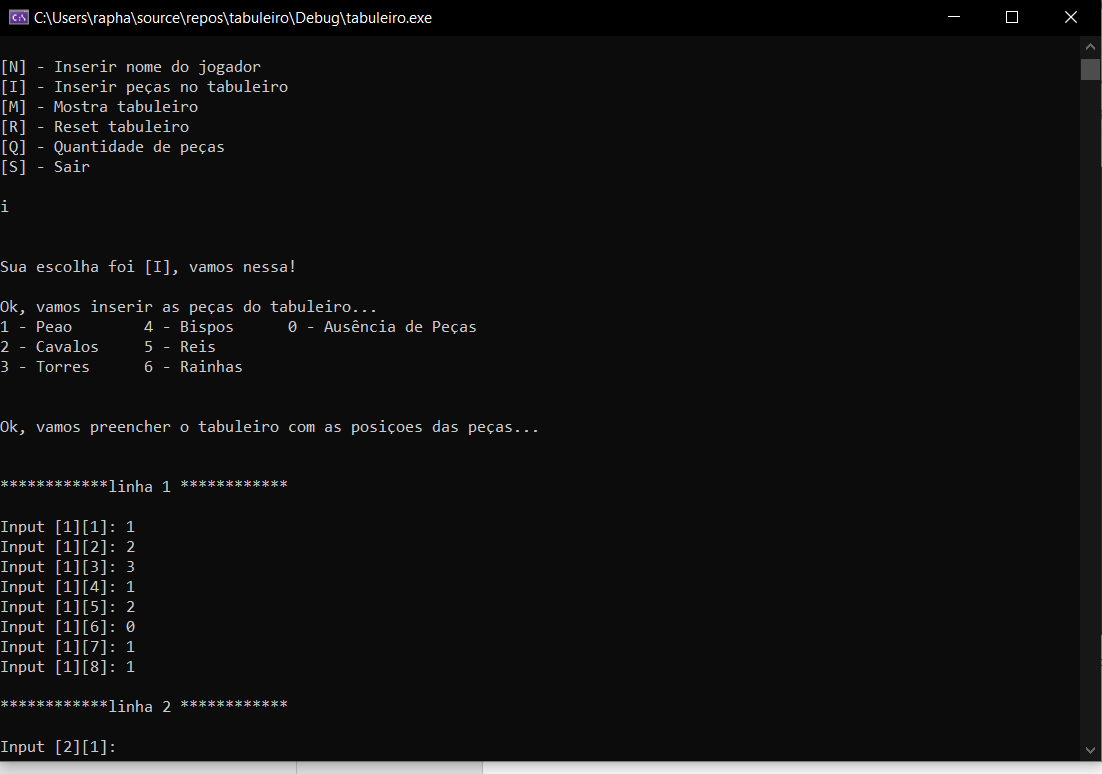


Figura 3: prompt com a execução do código desenvolvido. Inserindo as peças no tabuleiro.

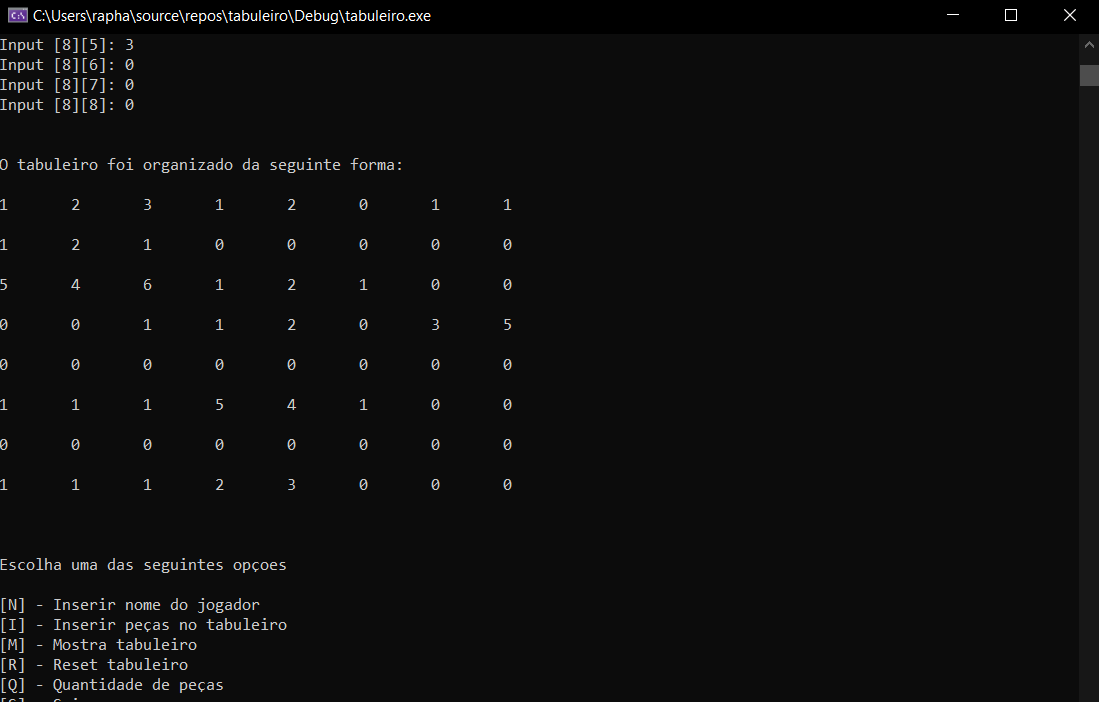


Figura 4: prompt com a execução do código desenvolvido. Apresentação do tabuleiro.

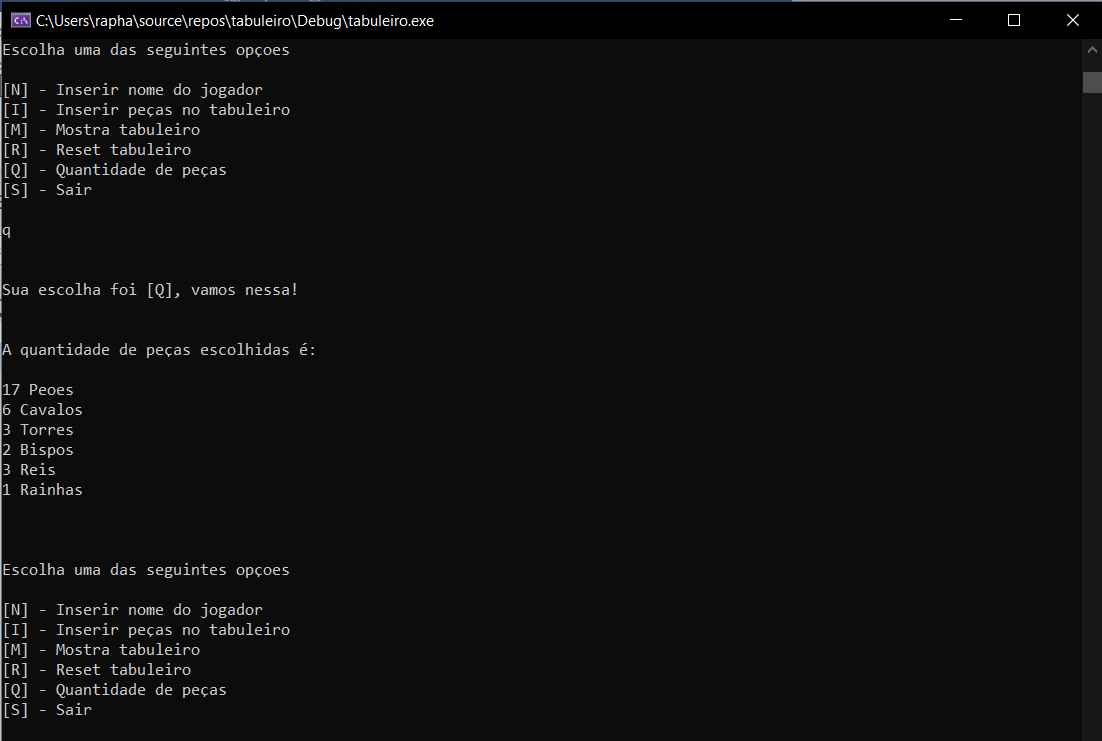


Figura 5: prompt com a execução do código desenvolvido. Exibição da quantidade de peças.

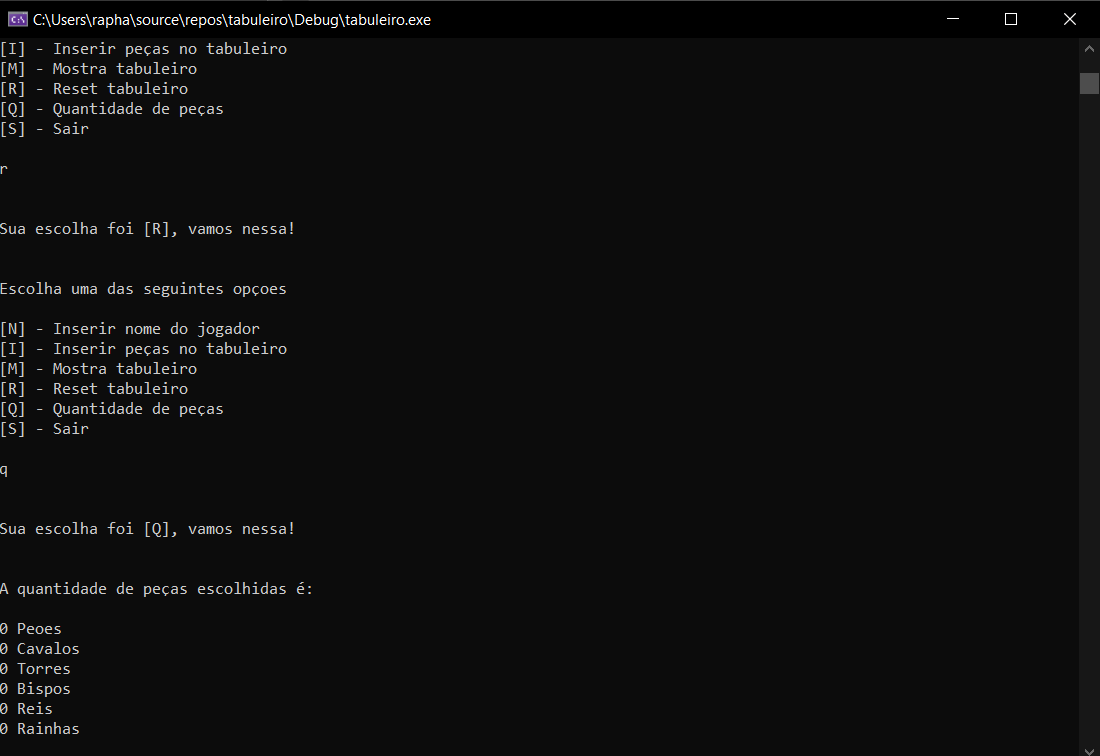


Figura 6: prompt com a execução do código desenvolvido. Resetando o tabuleiro.

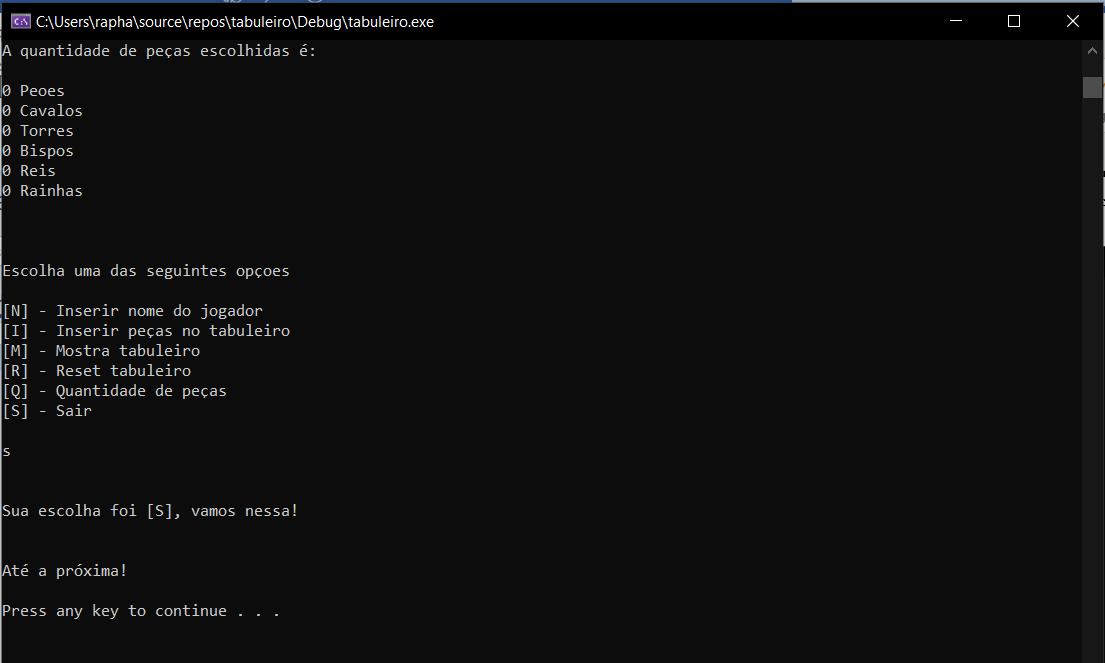


Figura 7: prompt com a execução do código desenvolvido. Saindo da aplicação.

# 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: Raphaell 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 referência ao

cadastro\_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("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

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");

}

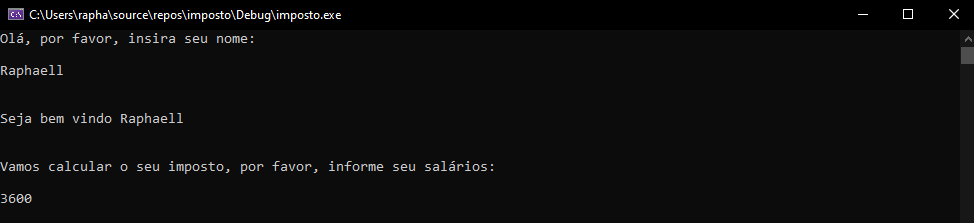


Figura 8: prompt com a execução do código desenvolvido. Entrada de dados.

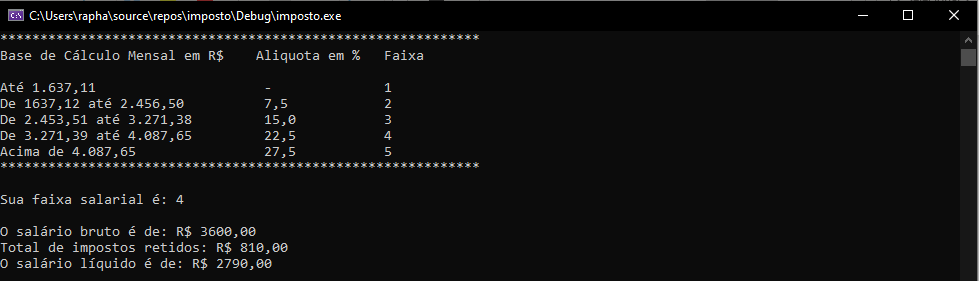


Figura 9: prompt com a execução do código desenvolvido. Saída de dados, cálculo do imposto e salário líquido.

# 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 dados

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.

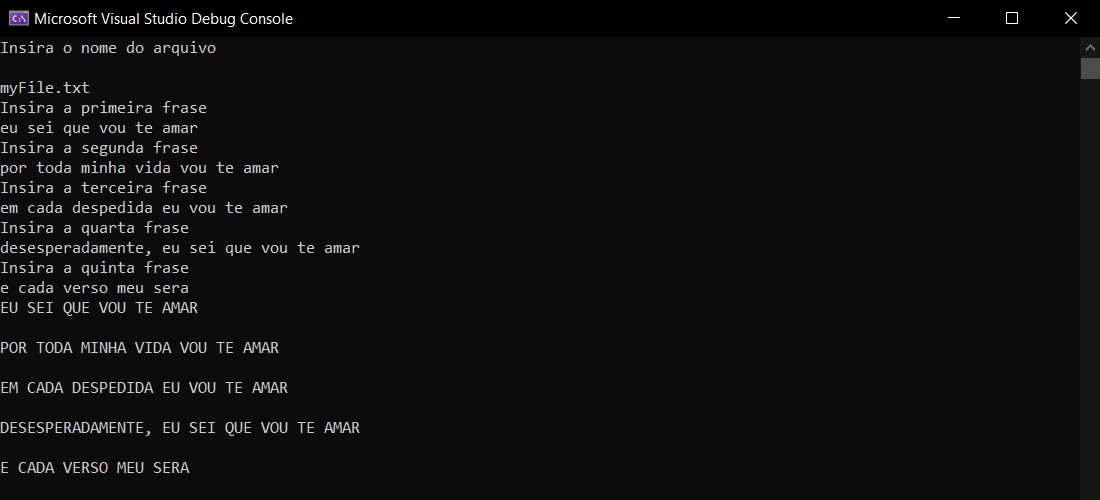


Figura 12: prompt com a execução do código desenvolvido. Inserindo os dados, convertendo o texto para maiúsculo, gravando em um arquivo .txt, realizando a leitura e apresentando para o usuário.

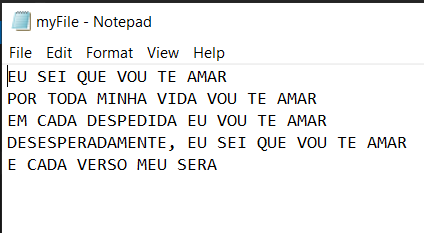


Figura 13: prompt com a execução do código desenvolvido. Arquivo .txt criado pelo usuário.

# 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 dados

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;

}

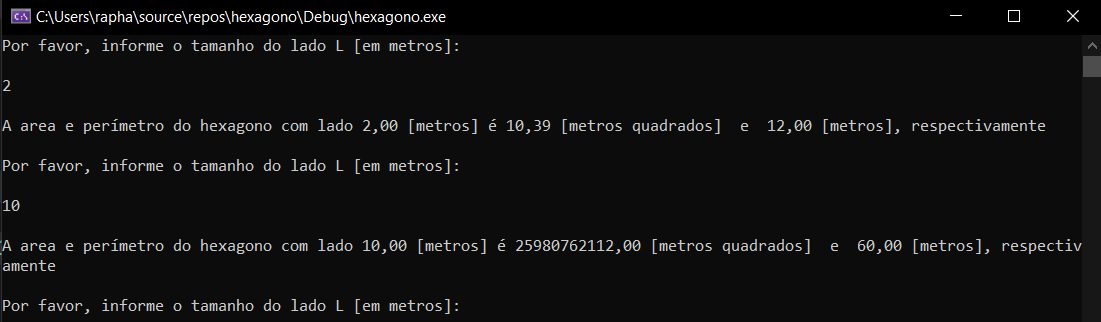


Figura 14: prompt com a execução do código desenvolvido. Cálculo da área e perímetro do hexágono

# 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 coodenadas do ponto inicial:\n");

read\_point(&ponto\_inicial);

break;

case 2:

printf("\nOpção 2\n");

printf("\nInforme as coodenadas 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 \*/

}

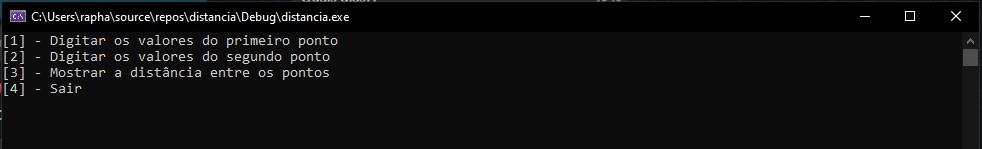


Figura 15: prompt com a execução do código desenvolvido. Menu do usuário.

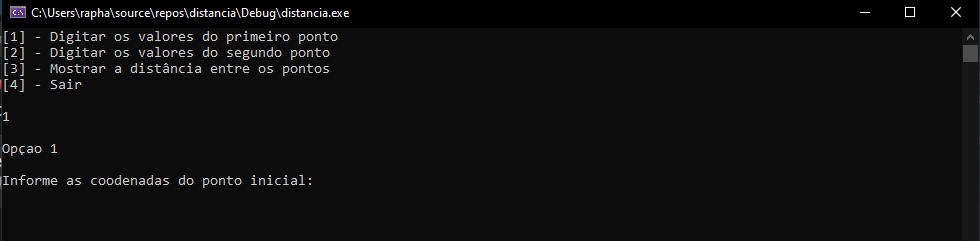


Figura 16: prompt com a execução do código desenvolvido. Informando primeira coordenada (x,y).

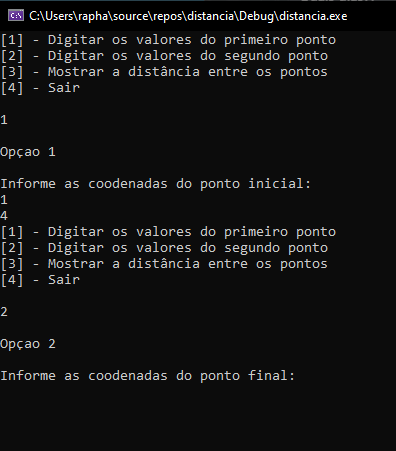


Figura 16: prompt com a execução do código desenvolvido. Informando segunda coordenada (x,y).

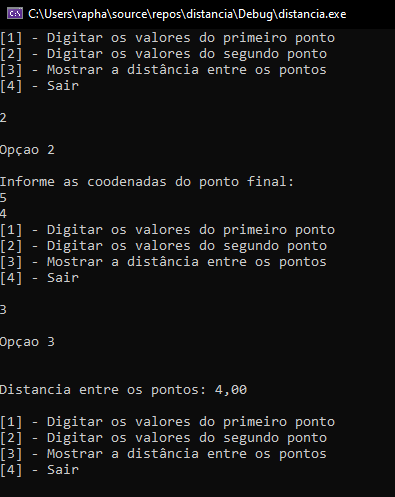


Figura 17: prompt com a execução do código desenvolvido. Cálculo da distância.

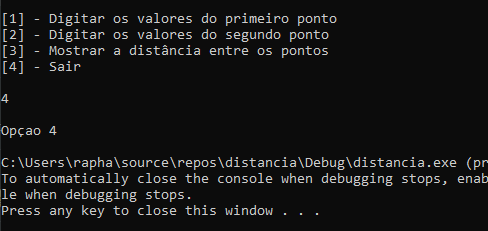


Figura 17: prompt com a execução do código desenvolvido. Opção sair.