

CENTRO UNIVERSITÁRIO INTERNACIONAL UNINTER

ESCOLA SUPERIOR POLITÉCNICA

BACHARELADO EM ENGENHARIA DA COMPUTAÇÃO

DISCIPLINA DE LINGUAGEM DE PROGRAMAÇÃO

ATIVIDADE PRÁTICA

nilzilene s araújo – RU: 3312120

PROF. WINSTON SEN LUN FUNG

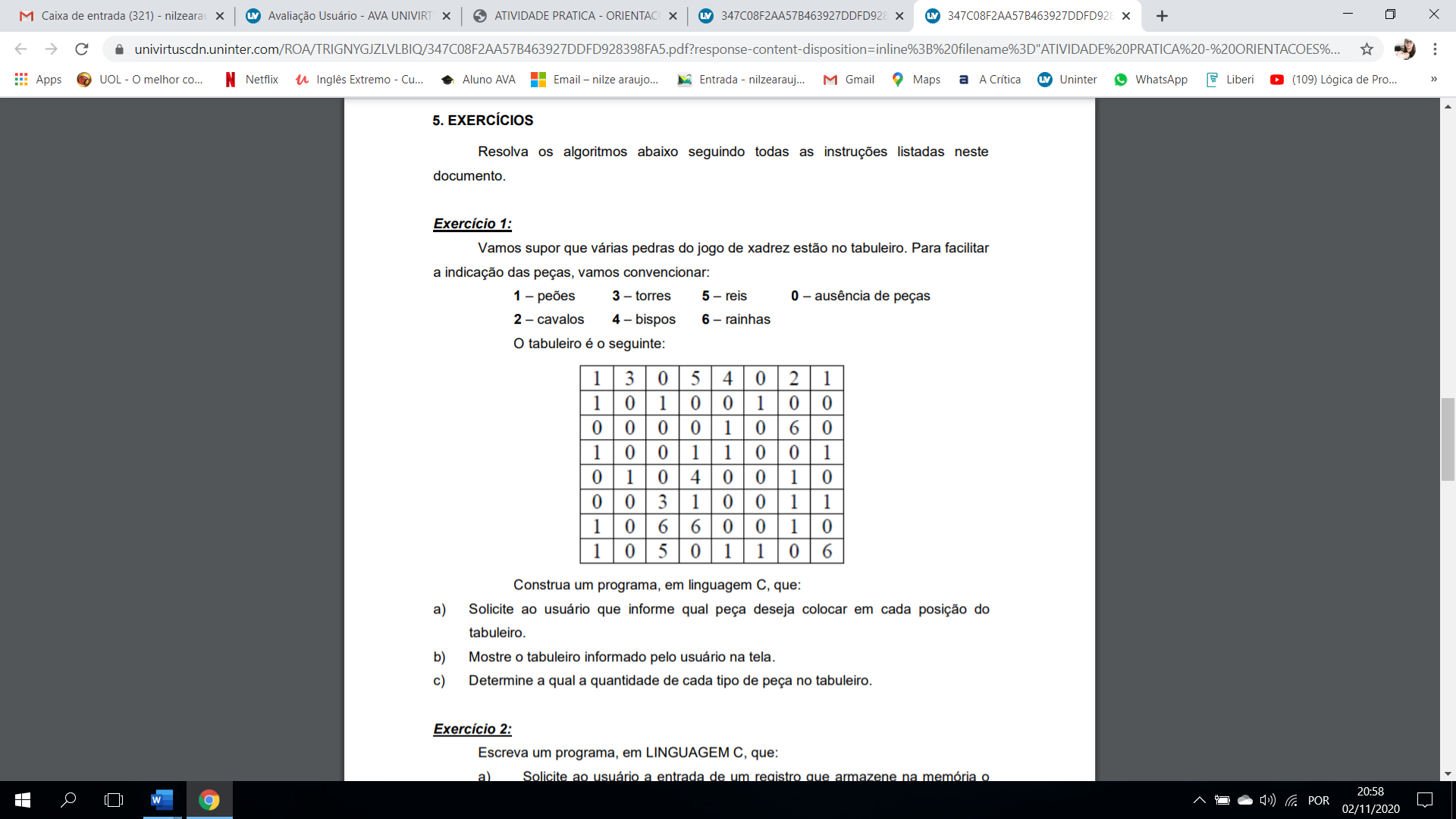
curitiba – Paraná

2020

Exercício 1:

Vamos supor que várias pedras do jogo de xadrez estão no tabuleiro. Para facilitar a indicação das peças, vamos convencionar: 1 – peões 3 – torres 5 – reis 0 – ausência de peças 2 – cavalos 4 – bispos 6 – rainhas

O tabuleiro é o seguinte:



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.

Solução:

#include <stdio.h>

#include <stdlib.h>

//Quantidade de cada peca

int peoes = 0;

int cavalos = 0;

int torres = 0;

int bispos = 0;

int reis = 0;

int rainhas = 0;

//Prototipos

void mostrar\_tabuleiro(int[8][8]);

int escolher\_peca();

void botar\_peca\_tabuleiro(int peca, int tabuleiro[8][8]);

int main()

{

       int tabuleiro[8][8];

//Inicializando a matriz

       for (int x = 0; x < 8; x++) {

             for (int y = 0; y < 8; y++) {

                    tabuleiro[x][y] = 0;

             }

       }

 //Loop principal

       while (1) {

             int opcao;

  //Menu

             printf("(1) Botar peça no tabuleiro\n");

             printf("(2) Mostrar tabuleiro\n");

             printf("(3) Quantidade de peças\n");

             printf("(4) Sair\n");

             printf("> ");

             scanf\_s("%d", &opcao);

             if (opcao == 1) {

                    printf("Escolha uma peça:\n");

                    int peca = escolher\_peca();

                    printf("Escolha onde botar no tabuleiro\n");

                    botar\_peca\_tabuleiro(peca, tabuleiro);

             }

             else if (opcao == 2) {

                    mostrar\_tabuleiro(tabuleiro);

             }

             else if (opcao == 3) {

                    printf("Quantidade de peças:\n");

                    printf("Peões: %d\nCavalos: %d\nTorres: %d\nBispos: %d\nReis: %d\nRainhas: %d\n\n", peoes, cavalos, torres, bispos, reis, rainhas);

             }

             else if (opcao == 4) {

                    break;

             }

             else {

                    printf("Opção inválida!\n");

             }

       }

       printf("\n");

       system("pause");

       return 1;

}

 //Funcao que mostra o tabuleiro na tela

void mostrar\_tabuleiro(int tabuleiro[8][8])

{

       printf("Peças:\n");

       printf("0: Vazio\n1: Peão\n2: Cavalo\n3: Torre\n4: Bispo\n5: Rei\n6: Rainha\n\n");

//For aninhado para percorrer toda matriz e mostrar na tela

       for (int x = 0; x < 8; x++) {

             for (int y = 0; y < 8; y++) {

                    int peca = tabuleiro[x][y];

                    printf("%d", peca);

             }

             printf("\n");

       }

       printf("\n");

}

 //Funcao para escolher a peca

int escolher\_peca()

{

       int peca = 0;

       printf("Peças:\n");

       printf("0: Vazio\n1: Peão\n2: Cavalo\n3: Torre\n4: Bispo\n5: Rei\n6: Rainha\n\n");

       printf("Escolha a peça:\n");

       printf(">");

       scanf\_s("%d", &peca);

       return peca;

}

 //Funcao para botar peca no tabuleiro

void botar\_peca\_tabuleiro(int peca, int tabuleiro[8][8])

{

       int x = 0;

       int y = 0;

       printf("Escolha uma posição valida para X (0-7)\n");

       scanf\_s("%d", &x);

//Verifica se a posicao é valida

       while (x < 0 || x > 7) {

             printf("Posição inválida! Por favor, um numero entre 0 e 7: ");

             scanf\_s("%d", &x);

       }

       printf("Escolha uma posição valida para Y (0-7)\n");

       scanf\_s("%d", &y);

       while (y < 0 || y > 7) {

             printf("Posição inválida! Por favor, um numero entre 0 e 7: ");

             scanf\_s("%d", &y);

       }

       //Aumenta peça escolhida

       if (peca == 1)peoes++;

       else if (peca == 2)cavalos++;

       else if (peca == 3)torres++;

       else if (peca == 4)bispos++;

       else if (peca == 5)reis++;

       else if (peca == 6)rainhas++;

       //Diminui peça substituida

       int peca\_atual = tabuleiro[x][y];

       if (peca\_atual == 1)peoes--;

       else if (peca\_atual == 2)cavalos--;

       else if (peca\_atual == 3)torres--;

       else if (peca\_atual == 4)bispos--;

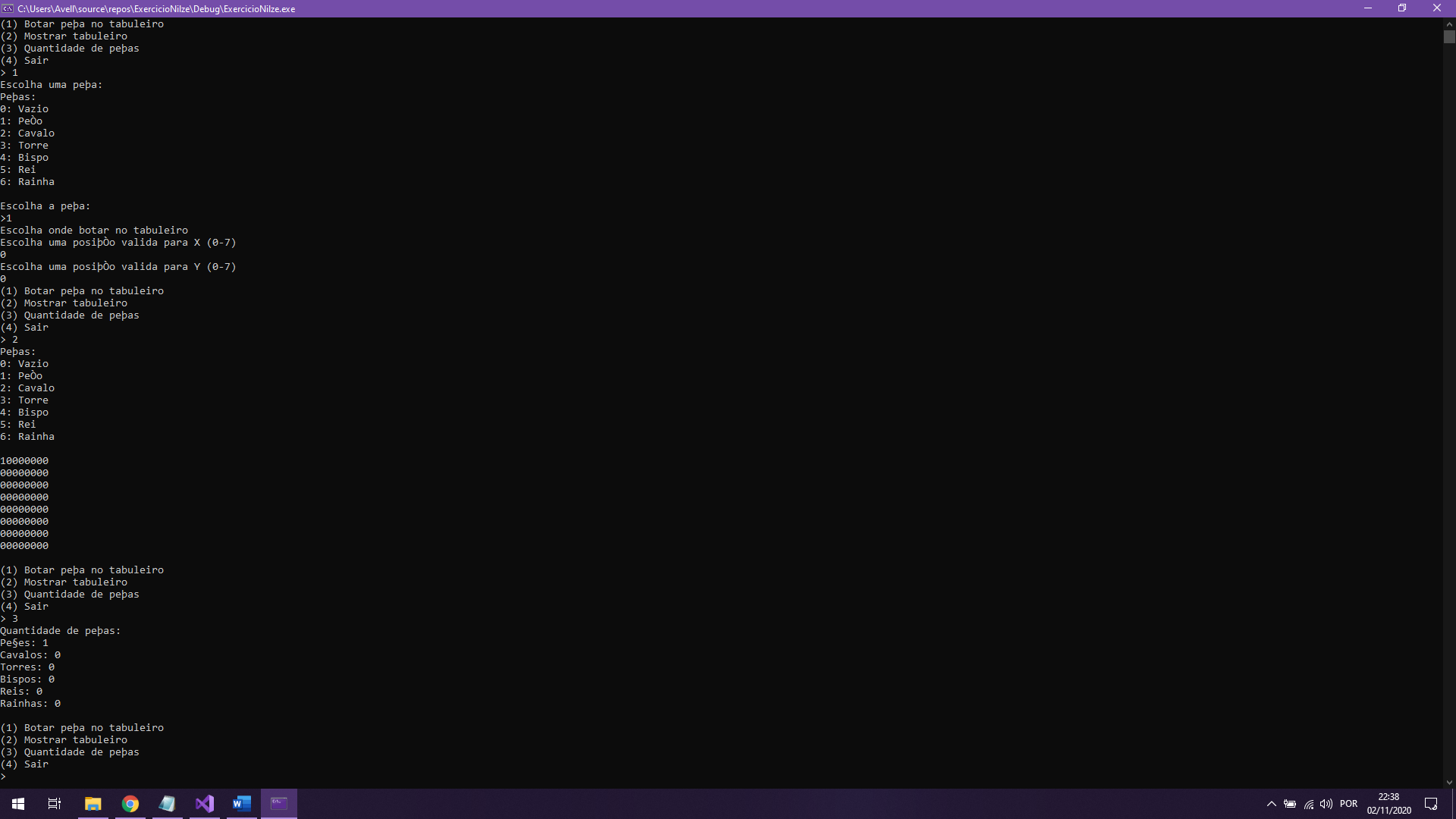
       else if (peca\_atual == 5)reis--;

       else if (peca\_atual == 6)rainhas--;

       tabuleiro[x][y] = peca;

}

exercício 1 - programa funcionando:

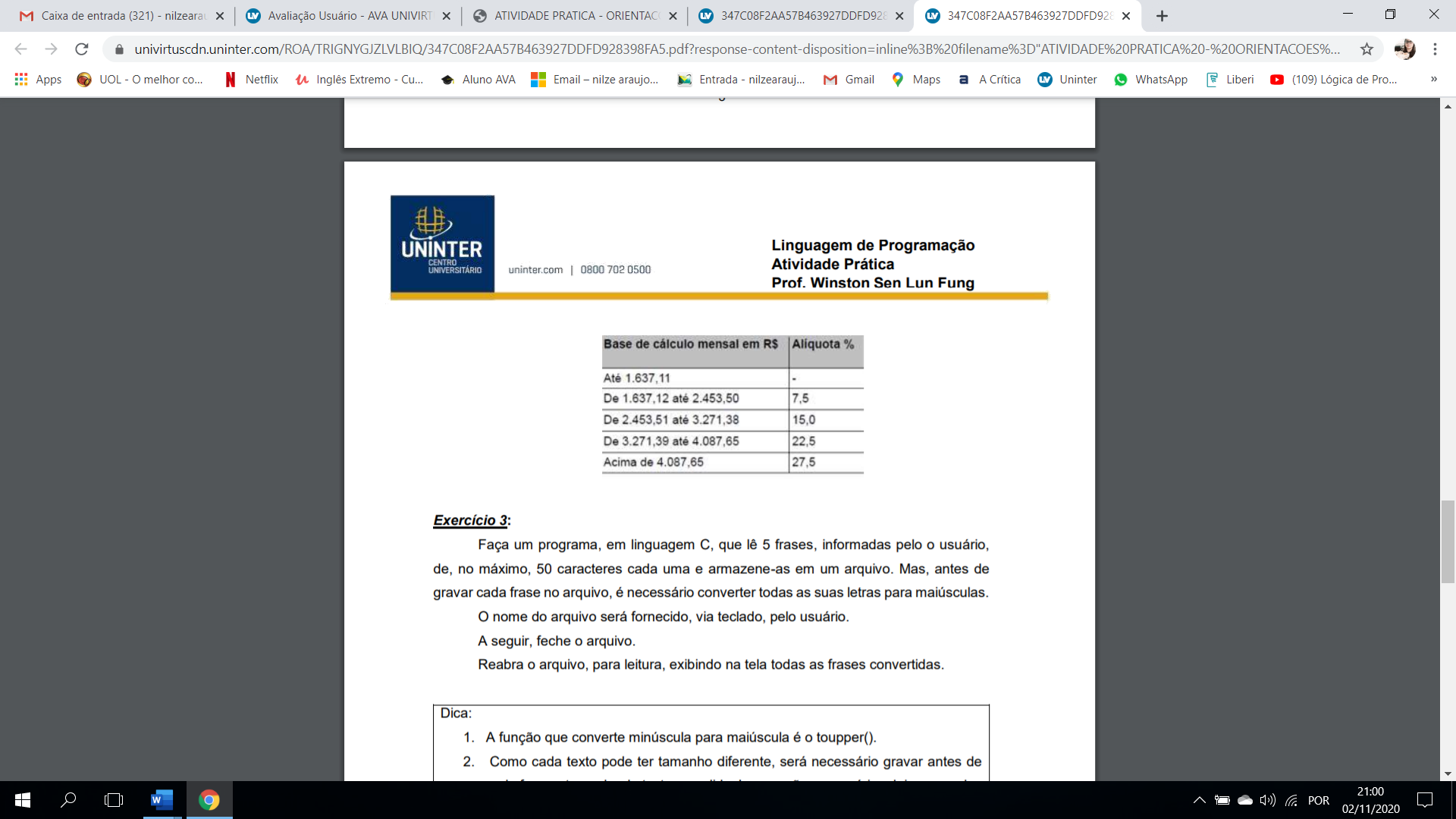


Exercício 2:

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 abaixo:



solução:

#include <stdio.h>

#include <stdlib.h>

int main()

{

       char nome[256];

       float salario, deducao, imposto\_devido;

       printf("Digite seu nome: ");

       scanf\_s("%s", nome, 256);

       while (1)

       {

             printf("Digite seu salario: ");

             scanf\_s("%f", &salario);

//Calcula a deducao de acordo com a faixa de salario

             if (salario <= 1637.11f) {

                    deducao = 0;

             }

             else if (salario <= 2453.50f) {

                    deducao = salario \* 0.075f;

             }

             else if (salario <= 3271.38f) {

                    deducao = salario \* 0.15f;

             }

             else if (salario <= 4087.65f) {

                    deducao = salario \* 0.225f;

             }

             else {

                    deducao = salario \* 0.25f;

             }

//Diminui do salario a deducao, para conseguir o imposto devido

             imposto\_devido = salario - deducao;

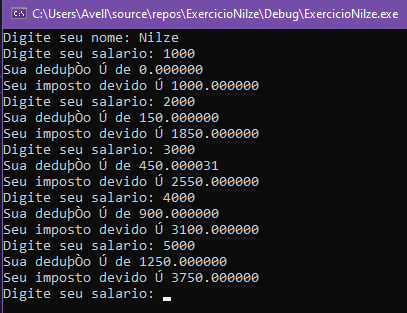
             printf("Sua dedução é de %f\n", deducao);

             printf("Seu imposto devido é %f\n", imposto\_devido);

       }

}

exercício 2 - programa funcionando:



Exercício 3:

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.

Exercício 4:

Faça um programa, em linguagem C, para calcular a área e o perímetro de um hexágono.

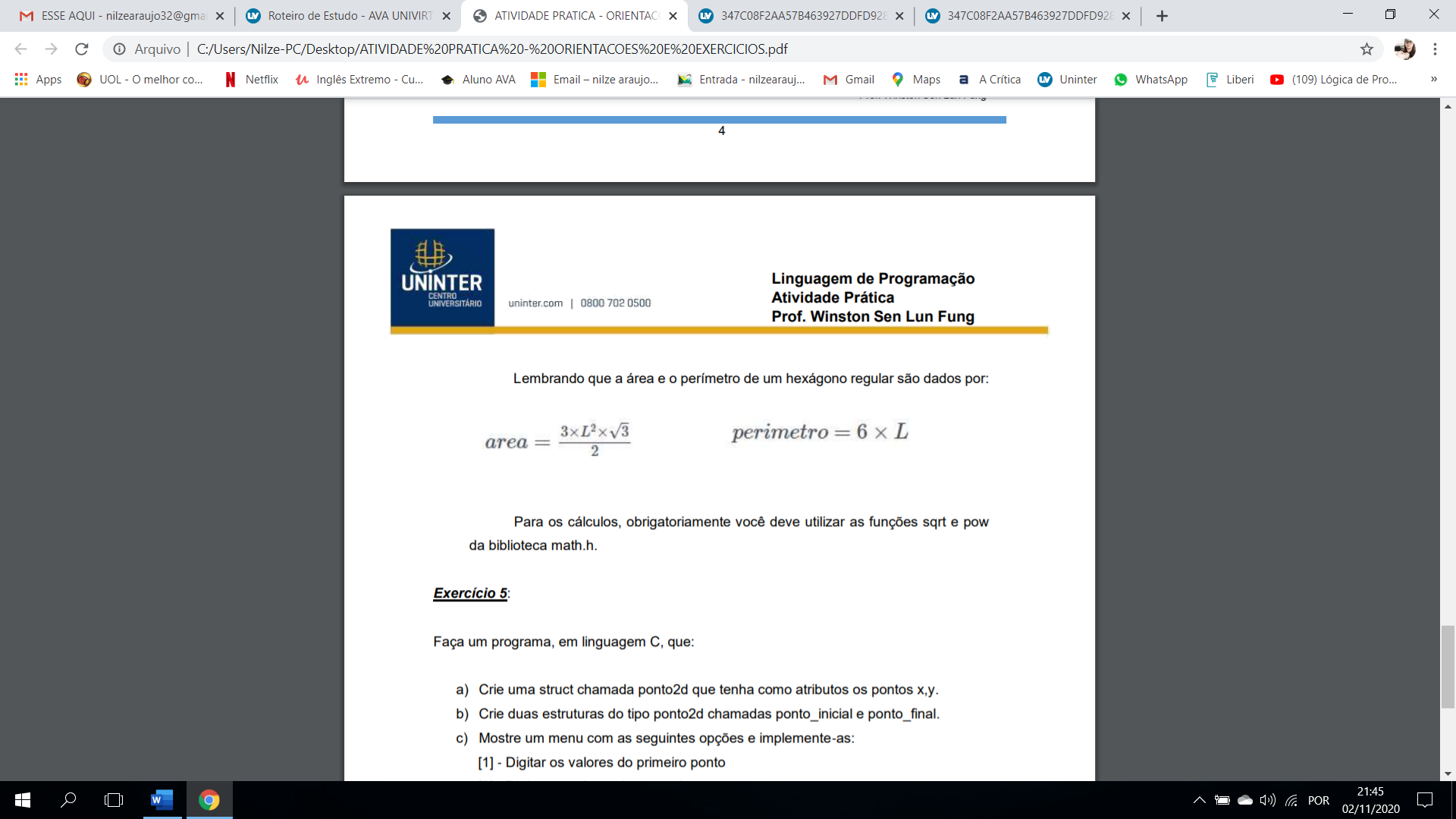
O programa deve implementar uma função chamada calc\_hexa que calcula a área e o perímetro de um hexágono regular de lado L.

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);

*Lembrando que a área e o perímetro de um hexágono regular são dados por:*

Para os cálculos, obrigatoriamente você deve utilizar as funções sqrt e pow da biblioteca math.h.

Solução:

#include <stdio.h>

#include <stdlib.h>

#include <math.h>

void calc\_hexa(float L, float\* area, float\* perimetro);

int main()

{

       float lado, area, perimetro;

       lado = area = perimetro = 0;

       while (lado >= 0)

       {

             printf("Digite o lado do hexagono: ");

             scanf\_s("%f", &lado);

             if (lado <= 0) continue;

             calc\_hexa(lado, &area, &perimetro);

             printf("A area é: %f e o perimetro é: %f\n", area, perimetro);

       }

}

void calc\_hexa(float L, float\* area, float\* perimetro)

{

       \*perimetro = 6 \* L;

  //Variaveis para facilitar o calculo

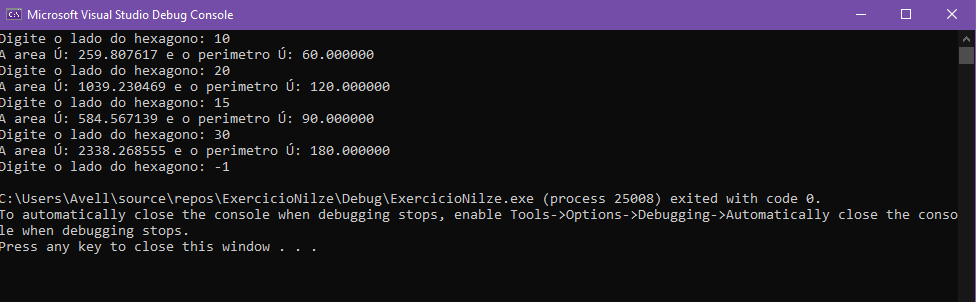
       float raiz\_de\_3 = sqrt(3);

       float lado\_quadrado = pow(L, 2);

//Calculo da area utilizando as variaveis anteriores

       \*area = (3 \* lado\_quadrado \* raiz\_de\_3) / 2;

}

exercício 4 - programa funcionando:

Exercício 5:

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

solução:

#include <stdio.h>

#include <stdlib.h>

#include <math.h>

//Definindo a struct

struct ponto2d {

       float x;

       float y;

};

//Prototipos

struct ponto2d definir\_ponto();

void mostrar\_distancia(struct ponto2d ponto\_inicial, struct ponto2d ponto\_final);

int main()

{

       //Definindo variaveis iniciais

       int opcao = 0;

       struct ponto2d ponto\_inicial = { 0,0 };

       struct ponto2d ponto\_final = { 0,0 };

       //Loop principal

       while (opcao != 4)

       {

             //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");

             scanf\_s("%d", &opcao);

             if (opcao == 1) {

                    ponto\_inicial = definir\_ponto();

             }

             else if (opcao == 2) {

                    ponto\_final = definir\_ponto();

             }

             else if (opcao == 3) {

                    mostrar\_distancia(ponto\_inicial, ponto\_final);

             }

             else if (opcao == 4) continue;

             else printf("Opcao invalida!");

       }

}

//Funcao para definir o ponto

struct ponto2d definir\_ponto()

{

       struct ponto2d ponto;

       printf("Defina o x do ponto: ");

       scanf\_s("%f", &ponto.x);

       printf("Defina o y do ponto: ");

       scanf\_s("%f", &ponto.y);

       return ponto;

}

//Funcao que calcula a distancia

void mostrar\_distancia(struct ponto2d ponto\_inicial, struct ponto2d ponto\_final)

{

       float distancia = 0;

       //Convertendo para float, pois pow e sqrt retornam double

       float calc\_x = (float)pow((float)ponto\_inicial.x - (float)ponto\_final.x, 2);

       float calc\_y = (float)pow((float)ponto\_inicial.y - (float)ponto\_final.y, 2);

       distancia = (float)sqrt((float)calc\_x + (float)calc\_y);

       printf("%f\n", distancia);

}

exercício 4 - programa funcionando:

