

CENTRO UNIVERSITÁRIO INTERNACIONAL UNINTER

ESCOLA SUPERIOR POLITÉCNICA

TECNÓLOGO EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

DISCIPLINA DE LÓGICA DE PROGRAMAÇÃO E ALGORITMOS

ATIVIDADE PRÁTICA

GESIANE G. F. PAJARINEN – RU: 2466550

PROF. VINICIUS POZZOBON BORIN, Me

IPATINGA – MINAS GERAIS

2020

# EXERCÍCIO 1

ENUNCIADO: Uma certa empresa fez uma pesquisa de mercado para saber se as pessoas gostaram ou não do seu último produto lançado. Para isto, coletou o sexo do entrevistado e a sua resposta (sim ou não). Sabendo que foram entrevistadas 150 pessoas, fazer um algoritmo que calcule e mostre ao final:

• O número de pessoas que responderam sim;

• O número de pessoas que responderam não;

• A percentagem de pessoas do sexo feminino que responderam sim;

• A percentagem de pessoas do sexo masculino que responderam não;

Para a resposta SIM/NÃO. Utilize uma variável do tipo CHAR, que armazena S ou N, ou use uma variável do tipo INT que armazena 1 (para SIM) e 2 (para NÃO).

Solução do aluno:

#include <stdio.h>

#include <stdlib.h>

int main(){

int i, sim = 0, nao= 0, mulher= 0, homem = 0;

char sexo, gostou, c; // esse c é para o 'while' que 'limpa' as entradas do 'scanf', se não me engano.

float porcentH, porcentM; //Float porque a porcentagem pode ter valor fracionado.

printf("\n REGISTRE OS DADOS:\n\n");

for (i=1; i <= 150; i++){ //i vai repetir o looping por 150 vezes.

printf("Sexo do cliente: F ou M\n");

scanf("%c", &sexo); //Recebe o valor f ou m.

while ((c= getchar())!= '\n' && c!=EOF){} //Limpeza de buffer.

printf("Cliente gostou: S OU N\n");

scanf("%c", &gostou); //Recebe o valor s ou n.

while ((c= getchar())!= '\n' && c!=EOF){} //Limpeza de buffer.

if (gostou == 'S'){ //Somando todos os 's' (SIM).

sim++;

if (sexo == 'F'){ //Somando as Mulheres que responderam SIM.

mulher++;}

}

else { //Somando os 'NAO'.

nao++;

if (sexo == 'M'){ //Somando os Homens que responderam NAO.

homem++;}

}

}

//sim+nao gera o total de pessoas que responderam ao questionário.

porcentM = (float)(mulher\*100)/ (sim+nao);

porcentH = (float)(homem\*100)/(sim+nao);

printf("\n \*\*\* RESULTADOS \*\*\*\n");

printf("\n - Número de pessoas que responderam SIM:\n %5d\n", sim);

printf("\n - Número de pessoas que responderam NÃO:\n %5d\n", nao);

printf("\n - Porcentagem de MULHERES que responderam SIM:\n %5.1f%%\n", porcentM);

//total de MULHERES que responderam SIM dentre todas as pessoas.

printf("\n - Porcentagem de HOMENS que responderam NÃO:\n %5.1f%%\n ", porcentH);

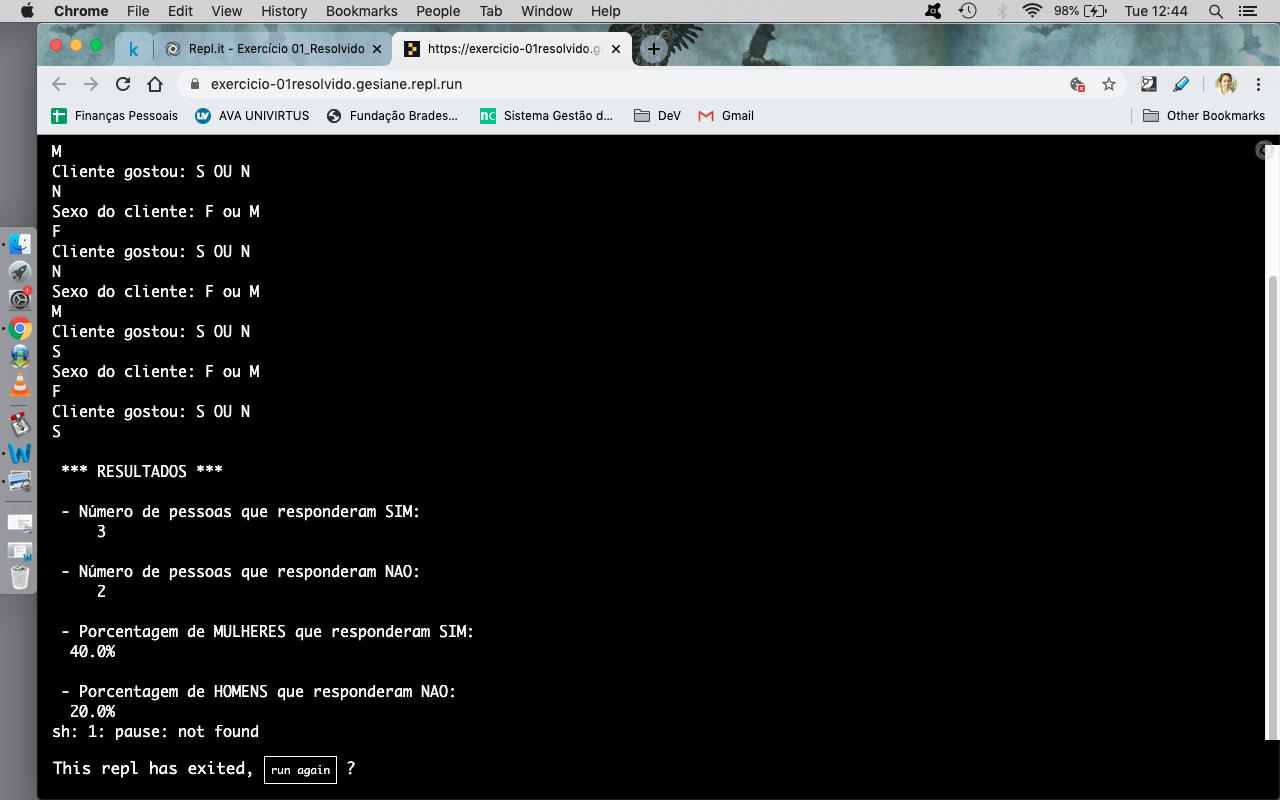
//Total de HOMENS que responderam NÃO dentre todas as pessoas.

system("pause");

return 0;

}

Imagem do código funcionando no seu computador:



O número de repetições para este teste foram só 5.

# EXERCÍCIO 2

ENUNCIADO: Para cada um dos consumidores de energia elétrica de uma cidade é informado o número da conta e o total de KW consumido no mês. Sabendo-se que o custo do KW é de R$ 1,75, fazer um algoritmo para:

* Armazenar e listar o número da conta, o total de KW consumidos e o valor a pagar de cada consumir cadastrado;
* Listar o número da conta, o total de KW consumidos e o valor a pagar do consumidor que mais gastou e o que menos gastou;
* Mostrar a média de consumo da cidade;
* Mostrar o número de consumidores que ultrapassaram o consumo de 170  KW;

Armazene as informações em estruturas de vetores e/ou matrizes. Na tela, deve existir um MENU que pergunta ao usuário se ele deseja cadastrar um novo consumidor ou listar alguma informação (maior, menor, média, etc.).

Solução do aluno:

#include <stdio.h>

#include <stdlib.h>

#define kw 1.75 //Queria colocar como constante

#define clientes 5 //Variável global

int c; //Pra fórmula do limpeza de buffer

//FUNÇÂO: Calcular o total a pagar:

float calculo(gasto){

float total;

total = kw \* gasto;

return total;

}

//FUNÇÃO: Imprimir a tabela dos clientes e suas respectivas informações

void imprimir(float matrix[clientes][3]){

printf("\n CLIENTES CADASTRADOS\n"); //Impressão da tabela de clientes já cadastrados

printf("\n------------------------------\n");

printf("Cadastro | Kw | A pagar");

printf("\n------------------------------\n");

for (int i= 0; i<clientes; i++){

for(int j=0; j<3; j++){

printf("%7.2f | ", matrix[i][j]);

}

printf("\n");

}

}

//FUNÇÂO: Quem gastou MAIS:

void mais\_gastou(float matrix[clientes][3]){

float maior;

for (int i=1; i<clientes; i++){ //Looping de comparacao entre os valores da matriz.

if (matrix[0][1] < matrix[i][1]){

matrix[0][1] = matrix[i][1]; //Se o valor do índice[0] for menor que o valor do índie[1], índice[0] recebe o valor de índie[1]. Caso contrário, índie[0] continua com seu valor.

matrix[0][0] = matrix[i][0]; //De acordo com a comparação dos valores, o sistema busca o número do cadastro que é correspondente

}

//O índice [0] se compara com o primeiro e fica estável, só recebendo os outros valores para comparacao. No final, ele recebe o maior valor.

maior = matrix[0][0]; //Substituindo o valor do kw pela sua referência "Cadastro".

}

printf("\n\* MAIOR consumo de Kw.\n Cliente número:\n %.0f com %.0fKW\n", maior, matrix[0][1]);

}

//FUNÇÂO: Quem gastou MENOS:

void menos\_gastou(float matrix[clientes][3]){

float menor;

for (int i=1; i<clientes; i++){

if (matrix[0][1] > matrix[i][1]){

matrix[0][1] = matrix[i][1]; //Se o valor do índice[0] for maior que o valor do índie[1], índice[0] recebe o valor de índie[1]. Caso contrário, índice[0] continua com seu valor.

matrix[0][0] = matrix[i][0]; //De acordo com a comparação dos valores, o sistema busca o número do cadastro que é correspondente

}

//O índice [0] se compara com o primeiro e fica estável, só recebendo os outros valores para comparacao. No final, ele recebe o maior valor.

menor = matrix[0][0]; //Substituindo o valor do kw pela sua referência "Cadastro".

}

printf("\n\* MENOR consumo de Kw.\n Cliente número:\n %.0f com %.0fKW\n", menor, matrix[0][1]);

}

//FUNÇÂO: Consumo MEDIO:

int consumo\_medio(float matrix[clientes][3]){

//se colocar VOID o resultado dá errado.

int media;

for (int i=0; i<clientes; i++){

media += matrix[i][1]; //Soma todos os valores da coluna da matriz.

}

printf("\n\* MÉDIA de consumo da cidade é:\n %.0d Kw \n", media/clientes); //O resultado da soma é dividido pelo número de clientes no cadastro.

}

//FUNÇÂO: Quantos ACIMA de 170KW:

void acima\_170(float matrix[clientes][3]){

int amais=0; //Tava dando errado porque nao tinha dado valor de 0 para esta variavel

for (int i=0; i<clientes; i++){

if (matrix[i][1] > 170){

amais++; //O looping vai ver quantos valores estão acima de 170 e caso verdadeiro, adiciona 1 para a variavel contadora.

} else{

amais = amais; //Caso falso, o valor da contadora permanece.

}

}

printf("\n\* Número de clientes ACIMA de 170Kw:\n %d\n", amais);

}

//FUNÇÂO: Cadastrar NOVO CLIENTE:

void cadastro(float matrix[clientes][3]){

int add= clientes +1; //Almentar o índice de linhas da matrix

printf("Número de cadastro: \n"); //O usuário coloca o número do cadastro

printf(">> "); //Indicar ao usuário uma espera da resposta.

scanf("%f", &matrix[add][0]);

while ((c= getchar())!= '\n' && c!=EOF){};//Limpeza de buffer

printf("Quantidade de Kw consumidos: \n"); //O usuário coloca a quantidade de KW.

printf(">> "); //Indicar ao usuário uma espera da resposta.

scanf("%f", &matrix[add][1]);

while ((c= getchar())!= '\n' && c!=EOF){};//Limpeza de buffer

matrix[add][2] = calculo(matrix[add][1]); //A função calculo calcula o valor a ser pago.

printf("\n- NOVO CLIENTE CADASTRADO COM SUCESSO! \n");

printf(" \*Cadastro no.: %.0f \n \*Kw: %.0f \n \*Total a pagar: %.2f\n\n", matrix[add][0], matrix[add][1], matrix[add][2]); //Impressão das informacões dadas: Cadastro, KW e valor a pagar.

}

//FUNÇÂO: MENU.

void menu(float matrix[clientes][3]){

int opcao;

printf("\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

printf("\*\*\*\* MENU \*\*\*\*\n");

printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

printf("Escolha uma das opções: \n\n");

printf("1 - Consumidor com MAIOR gasto \n");

printf("2 - Consumidor com MENOR gasto \n");

printf("3 - Quantidade de consumidores ACIMA de 170Kw \n");

printf("4 - Consumo MÉDIO da cidade (em Kw) \n");

printf("5 - Cadastro de NOVO CLIENTE \n");

printf(">> "); //Indicar ao usuário uma espera da resposta.

scanf("%d", &opcao);

while ((c= getchar())!= '\n' && c!=EOF){};//Limpeza de buffer

if ((opcao >=1)&&(opcao <=5)){

switch (opcao){

case 1:

mais\_gastou(matrix); //Chama a função de quem gastou mais.

break;

case 2:

menos\_gastou(matrix); //Chama a função de quem gastou menos.

break;

case 3:

acima\_170(matrix); //Chama a função de quantos gastaram acima de 170.

break;

case 4:

consumo\_medio(matrix); //Chama a função do consumo médio da cidade.

break;

case 5:

cadastro(matrix); //Chama a função para cadastro de cliente.

break;

default:

break;

}

} else {

printf("Por favor, escolha uma opção válida."); //Se escolher opcao inválida

}

char voltar;

printf("\nDeseja voltar ao Menu? s/n\n"); //Opcao de voltar ao MENU

printf(">> ");//Indicar ao usuário uma espera da resposta.

scanf("%c", &voltar);

while ((c= getchar())!= '\n' && c!=EOF){};//Limpeza de buffer

if (voltar== 's'){ //Opcao de voltar ao MENU é SIM chama a função menu().

menu(matrix);

}

else{

printf("\n---Agradecemos seu contato.---\n\n"); //Opcao de voltar ao MENU é NAO imprime um agradecimento na tela.

}

}

// \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// \*\*\*\* MAIN \*\*\*\*

// \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

int main() {

int numConta, totalKm, valPagar; //Variáveis da função

// Clientes já cadastrados:

float regis[clientes][3] = {

{1234, 110, calculo(regis[0][1])},

{2345, 200, calculo(regis[1][1])},

{3456, 34, calculo(regis[2][1])},

{4567, 77, calculo(regis[3][1])},

{5678, 89, calculo(regis[4][1])}

};

imprimir(regis); //Chama a função de imprimir a tabela dos clientes já cadastrados.

printf("\n");

menu(regis); //Chama a função MENU para dar início ao atendimento.

system ("pause");

return 0;

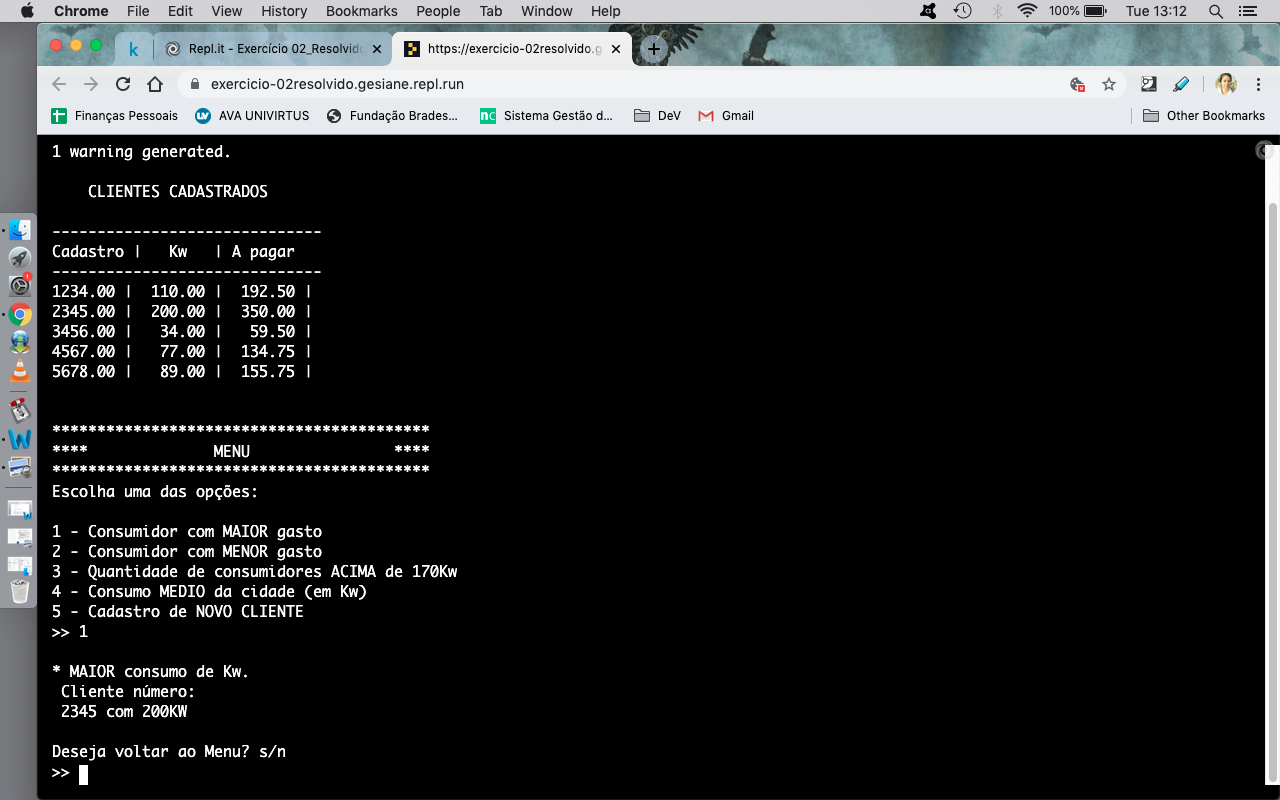
}

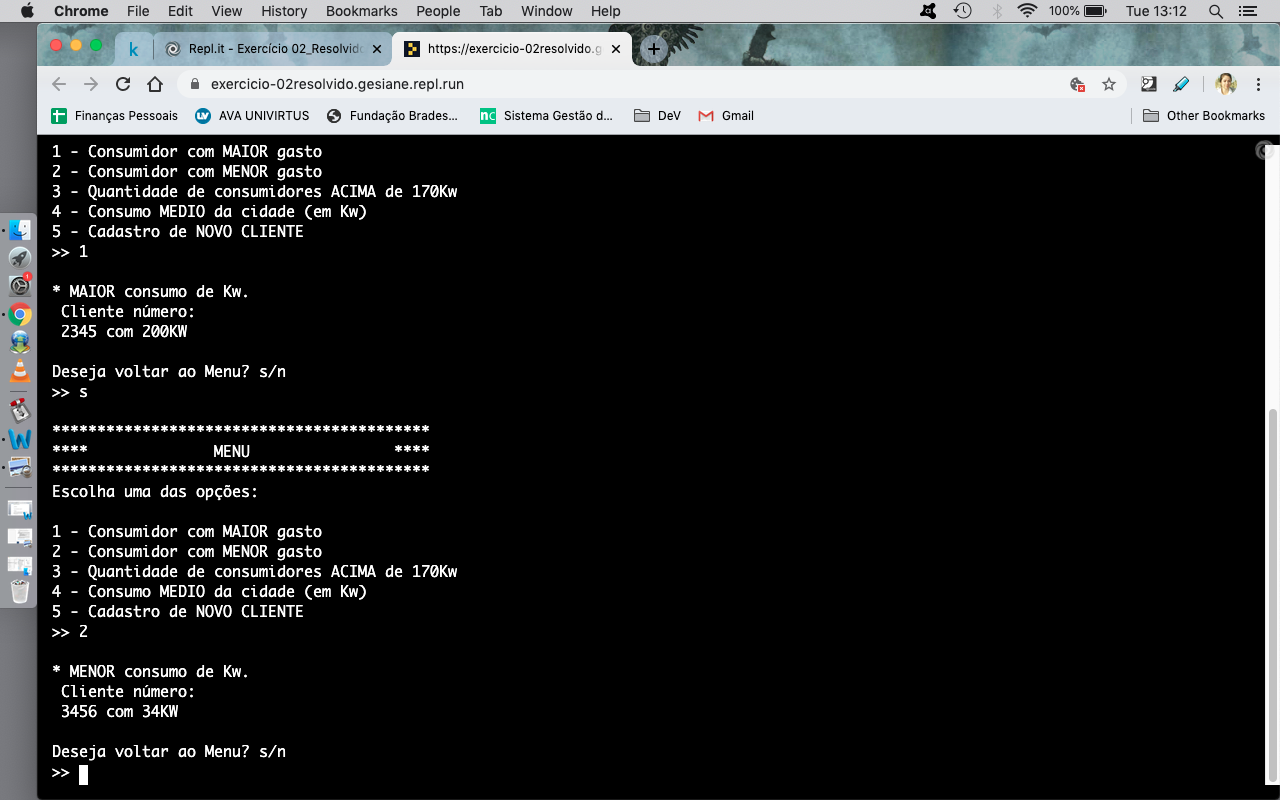
//\*\* REFERÊNCIAS

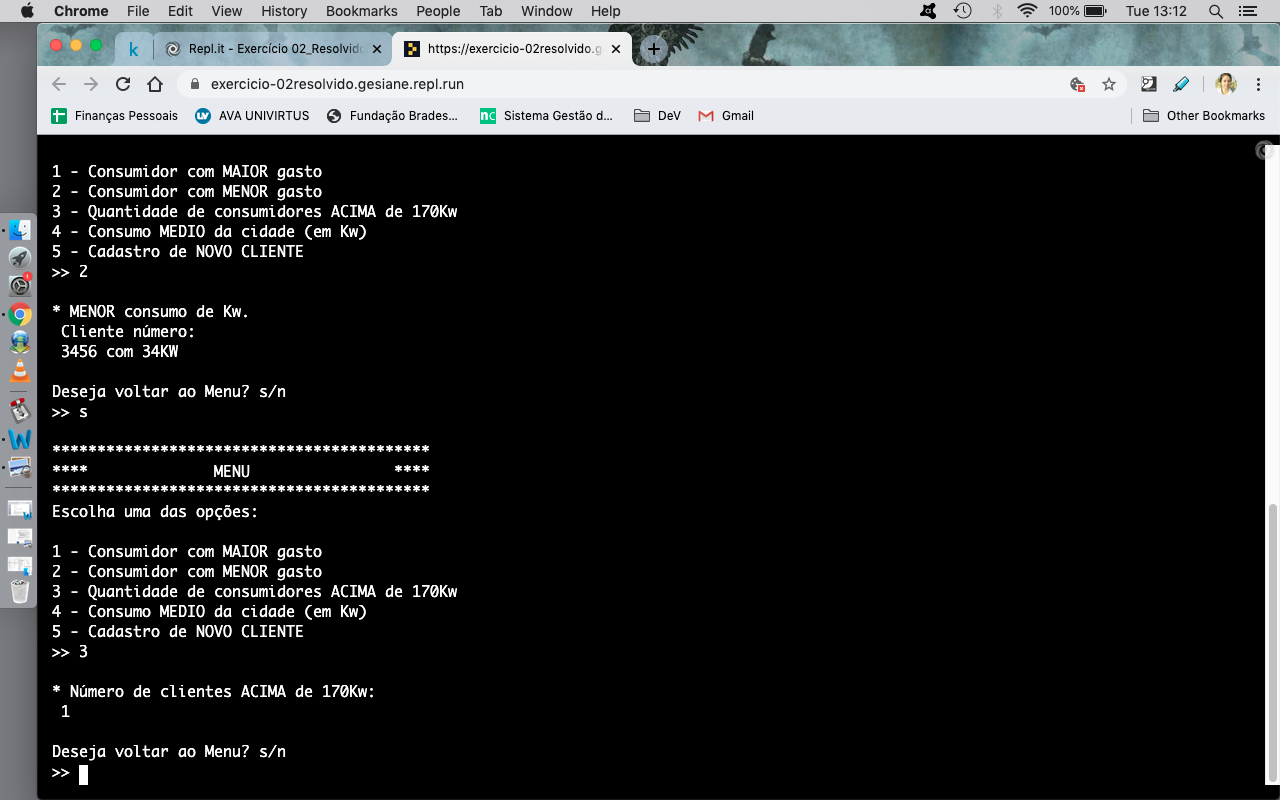
// sobre funcoes com matrizes: <https://www.programiz.com/c-programming/c-arrays-functions>

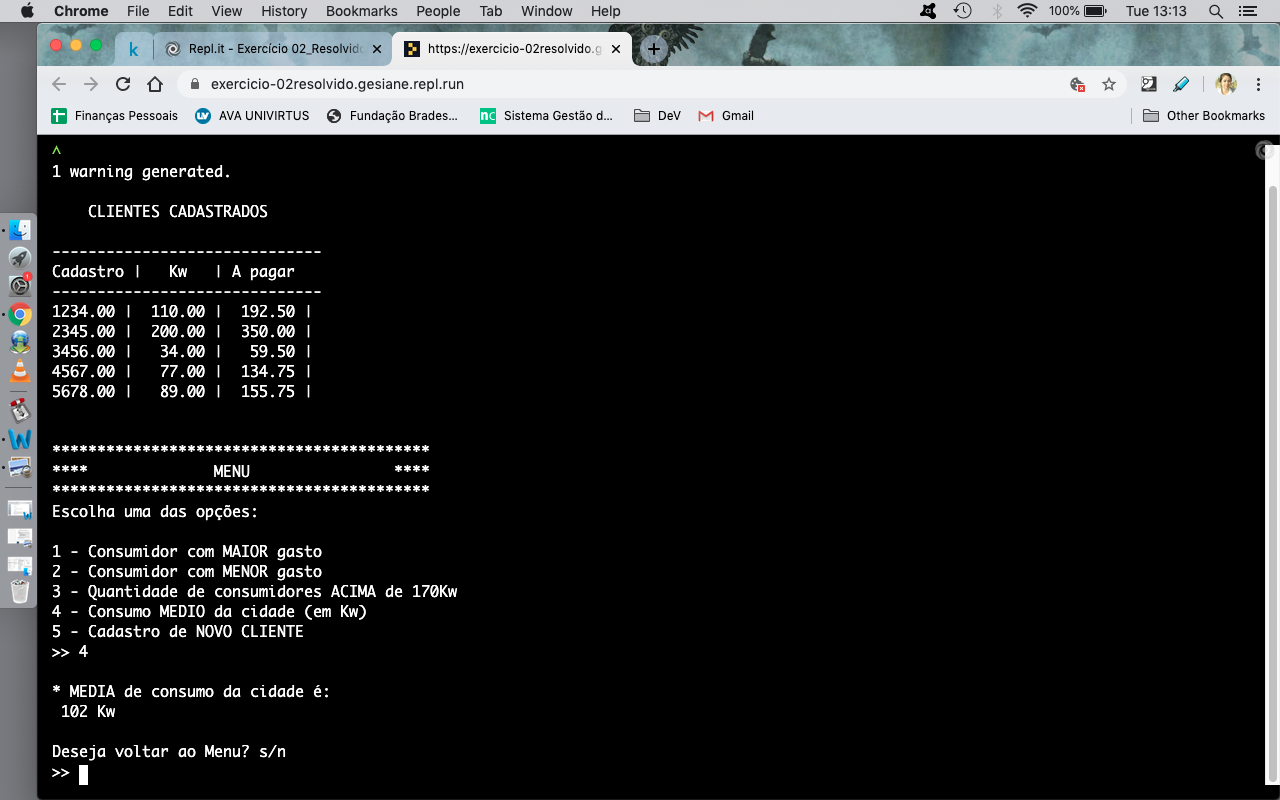
// sobre looping de comparações: <https://www.programiz.com/c-programming/examples/array-largest-element>

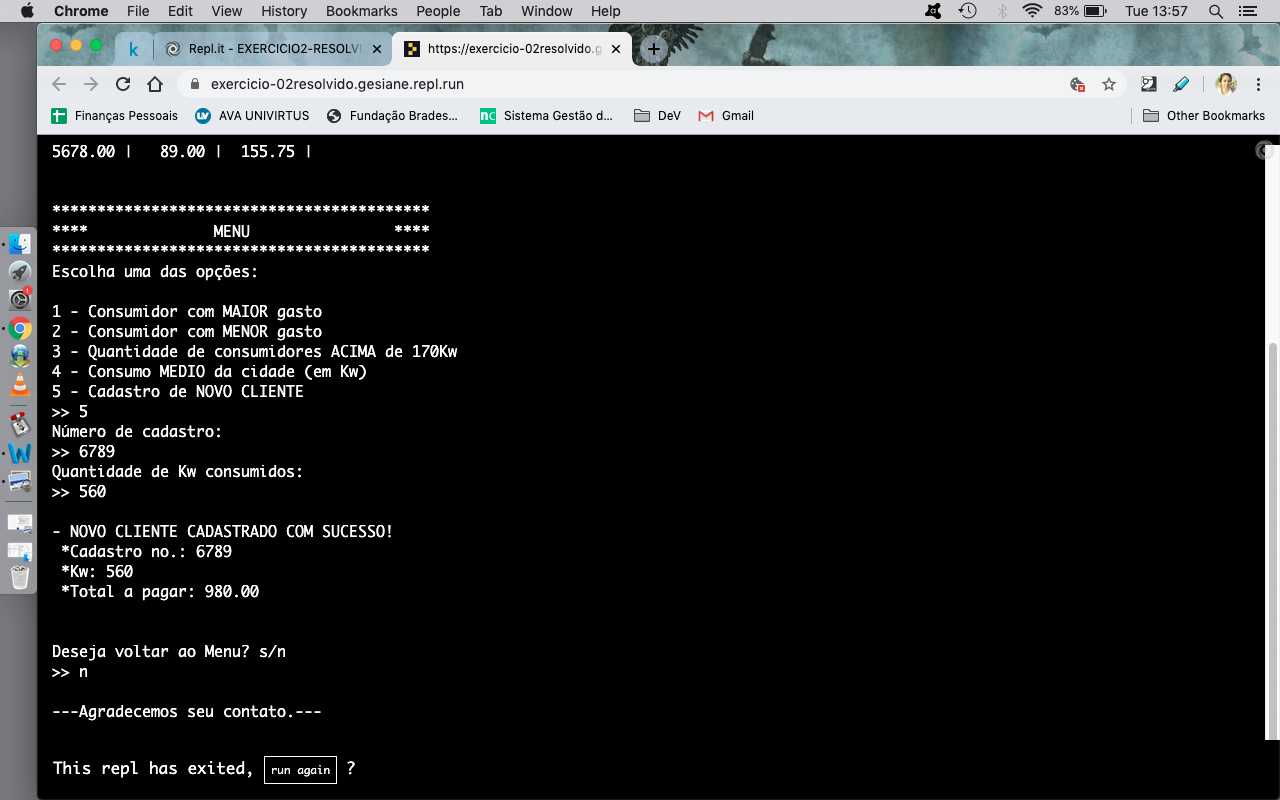
Imagem do código funcionando no seu computador:

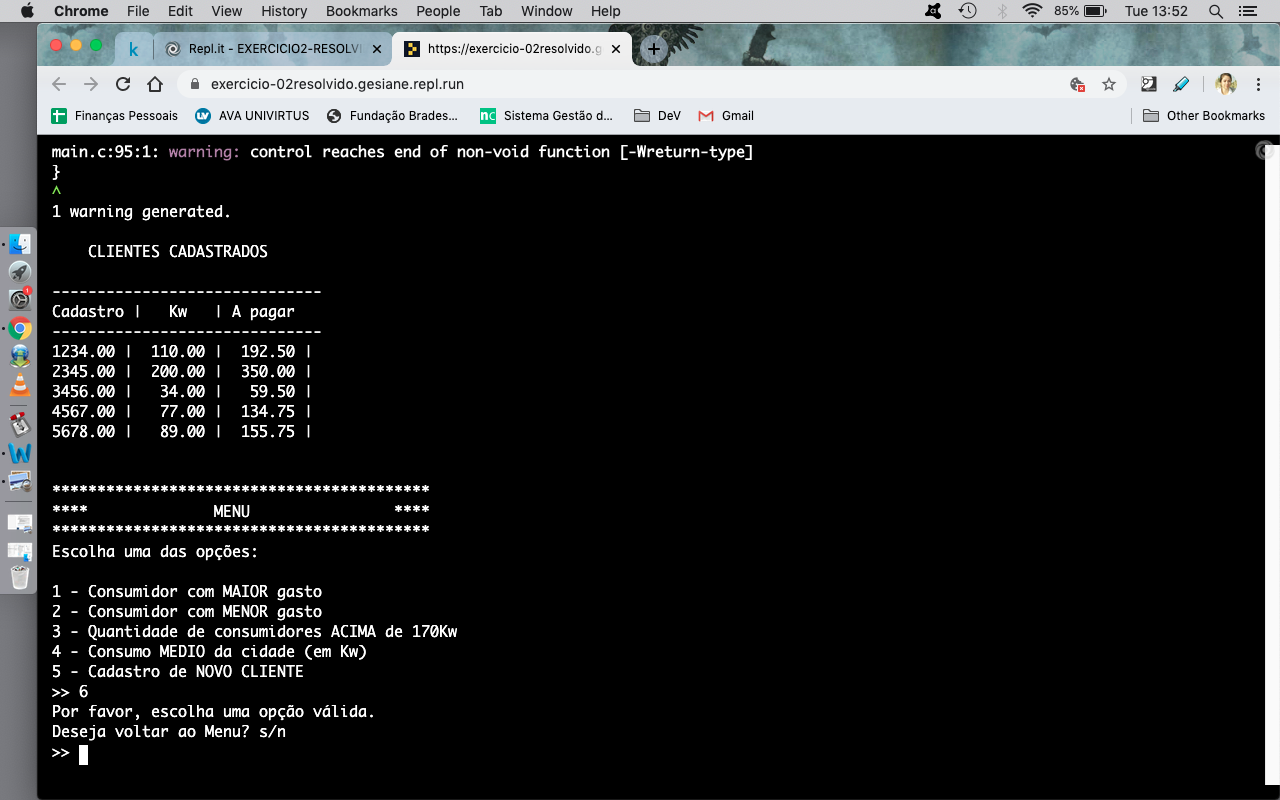












# EXERCÍCIO 3

ENUNCIADO: Dizemos que uma matriz quadrada inteira é um quadrado mágico se a soma dos elementos de cada linha, a soma dos elementos de cada coluna e a soma dos elementos das diagonais principal e secundária são todas iguais. Dada uma matriz quadrada dimensão MxM, verifique se ela é um quadrado mágico através de um algoritmo.

Exemplo de matriz quadrado mágico:

[ 8 0 7 ]

[ 4 5 6 ]

[ 3 10 2 ]

Solução do aluno:

#include <stdio.h>

#include<stdlib.h>

int main() {

int tam, i, j, lincol, somadiago, x;

int linha[i], coluna[i], dia1, dia2;

//Pedido para o usuário criar a matriz

printf("Qual o tamanho da sua matriz?\n>> ");//Definindo o tamanho da matriz.

scanf("%d", &tam);

int matriz[tam][tam];

for (i=0; i<tam; i++){ //Looping para montar a matriz

for (j=0; j<tam; j++){

printf("Digite um número para a posição [%d][%d]: \n>> ", i, j);

scanf("%d", &matriz[i][j]);

}

}

//Imprimindo a matriz

printf("\nA sua matriz ficou assim: \n"); //Looping para imprimir a matriz

for (i=0; i<tam; i++){

for (j=0; j<tam; j++){

printf("%d ",matriz[i][j]);

}

printf("\n");

}

//Somando as Linhas (e as Colunas)

for (j=0; j<tam; j++){

for (i=0; i<tam; i++){

linha[i] += matriz[i][j]; //Looping para somar os valores das linhas

coluna[i] += matriz[j][i]; //e colunas.

}

}

//Comparando a soma das linhas e colunas

for (i=0; i<tam; i++){

//Nao consegui usar boolean

if (linha[i] == coluna[i]){ //Se a soma da linha for igual a soma da coluna

lincol = 1; //A variavel lincol recebe 1 caso contrário, recebe 0.

} else{

lincol = 0;

}

}

//Teste:

//printf("diagonal1 = %d \n", dia1);

// Esse Não funciona por nada no mundo:

//Tentei um lopping pra somar os valores da diagonal secundária.

// for (i=(tam-1); i<0; i--){

// for (j=0; j<tam; j++){

// int x = i + j;

// if (x == (tam-1)){

// dia2 += matriz[i][j];

// }

// }

// }

//Esse funcionou mas dá sempre uma unidade à mais que a soma correta:

for (i=0; i<tam; i++){

dia1 += matriz[i][i]; //Soma da diagonal principal

for (j=0; j<tam;j++){

if (j==(tam-1)-i){

dia2 += matriz[i][j]; //Soma da diagona secundária

}

}

}

//Como resulta sempre em uma unidade à mais na soma, por isso:

dia2 = dia2-1;

//Teste:

//printf("\n diagona2 = %d \n", dia2);

//Comparando as somas das diagonais principal e secunária

if (dia1 == dia2){ //Se a soma de cada diagonal for igual, somadiago recebe 1, caso contrário, recebe 0.

somadiago = 1;

} else{

somadiago = 0;

}

//É mágico ou nao:

if (lincol == 1 && somadiago == 1){ //Se na comparacao de linhas e colunas o lincol recebeu valor igual a 1, elas são de mesmo valor. E se somadiago também recebeu 1, o quadro é mágico porque todas as somas, de linhas, colunas e diagonais são iguais.

printf("\n \*\*\*\*\*\*\*\*\*\n");

printf("O seu Quadrado É MÁGICO!!\n");

printf(" \*\*\*\*\*\*\*\*\*\n");

} else{ //Se não, se lincol e/ou somadiago recebeu valor 0, é porque as somas não são iguais e o quadro não é mágico.

printf("\n \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

printf("O seu Quadrado NÃO é Mágico. :( \n");

printf(" \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

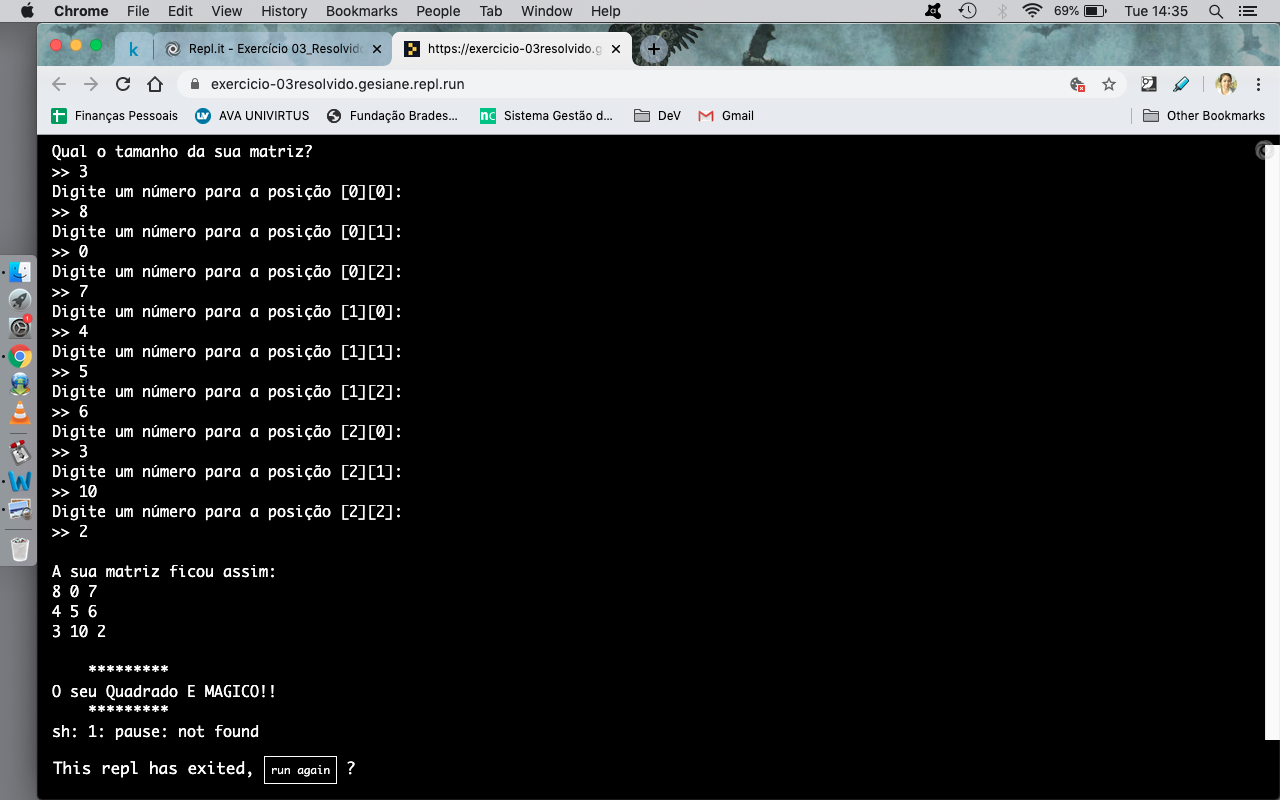
}

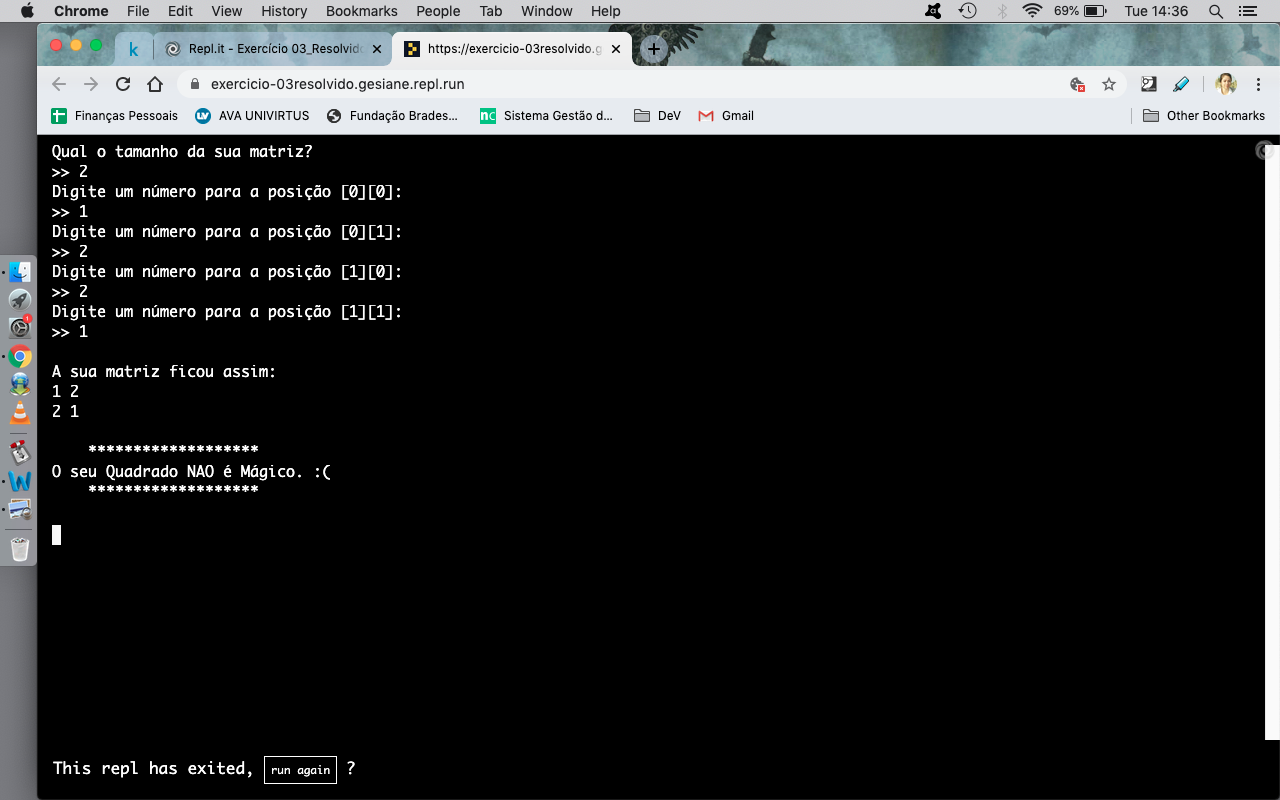
system("pause");

return 0;

}

Imagem do código funcionando no seu computador:





# EXERCÍCIO 4

ENUNCIADO: Escrever uma função que receba como parâmetro duas *strings*, bem como um valor inteiro que representa uma posição. A função deve, portanto, inserir a segunda *string* no na posição indicada da primeira.

Escreva um programa que receba estas duas *strings* do usuário, o valor da posição desejada, e chame a função anteriormente implementada e exiba o resultado ao usuário na tela.

Para imprimir na tela e tirar o *print screen*, mostre o resultado utilizando o seu primeiro nome como *string* 1 e o seu último nome como *string* 2, e a posição de teste deverá ser o último digito do seu RU.

Solução do aluno:

#include <stdio.h>

#include <stdlib.h>

#include <string.h> //<--Biblioteca pra trabalhar com strings

/\*

Para esse trabalho vão ser usadas as funcoes da biblioteca STRING.H:

- strncpy -: copies a specified bytes of characters from source to destination.

- strcat -: The function strcat() concatenates two strings. It takes two arguments, i.e, two strings or character arrays, and stores the resultant concatenated string in the first string specified in the argument.

- strcpy -: The strcpy() function copies the string to the another character array.

REFERENCIA

<https://www.programiz.com/cpp-programming/library-function/cstring>

\*/

//Declarando a função a ser usada.

//void juntando(char primeira[], char segunda[], int num);

//(!) MAS, se a funçao for escrita antes do MAIN, nao tem necessidade de declarar ela E depois escreve-la (depois do MAIN):

void juntando(char primeira[], char segunda[], int num){

char resultado[50];

strncpy(resultado, primeira, num);

// Corta a primeira palavra até o número que foi indicado e guarda em ´resultado´:

// Exe.: primeira = Gesiane, num= 4

// resultado = Gesi

/\*----------TESTE------------\*/

//printf("strncpy= %s \n", resultado);

strcat(resultado, segunda);

//Contatena o resultado anterior com a segunda palavra:

//Ex.: segunda = Pajarinen

//resultado = GesiPajarinen

/\*----------TESTE------------\*/

//printf("strcat - segunda = %s \n", resultado);

strcat(resultado,&primeira[num]);

//Concatena o resultado aterior com o resto da primeira palavra:

//&primeira[num] = ane

//resultado = GesiPajarinenane

/\*----------TESTE------------\*/

//printf("&primeira[num] = %s \n", &primeira[num]);

/\*----------TESTE------------\*/

//printf("strcat - &primeira[num] = %s \n", resultado);

strcpy(primeira, resultado);

//Copia o valor de 'resultado' para o valor de 'primeira', que será impresso como resultado final no MAIN. (Tentei colocar [primeira = resultado] mas nao dá certo.)

/\*----------TESTE------------\*/

//printf("strcpy - primeira + resultado = %s \n", resultado);

}

int main( ) {

char primeira[50], segunda[50];

int num;

//Pedindo a entrada dos dados:

printf("Digite uma palavra: \n>> "); //Priemira palavra

scanf("%s", primeira);

printf("Digite outra palavra: \n>> "); //Segunda palavra

scanf("%s", segunda);

printf("Digite um número: \n>> "); //Número

scanf("%d", &num);

//Mostrando os dados:

printf("\nColocando a segunda palavra '%s'", segunda);

printf(" no lugar '%d'", num);

printf(" da primeira palavra '%s', teremos: \n", primeira);

//Chamando a funcao que vai trabalhar com as strings:

juntando( primeira, segunda, num);

//Imprimir o resultado:

printf("------------------------\n");

printf(" %s \n", primeira);

printf("------------------------\n");

system ("pause");

return 0;

}

//Agora coloco a funcao (Vou deixar a funcao no inicio, porque é mais prático)

/\*

void juntando(char primeira[], char segunda[], int num){

char resultado[50];

strncpy(resultado, primeira, num);

strcat(resultado, segunda);

strcat(resultado,&primeira[num]);

strcpy(primeira, resultado);

} \*/

Imagem do código funcionando no seu computador:

