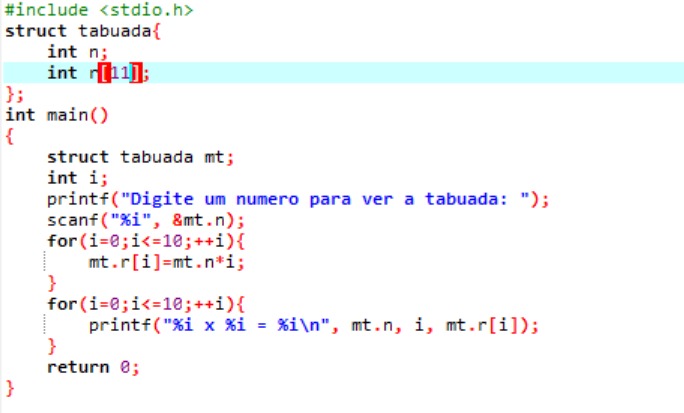
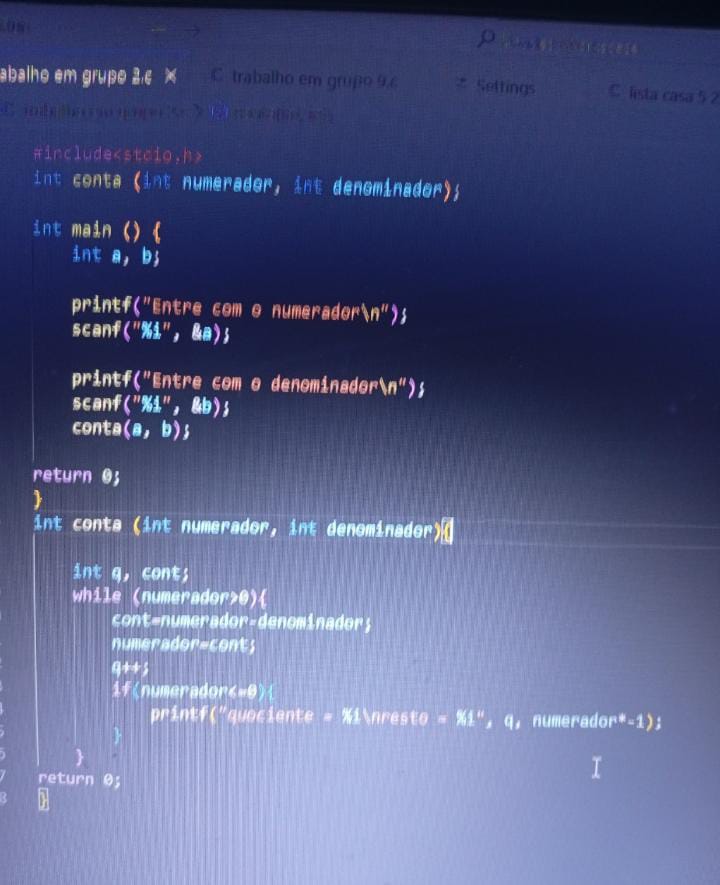
Trabalho em grupo

Feito por: **Arthur Fukuyama de Andrade, Giovanni Carro, Heitor Akira Issatugo, Isabela Gomes de Oliveira, Leonardo Cabrera Correa, Thiago Daiki Sato.**

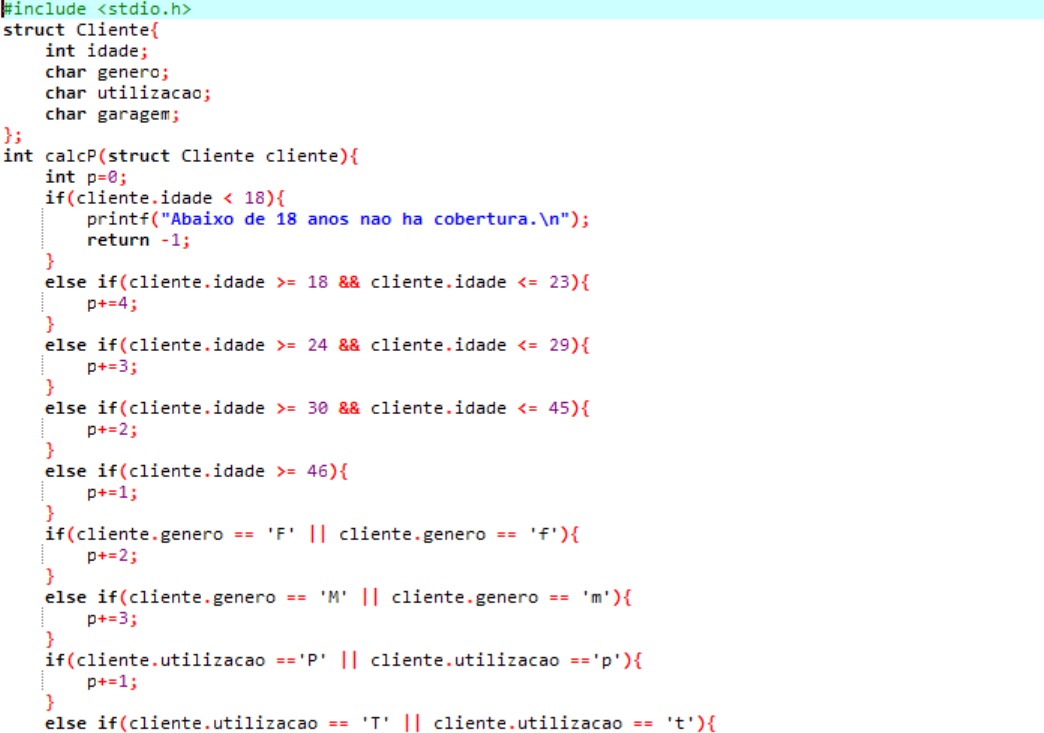
EXERCICIO 1

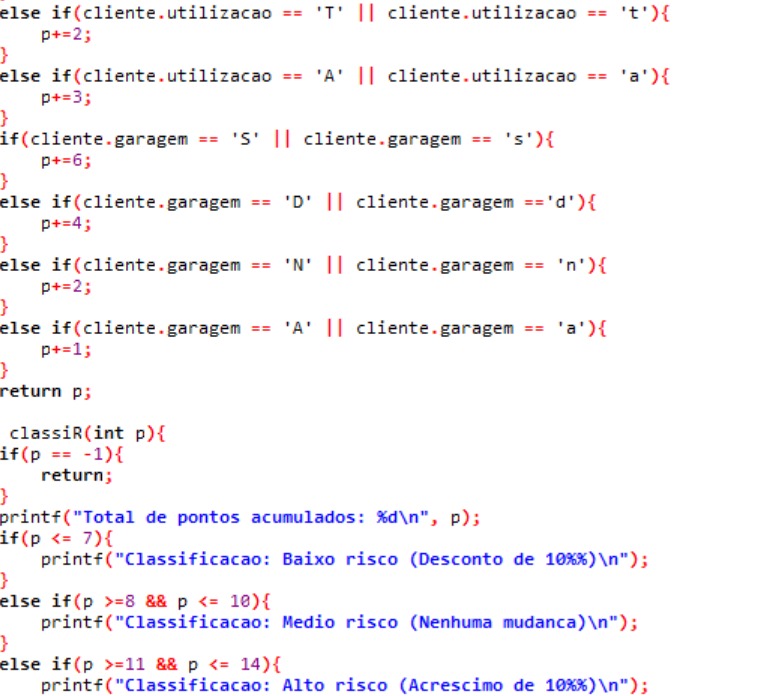


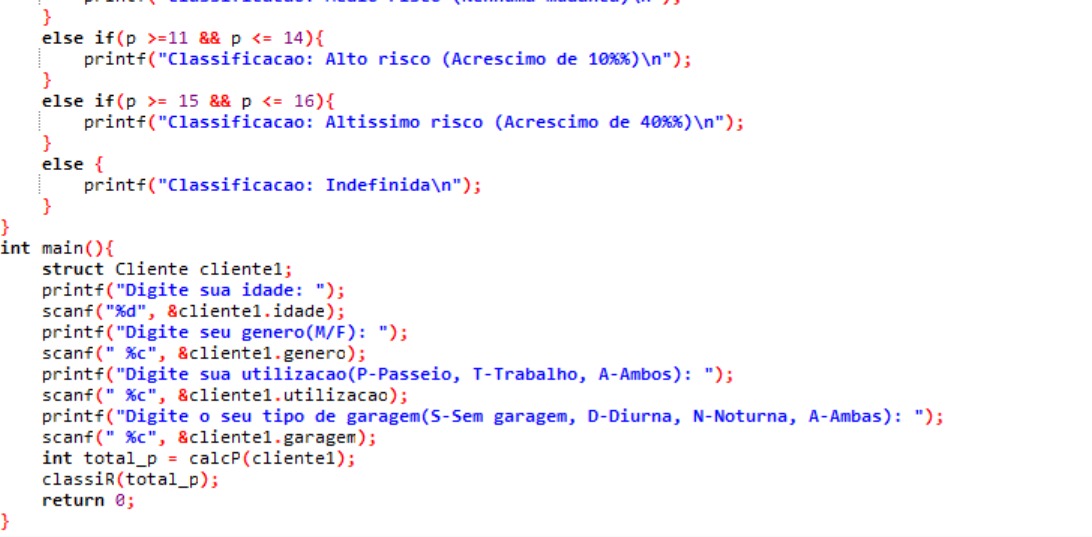
EXERCICIO 2



EXE 3







EXE 4

#include<stdio.h>

#include<string.h>

struct pessoa{

char nome[100];

} aluno[40];

int i, j;

char x[100];

void nome()

{

for(i=0;i<40;i++){

scanf("%s", &aluno[i].nome);

}

}

void crescente()

{

for(i=0;i<39;i++){

for(j=i+1;j<40;j++){

if((strcmp(aluno[i].nome, aluno[j].nome))>0){

strcpy(x,aluno[i].nome);

strcpy(aluno[i].nome,aluno[j].nome);

strcpy(aluno[j].nome,x);

}

}

}

printf("ordem crescente dos nomes:\n");

for(i=0;i<40;i++){

printf("%s\n", aluno[i].nome);

}

}

void pesquisa()

{

printf("entre com o nome que deseja pesquisar:\n");

char pesquisa[50];

scanf("%s", &pesquisa);

for(i=0;i<40;i++){

if(strcmp(pesquisa, aluno[i].nome)==0){

break;

}

}

printf("\no nome %s esta na posicao %d", aluno[i].nome, i+1);

}

int main()

{

printf("entre com o nome de 40 alunos:");

nome();

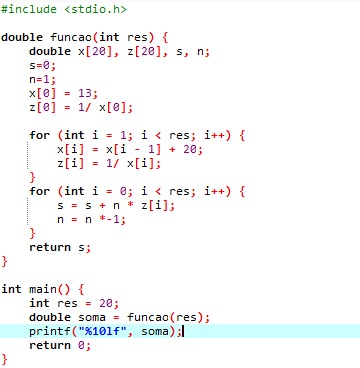
crescente();

pesquisa();

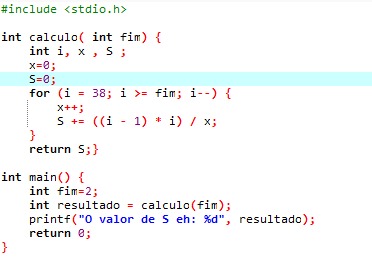
return 0;

}

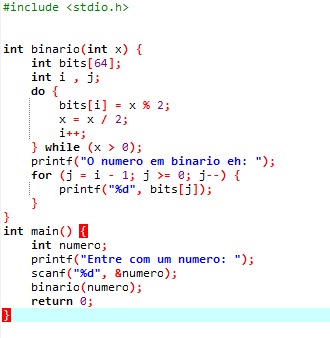
EXE 5- A



EXE 5- B



EXE 6



EXE 7

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

#define MAX\_PESSOAS 5

typedef struct { char nome[50]; char endereco[100]; char email[50]; char telefone[15];

} Pessoa; void cadastrar(Pessoa agenda[], int \*contador); void pesquisar(Pessoa agenda[], int contador); void classificar(Pessoa agenda[], int contador); void alterar(Pessoa agenda[], int contador); void remover(Pessoa agenda[], int \*contador);

int main() { Pessoa agenda[MAX\_PESSOAS]; int contador = 0; int opcao; do { printf("\n--- MENU ---\n"); printf("1. Cadastrar pessoa\n"); printf("2. Pesquisar pessoa\n"); printf("3. Classificar por nome\n"); printf("4. Alterar cadastro\n"); printf("5. Remover cadastro\n"); printf("6. Sair\n"); printf("Escolha uma opção: "); scanf("%d", &opcao); getchar();

switch (opcao) { case 1: cadastrar(agenda, &contador); break; case 2: pesquisar(agenda, contador); break; case 3: classificar(agenda, contador); break; case 4: alterar(agenda, contador); break; case 5: remover(agenda, &contador); break; case 6: printf("Saindo...\n"); break; default: printf("Opção inválida! Tente novamente.\n"); } } while (opcao != 6); return 0; }

void cadastrar(Pessoa agenda[], int \*contador) {

if (\*contador < MAX\_PESSOAS) {  
  
 printf("\n--- Cadastro de Pessoa ---\n");  
  
 printf("Nome: ");

fgets(agenda[\*contador].nome, sizeof(agenda[\*contador].nom), stdin); agenda[\*contador].nome[strcspn(agenda[\*contador].nome, "\n")] = 0; printf("Endereço: "); fgets(agenda[\*contador].endereco, sizeof(agenda[\*contador].endereco), stdin);

agenda[\*contador].endereco[strcspn(agenda[\*contador].endereco, "\n")] = 0;  
  
 printf("E-mail: ");  
  
 fgets(agenda[\*contador].email, sizeof(agenda[\*contador].email), stdin);  
  
 agenda[\*contador].email[strcspn(agenda[\*contador].email, "\n")] = 0;  
 printf("Telefone: ");  
 fgets(agenda[\*contador].telefone, sizeof(agenda[\*contador].telefone), stdin); agenda[\*contador].telefone[strcspn(agenda[\*contador].telefone, "\n")] = 0;  
 (\*contador)++;  
} else {  
 printf("Agenda cheia!\n");  
}

} void pesquisar(Pessoa agenda], int contador) char nomePesquisado[50]; int encontrado = 0; printf("\nDigite o nome para pesquisa: "); fgets(nomePesquisado, sizeof(nomePesquisado), stdin); nomePesquisado[strcspn(nomePesquisado, "\n")] = 0;

for (int i = 0; i < contador; i++) { if (strcmp(agenda[i].nome, nomePesquisado) == 0) { printf("\n--- Dados encontrados ---\n"); printf("Nome: %s\n", agenda[i].nome); printf("Endereço: %s\n", agenda[i].endereco); printf("E-mail: %s\n", agenda[i].email); printf("Telefone: %s\n", agenda[i].telefone); encontrado = 1; break; } } if (!encontrado) { printf("Pessoa não encontrada!\n"); } } void classificar(Pessoa agenda[], int contador) { Pessoa temp; for (int i = 0; i < contador - 1; i++) { for (int j = i + 1; j < contador; j++) { if (strcmp(agenda[i].nome, agenda[j].nome) > 0) { temp = agenda[i]; agenda[i] = agenda[j]; agenda[j] = temp; } } } printf("\nAgenda classificada por nome!\n"); } void alterar(Pessoa agenda[], int contador) { char nomeAlterado[50]; int encontrado = 0; printf("\nDigite o nome para alterar: "); fgets(nomeAlterado, sizeof(nomeAlterado), stdin); nomeAlterado[strcspn(nomeAlterado, "\n")] = 0; for (int i = 0; i < contador; i++) { if (strcmp(agenda[i].nome, nomeAlterado) == 0) { printf("Digite as novas informações:\n");

printf("Nome: ");  
 fgets(agenda[i].nome, sizeof(agenda[i].nome), stdin); agenda[i].nome[strcspn(agenda[i].nome, "\n")] = 0;  
 printf("Endereço: ");  
 fgets(agenda[i].endereco, sizeof(agenda[i].endereco), stdin); agenda[i].endereco[strcspn(agenda[i].endereco, "\n")] = 0;  
 printf("E-mail: ");  
 fgets(agenda[i].email, sizeof(agenda[i].email), stdin);  
 agenda[i].email[strcspn(agenda[i].email, "\n")] = 0;  
 printf("Telefone: ");  
 fgets(agenda[i].telefone, sizeof(agenda[i].telefone), stdin); agenda[i].telefone[strcspn(agenda[i].telefone, "\n")] = 0;  
 encontrado = 1;  
 break;  
 }  
}  
if (!encontrado) {  
 printf("Pessoa não encontrada!\n");

} } void remover(Pessoa agenda[], int \*contador) {

char nomeRemover[50];  
int encontrado = 0;  
printf("\nDigite o nome para remover: ");  
fgets(nomeRemover, sizeof(nomeRemover), stdin); nomeRemover[strcspn(nomeRemover, "\n")] = 0;  
for (int i = 0; i < \*contador; i++) {  
 if (strcmp(agenda[i].nome, nomeRemover) == 0) {  
 for (int j = i; j < \*contador - 1; j++) {  
 agenda[j] = agenda[j + 1];   
 }  
 (\*contador)--;  
 encontrado = 1;  
 break;  
 }  
}  
if (!encontrado) {  
 printf("Pessoa não encontrada!\n");  
}

}

EXE 8

#include <stdio.h>

#include <math.h>

// Estrutura

struct consumo {

float consumom;

float consumojn;

float consumojl;

float consumoa;

};

typedef struct consumo consumo;

consumo con[5]; // Global

// Variáveis globais

float meta;

int i;

int dentro = 0, fora = 0;

// Função para calcular a meta

void calculaMeta() {

float media = (con[i].consumom + con[i].consumojn + con[i].consumojl) / 3.0;

meta = media \* 0.8;

}

// Função para ler os dados de uma casa

void lerDados() {

printf("Casa %i: \n", i + 1);

printf("Insira a media de consumo de Maio de 2006: ");

scanf("%f", &con[i].consumom);

printf("Insira a media de consumo de Junho de 2006: ");

scanf("%f", &con[i].consumojn);

printf("Insira a media de consumo de Julho de 2006: ");

scanf("%f", &con[i].consumojl);

printf("Insira o consumo do mes atual: ");

scanf("%f", &con[i].consumoa);

}

// Função para verificar se está dentro da meta

void verificarMeta() {

calculaMeta();

if (con[i].consumoa < meta) {

printf("Voce esta dentro da meta.\n\n");

dentro++;

} else {

printf("Voce esta fora da meta.\n\n");

fora++;

}

}

// Função para mostrar o resumo final

void mostrarResumo() {

int total = dentro + fora;

if (total > 0) {

float porc = (dentro \* 100.0) / total;

printf("\nResumo:\n");

printf("Casas dentro da meta: %d\n", dentro);

printf("Casas fora da meta: %d\n", fora);

printf("Porcentagem de casas dentro da meta: %.2f%%\n", porc);

}

}

// Função principal

int main() {

char escolha;

for (i = 0; i < 5; ++i) {

lerDados();

verificarMeta();

printf("Deseja continuar? (s/n)\n");

scanf(" %c", &escolha);

if (escolha == 'n' || escolha == 'N') {

break;

}

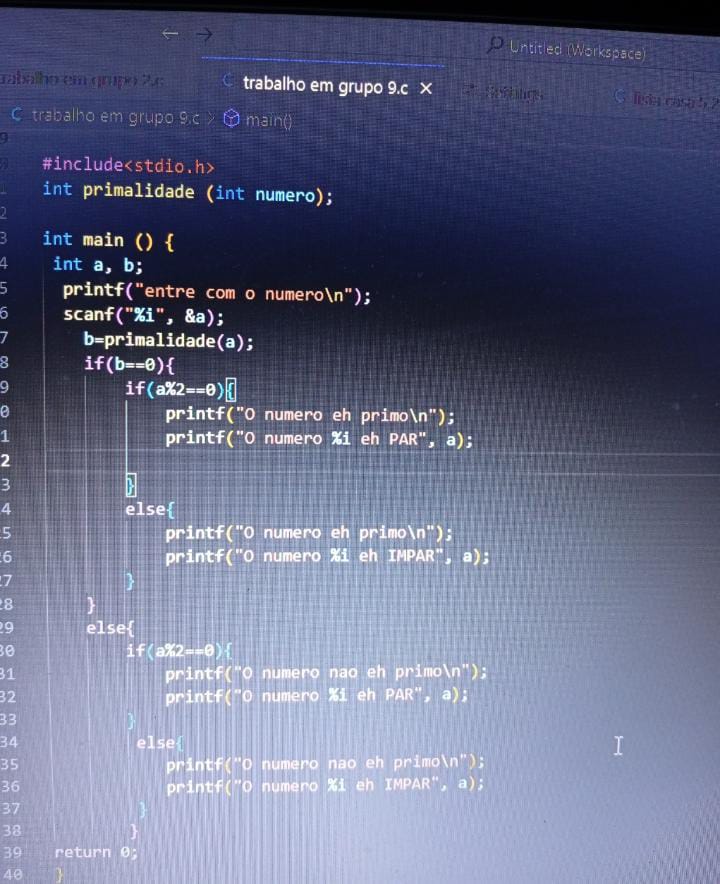
}

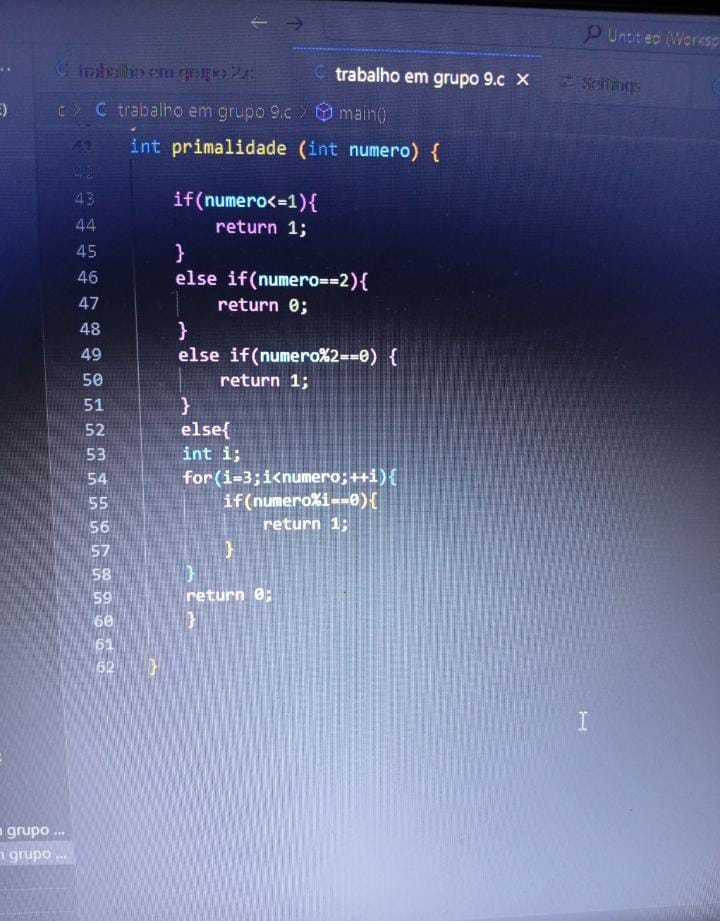
mostrarResumo();

return 0;

}

EXE 9





EXE 10

#include<stdio.h>

int n50(int d, int \*p) { int resultado; resultado = d/50; \*p= d % 50; return(resultado); } int n10(int \*p) { int resultado; resultado = \*p /10; \*p = \*p %10; return(resultado); } int n5(int \*p) { int resultado; resultado = \*p /5; \*p = \*p %5; return(resultado); } int n1(int \*p) { int resultado; resultado = \*p /1; \*p = \*p %1; return(resultado); } int main() { int d, resto; int A1, A2, A3, A4; printf("entre com o valor que deseja sacar:\n"); scanf("%d", &d); A1 =n50(d, &resto); A2 =n10(&resto); A3 =n5(&resto); A4 =n1(&resto); printf("%d %d %d %d", A1, A2, A3, A4); return 0; }