

Questão 1 (2,0 pontos)

Faça um programa que peça para o usuário fornecer os valores inteiros de 2 matrizes M e N ,ambas com 3 linhas 4 colunas e que imprima na tela a matriz $O = 3 * M - 5 * N$

RESPOSTA

```
#include <stdio.h>
#define LINHA 3
#define COLUNA 4
void preencheMatriz(int matriz[LINHA][COLUNA], char nomeMatriz[10]) {
    for (int i = 0; i < LINHA; i++) {
        for (int j = 0; j < COLUNA; j++) {
            printf("Informe o numero para a posição [%d][%d] da matriz_%s \n", i + 1, j + 1, nomeMatriz);
            scanf("%d", &matriz[i][j]);
        }
    }
}
void imprimeMatriz (int matriz[LINHA][COLUNA], char nomeMatriz[10]) {
    int contadorLinha, contadorColuna;
    printf("-----MATRIZ %s----- \n", nomeMatriz);
    for (contadorLinha = 0; contadorLinha < LINHA; contadorLinha++) {
        for (contadorColuna = 0; contadorColuna < COLUNA; contadorColuna++) {
            if (contadorColuna < COLUNA - 1) {
                printf("[%d]-", matriz[contadorLinha][contadorColuna]);
            } else {
                printf("[%d]", matriz[contadorLinha][contadorColuna]);
            }
        }
        printf("\n");
    }
    printf("\n \n");
}
int main() {
    int matriz_M[LINHA][COLUNA];
    int matriz_N[LINHA][COLUNA];
    int matriz_O[LINHA][COLUNA];
    int contadorLinha, contadorColuna;
    preencheMatriz(matriz_M, "M");
    preencheMatriz(matriz_N, "N");
    for (contadorLinha = 0; contadorLinha < LINHA; contadorLinha++) {
        for (contadorColuna = 0; contadorColuna < COLUNA; contadorColuna++) {
            matriz_O[contadorLinha][contadorColuna] =
                (3 * matriz_M[contadorLinha][contadorColuna]) -
                (5 * matriz_N[contadorLinha][contadorColuna]);
        }
    }
    imprimeMatriz(matriz_M, "M");
    imprimeMatriz(matriz_N, "N");
```

```

imprimeMatriz(matriz_O, "O");
return 0;
}

```

Questão 2 (2,0 pontos)

Faça um programa declarando uma variável estática x do tipo float e uma variável dinâmica p também do tipo float. O programa deve pedir para o usuário fornecer um valor para a variável estática. Depois deve colocar o endereço de memória da variável dinâmica igual ao endereço de memória da variável estática e atribuir a variável estática o valor $\ast p / 3$. No final o programa deve imprimir na tela o endereço de p e o valor da variável estática x com 2 casas decimais.

RESPOSTA

```

#include <stdio.h>

int main () {
    float x;
    float *p;
    printf("Informe o valor da variável estática X \n");
    scanf("%f", &x);

    p = &x;

    x = *p / 3;

    printf("O endereço da variável dinâmica P é %p e o valor da variável estática X é %.2f", p, x);
    return 0;
}

```

Questão 3 (2,0 pontos) : Faça um programa fonte leia nome (Nome Completo) ,endereço,idade e telefone de 10 clientes guardando estes dados num vetor de struct sendo o nome do struct `struct lista_clientes` e o nome do vetor deste struct clientes que possa guardar até 100 clientes e depois mostre na tela o nome do sexto cliente, o endereço do nono cliente, a idade do quarto cliente e o telefone do terceiro cliente, sendo cada uma desta informações em uma linha diferente.

RESPOSTA

```

#include <stdio.h>

/* Função utilizada para limpa o buffer que o scanf carrega*/
void flush_in()
{
    int ch;
    while( (ch = fgetc(stdin)) != EOF && ch != '\n' ){ }
}

struct lista_clientes {
    char nome[50];
    char endereco[50];
    int idade;
}

```

```

char telefone[20];
};

int main() {
    struct lista_clientes clientes[100];

    for (int i = 0; i <= 9; i++) {
        printf("Informe o nome do cliente %d \n", i + 1);
        gets(clientes[i].nome);
        printf("Informe o endereço do cliente %d \n", i + 1);
        gets(clientes[i].endereco);
        printf("Informe a idade do cliente %d \n", i + 1);
        scanf("%d", &clientes[i].idade);
        flush_in();
        printf("Informe o telefone do cliente %d \n", i + 1);
        gets(clientes[i].telefone);
    }

    printf("Nome do SEXTO cliente, %s \n", clientes[5].nome);
    printf("Endereço do NONO cliente, %s \n", clientes[8].endereco);
    printf("Idade do QUARTO cliente, %d \n", clientes[3].idade);
    printf("Telefone do TERCEIRO cliente, %s \n", clientes[2].telefone);

    return 0;
}

```

Questão 4 (2,0 pontos) : Faça um programa fonte leia nome (Nome Completo),endereço,idade e telefone de 10 clientes e guarde estes dados em um arquivo denominado `clientes.txt` do tipo “w”, sendo que o nome,endereço,idade e telefone devem ficar em uma linha para cada cliente. Coloque o programa fonte como resposta

RESPOSTA

```

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#define CLIENTES 10

/* Função utilizada para limpa o buffer que o scanf carrega*/
void flush_in() {
    int ch;
    while ((ch = fgetc(stdin)) != EOF && ch != '\n') {
    }
}

/*Função para limpar o \n (enter) que o fgets coloca por padrão no final de cada entrada*/
char *lerStringSeguramente(char *string, int tamanho) {
    if (fgets(string, tamanho, stdin) != NULL) {
        /* Remove a nova linha (\n), caso ela tenha sido lida pelo fgets */
        int indiceUltimoCaractere = strlen(string) - 1;
        if (string[indiceUltimoCaractere] == '\n') {
            string[indiceUltimoCaractere] = '\0';
        }
    }
}

```

```

    return string;
}

return NULL;
}

struct lista_clientes {
    char nome[50];
    char endereco[50];
    int idade;
    char telefone[20];
};

int main() {
    struct lista_clientes clientes[100];
    int contadorCliente, contadorEscritaArquivo;

    /*Leitura de informações do cliente*/
    for (contadorCliente = 0; contadorCliente < CLIENTES; contadorCliente++) {
        printf("Informe o nome do cliente %d \n", contadorCliente + 1);
        lerStringSeguramente(clientes[contadorCliente].nome, sizeof(clientes[contadorCliente].nome));
        printf("Informe o endereço do cliente %d \n", contadorCliente + 1);
        lerStringSeguramente(clientes[contadorCliente].endereco,
sizeof(clientes[contadorCliente].endereco));
        printf("Informe a idade do cliente %d \n", contadorCliente + 1);
        scanf("%d", &clientes[contadorCliente].idade);
        flush_in();
        printf("Informe o telefone do cliente %d \n", contadorCliente + 1);
        lerStringSeguramente(clientes[contadorCliente].telefone,
sizeof(clientes[contadorCliente].telefone));
    }

    /*Manipulando arquivo*/
    FILE *ponteiroArquivo;

    ponteiroArquivo = fopen("clientes.txt", "w");

    /*Verifica a disponibilidade de acesso ao arquivo*/
    if (ponteiroArquivo == 0) {
        printf("ERROR - Lista de clientes não disponível !!! \n\n");
        exit(1);
    }

    /*Laço para escrever cada cliente no arquivo*/
    for (contadorEscritaArquivo = 0; contadorEscritaArquivo < CLIENTES; contadorEscritaArquivo++) {
        fprintf(
            ponteiroArquivo,
            "CLIENTE: %d --- Nome: %s - Endereço: %s - Idade: %d - Telefone: %s \n",
            contadorEscritaArquivo+1,
            clientes[contadorEscritaArquivo].nome,
            clientes[contadorEscritaArquivo].endereco,

```

```
    clientes[contadorEscritaArquivo].idade,  
    clientes[contadorEscritaArquivo].telefone);  
}  
  
printf("SUCCESS - Dados cadastrados em arquivo !!!! \n");  
  
fclose(ponteiroArquivo);  
  
}
```