

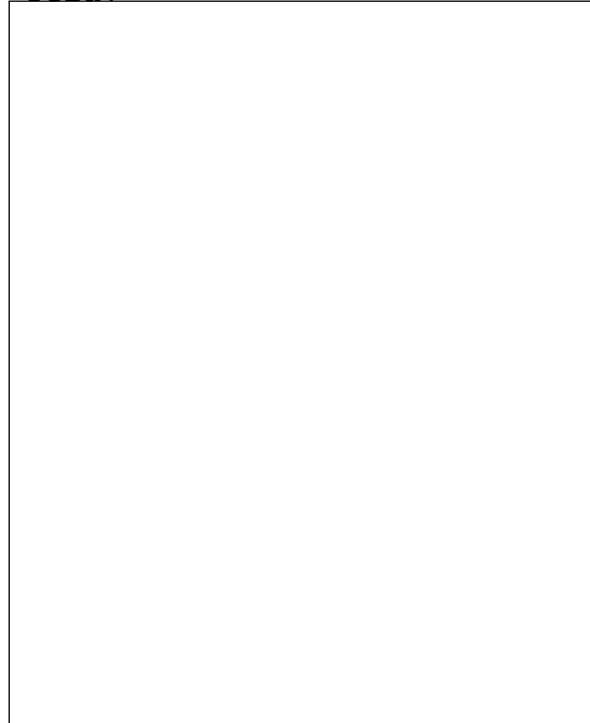
Exercício G2:

1. Faça o teste de mesa para o programa abaixo e indique o que será impresso na tela.

```
# include <conio.h>
# include <stdio.h>
void troca(int x,int y)
{
    int aux,z;
    aux = x;
    x = y;
    y = aux;
    z = 40;
}
int soma(int *x, int *y, int *z)
{
    int aux;

    aux = *x;
    *x = *y;
    *z=100;
    *y = aux;
    return (*x+*y)
}
void main (void)
{
    int x,y,z;
    x=200;
    y=90;
    z= y + 1;
    printf ("\n%d",x);
    printf ("\n%d",y);
    printf ("\n%d",z);
    troca(x,y);
    printf ("\n%d",x);
    printf ("\n%d",y);
    printf ("\n%d",z);
    z=soma(&x,&y);
    printf ("\n%d",x);
    printf ("\n%d",y);
    printf ("\n%d",z);
    printf ("\n%d",z);
}
```

Tela:



2. Um programa precisa fazer a leitura de um número indeterminado de nomes (no máximo 100), os quais devem ser armazenados em uma matriz de 100 linhas e 30 colunas. A leitura deve ser encerrada quando o nome “FIM” for informado ou quando todas as linhas da matriz estiverem preenchidas. OBS: o nome fim não deve ser armazenado na matriz. Analise os códigos descritos abaixo e marque a alternativa que corresponde fielmente ao requisito apresentado (0,75).

a)(_) #include<conio.h>
#include<stdio.h>
#include<string.h>
void main ()
{
Char nomes[30][100];
Int i,j;

b)(_) #include<conio.h>
#include<stdio.h>
#include<string.h>
void main ()
{
Char nomes[100][30], nome[30];
Int i,j;

```

for(i=0;i<30;i++)
for(j=0;j<100;j++)
{ printf("\nDigite um nome: ");
gets(nomes[i]);
if ((nomes[i]=="FIM"))
break;
}
}

```

```

c)(_) #include<conio.h>
#include<stdio.h>
#include<string.h>
void main ()
{
Char nomes[100][30], nome[30];
Int i;
for(i=0;i<100;i++)
{ printf("\nDigite um nome: ");
gets(nome);
if (strcmp(nome,"FIM")==0)
break;
strcpy(nomes[i],nome)
}
}

```

```

e)(_) #include<conio.h>
#include<stdio.h>
#include<string.h>
void main ()
{
Char nomes[100][30];
Int i;
for(i=0;i<100;i++)
{ printf("\nDigite um nome: ");
gets(nomes[i]);
if (strcmp(nomes[i],"FIM")==0)
break;
}
}

```

```

for(i=0;i<100;i++)
for(j=0;j<30;j++)
{ printf("\nDigite um nome: ");
gets(nome);
if (strcmp(nome,"FIM")==0)
break;
nomes[i]=nome;
}
}

```

```

d)(_) #include<conio.h>
#include<stdio.h>
#include<string.h>
void main ()
{
Char nomes[100][30], nome[30];
Int i;
for(i=0;i<100;i++)
{ printf("\nDigite um nome: ");
gets(nome);
if (nome=="FIM")
break;
strcpy(nomes[i],nome);
}
}

```

3. Escreva um programa que leia duas matrizes de 4 linhas e 4 colunas compostas por valores reais. Após a leitura das matrizes, o programa deve mostrar o número de valores iguais em posições coincidentes (iguais). A impressão do número de valores iguais deve ser realizada no main. (main 0,5)
Para a elaboração do programa utilize os seguintes módulos:

Le_Matriz (0,75)

Descrição: faz a leitura de uma matriz 4 x 4 (valores reais).

Procura_posicoes_iguais (0,75)

Descrição: tem como entrada duas matrizes 4 X 4 (preenchidas com números reais) e tem como saída o número(inteiro) de elementos iguais em posições iguais.

Exemplo:

M1

2	55	3	7
10	4	30	86
22	81	34	77
32	1	19	6

M2

4	5	77	17
10	64	9	86
3	45	84	21

32	1	65	6
----	---	----	---

Número de elementos iguais em posições iguais: 4

4. Faça um programa na linguagem C para controlar as informações sobre o cardápio de uma confeitaria. A confeitaria trabalha com bolos e docinhos. Cada produto possui as seguintes informações: nome, tipo (bolo ou docinho) e preço. O programa deve ler os 30 itens do cardápio e armazenar em um vetor do tipo struct com todos os dados dos produtos. Após a leitura, o programa deve escrever a média dos preços dos bolos e os nomes e valores do docinho mais barato e do docinho mais caro. (main 0,5)

Para a elaboração do programa defina as seguintes funções:

Incluir_doces (0,5)

Descrição: recebe como entrada um vetor do tipo struct vazio e faz a leitura dos 30 doces do cardápio.

Entrada: vetor de struct para ser preenchido com os 30 doces.

Saída: nenhuma.

Calcula_media_de_preco (1,0)

Descrição: recebe como entrada o vetor do tipo struct com os doces e calcula a média dos preços dos bolos.

Entrada: vetor de struct com os 30 doces cadastrados.

Saída: um valor real referente a média dos preços dos bolos.

Imprime_barato_caro (1,0)

Descrição: recebe o vetor de doces e procura o docinho mais barato e o docinho mais caro e escreve ambos na tela.

Entrada: vetor de struct com os 30 doces cadastrados.

Saída: nenhuma.

Boa Prova!