

Exercícios de Fixação **04** – Prof.^a Myriam
Tópicos : Constantes, Variáveis e Expressões em C

1) As expressões a seguir envolvem que tipos de constantes ?

- a) $2+10+3$
- b) $(3.31e-8 + 2.10E-7)/(7.16 + 2.1E2)$
- c) 'a' + 'b' + 'c' + 'd'
- d) $(4.2f+5.3f) * (2.7 - 3.4567) / (3.1e2 + 4.5e2)$

2) Quais das seguintes constantes são INVÁLIDAS? Por quê ?

- No caso das válidas define de que tipo são.

123.456	0001	0xABCDEF	0Xabcv
+123	05abc	0777	-12E-12
0089	0xabcd	98.6F	17777
0X0G1	0xab05	.0001	0xFFFF
-597.25	123.5e2		

3) Quais dos seguintes nomes de variáveis são INVÁLIDOS ? Por quê ?

Int	alfa_beta_rotina
char	f
6_05	_1312
_var1	Reinicializa
Xs	A\$
z	

4) Avalie as expressões abaixo e determine quais são Verdadeiras($\neq 0$) e quais são Falsas(0) :

- a) $10 == (9+1)$
- b) $10 \&\& 8$
- c) $8 \parallel 0$
- d) $0 \&\& 0$
- e) $0 \&\& 8$
- f) $x = 10;$
 $y = 9;$
 $(x >= 8) \&\& (y <= x)$

5) Comente o que cada declaração faz e o valor das variáveis, na seguinte seqüência de comandos :

```
char c,d;
int X;
int var_a=2;
float z, y=2.7f;
c = 'a';
d = c;
x = 8%5;
var_a += 10;
++var_a;
z = y + 3.0f;
```

6) Determine a ordem de execução das seguintes expressões e coloque o resultado com os parênteses representando a ordem de execução correta :

- a) $d = a + b * c;$
- b) $x - a + b * c / d$
- c) $x - (a+b) * c / d;$
- d) $w = 2 * i \% 5 * 4 + (j-3) / (k + 2);$

Exercícios de Fixação 05 – Prof.^a Myriam
 Conteúdo : Declaração e Entrada-Saída em C

1) Faça a seguinte declaração de variáveis em C :

inteiro : NFILHOS, IDADE;

caractere: LETRA;

real : VALOR, PESO;

- inicialize as variáveis declaradas e imprima os valores atribuídos.

2) Supondo que as variáveis NOTA_ALUN, INIC_NOME, NMAT, SX sejam utilizadas para armazenar a nota do aluno, a inicial do nome do aluno, o número de matrícula e o sexo, declare-as corretamente, atribua valores e imprima os resultados.

3) Passe para C os algoritmos a seguir:

a) início { Algoritmo para o cálculo da Média Final }

real : P1, P2, P3, MF;

imprima("Entre com o valor das Parciais P1, P2, P3");

leia (P1, P2, P3);

MF ← (P1 + P2 + P3) / 3.0;

imprima ("A Média Final é: ", MF);

fim

b) início

real : H, R, C, QLAT, AREA, LITRO;

imprima("Entre com o valor da Altura e do Raio");

leia (H, R);

AREA ← (3.14 * R ** 2) + (2 * 3.14 * R * H);

LITRO ← AREA / 3.0;

QLAT ← LITRO / 5.0;

C ← QLAT * 10.00;

imprima ("O Custo da Pintura é de R\$ ", C);

imprima ("O Gasto Total de Latas é de ", QLAT);

fim

c) início

inteiro : X, Y;

real : Z;

imprima("Entre com o valor de X");

leia (X);

imprima(X, "ao cubo = ", X ** 3);

imprima("Entre com o valor de Y");

leia(Y);

imprima("O resultado de X + Y = ", X + Y);

Z ← X / Y;

imprima (Z);

Z ← int(Z);

imprima(Z);

Z ← Z + 1;

X ← (Y + Z) res 2;

imprima(X);

fim

Exercícios de Fixação 06 – Prof.^a Myriam
Algoritmos em C : Estruturas de Seleção

1) Considere programa a seguir :

```
#include <stdio.h>
void main(void)
{
    float valor1,valor2;
    char operador;

    printf("Entre com a expressão : \n");
    scanf("%f %c %f", &valor1, &operador, &valor2);

    if(operador == '+')
        printf("Resultado : %.2f\n",valor1+valor2);
    else if (operador == '-')
        printf("Resultado : %.2f\n",valor1-valor2);
    else if (operador == '*')
        printf("Resultado : %.2f\n",valor1*valor2);
    else if (operador == '/')
        printf("Resultado : %.2f\n",valor1/valor2);
    else printf(" Operador desconhecido\n");
}
```

Defina a saída na tela quando os valores de entrada forem :

- 123.5 + 59.3
- 198.7 / 26
- 89.3 * 2.5

2) Rescreva o programa anterior utilizando o **switch** teste e verifique se os resultados são idênticos ao caso anterior

3) Faça um algoritmo utilizando **escolha** e passe para C utilizando o **switch**, um programa que leia dois valores A e B, calcule e imprima o resultado da operação escolhida pelo usuário conforme a tabela a seguir :

- 's' → some os valores de A e B
- 'b' → subtraia os valores de A e B
- 'm' → multiplique os valores de A e B
- 'd' → divida os valores de A e B
- 'q' → calcule o quociente da divisão de A por B
- 'r' → calcule o resto da divisão de A por B
- 'e' → calcule o resultado de A elevado a B
- 'z' → calcule o resultado de raiz A^{ésima} de B, ou seja, B elevado a (1/A)

O programa deve imprimir na tela o MENU acima informando ao usuário quais as opções disponíveis.

4) Dados 3 valores A, B, C (fornecidos pelo usuário) construir um algoritmo para verificar se estes valores formam um triângulo, e em caso afirmativo, classificá-lo como equilátero, isósceles ou escaleno (imprimir mensagem caso não formem um triângulo).

5) Passar para C todos os exercícios da lista Fixação 02 –Estruturas de Seleção .

Exercícios de Fixação 07 : (prof.^a Myriam)
Estruturas de Seleção e Laços de Repetição em C

1) Escreva em C todos os algoritmos pedidos na lista de exercícios de algoritmos de repetição (Fixação 03)

2) Passe para C e determine a saída na tela para os algoritmos a seguir :

a)

início { Algoritmo para o cálculo da Média Final de uma classe no CEFET onde o total de alunos é fornecido pelo usuário }

real:P1, P2, P3, MF;

inteiro: CONTAL, NALUNOS;

caractere : NOME; {só a primeira letra do nome}

leia(NALUNOS); {Modificação}

para CONTAL=1 até CONTAL <=NALUNOS passo 1 faça

leia(NOME);

leia(P1,P2,P3);

leia(FREQUENCIA);

MF←(P1+P2+P3)/3.0;

se MF>=7.0

imprima(NOME, "Aprovado");

imprima("Media Final = ",MF,"freq.= ",FREQUENCIA);

senão seMF >=4.0

imprima(NOME, "Exame Final");

imprima("Media Final = ",MF);

senão

imprima(NOME, "Reprovado");

imprima("Media Final = ",MF,"e parciais = ", P1,P2,P3);

fim se

fim se

fim para

fim

b) O algoritmo a seguir lê o número de impulsos telefônicos por mês de cada usuário. Calcula o maior consumo e quantos usuários atingiram este valor. O valor de parada é -1 e este valor não deverá ser processado.

início { Exemplo de Repetição com teste no **início** do laço }

inteiro : IMPULSOS, {numero de impulsos por assinante}

QUANTOS, {quantos assinantes atingiram maior número de impulsos}

MAIORPULSOS, {maior número de impulsos encontrados}

MAIORPULSOS ← -1; {inicializado com o menor valor possível }

QUANTOS ← 0;

leia (IMPULSOS);

enquanto IMPULSOS ≠ -1 faça

se IMPULSOS > MAIORPULSOS então

MAIORPULSOS ← IMPULSOS;

QUANTOS ← 1;

senão

se IMPULSOS = MAIORPULSOS então

QUANTOS ← QUANTOS + 1;

fim se

fim se

leia (IMPULSOS);

fim enquanto;

imprima ("Maior número de impulsos no mês : ", MAIORPULSOS);

imprima("Número de assinantes : ", QUANTOS);

fim

c)

inicio { Repetição com teste no fim do laço }caractere : TV; { tipo do vinho }inteiro : CONV, { contador de vinhos }

CT, { contador de tinto }

CB, { contador de branco }

CR; { contador de rosê }

real : PT, PB, PR { porcentagem de tinto, branco e rosê }

CONV ← 0;

CT ← 0;

CB ← 0;

CR ← 0;

repitaimprima("(T)into");imprima("(B)ranco");imprima("(R)ose");imprima("(F)im");imprima("Entre com a Opcao");leia(TV);se TV ≠ 'F' então

CONV ← CONV + 1;

escolha(TV)caso 'T' : CT ← CT + 1;caso 'B' : CB ← CB + 1;caso 'R' : CR ← CR + 1;caso contrário: CONV ← CONV - 1;fim escolhafim seaté TV = 'F'se CONV > 0 então

PT ← (CT *100) / CONV;

PB ← (CB *100) / CONV;

PR ← (CR *100) / CONV;

imprima ("Porcentagem de Tintos = ", PT);imprima ("Porcentagem de Brancos = ", PB);imprima ("Porcentagem de Roses = ", PR);senãoimprima("Nenhum tipo foi fornecido");fim sefim

Fixação 08 : (prof.^a Myriam)
Estrutura de Dados Homogêneas Unidimensionais em C - Vetores

1) Dado um vetor VET declarado como:

int VET[5];

- preenchê-lo (todas as posições) com o valor 30;
- preenchê-lo com os inteiros 1,2,3 ...;
- preenchê-lo com 1 se a posição é par e com 0 se a posição é ímpar
- refazer a declaração e o os itens a), b) e c) considerando-se agora um vetor de 100 elementos

2) Escreva um programa que gere e imprima um vetor Y obtido pela inversão da ordem de 200 valores numéricos lidos do teclado.

Exemplo:

X

32	17	12.5	14	193.7	15.8
----	----	------	-------	----	-------	------

Y

15.8	193	14	12.5	17	32
------	-----	----	-------	------	----	----

3) Escreva um programa que calcule e escreva o somatório dos valores armazenados num vetor V de 100 elementos lidos via teclado

Exemplo:

32	17	12.5	14	193.7	15.8
----	----	------	-------	----	-------	------

4) Fazer um programa que

- leia dois vetores contendo, cada um, 25 elementos numéricos.
- Intercale os elementos destes dois conjuntos formando um novo vetor de 50 elementos
- Imprima o novo vetor obtido

Exemplo:

A

31	70	125	14	193	15
----	----	-----	-------	----	-----	----

B

5	40	56	7	51	101
---	----	----	-------	---	----	-----

C

31	5	70	40	125	56	14	7	193	51	15	101
----	---	----	----	-----	----	-------	----	---	-----	----	----	-----

5) Elabore um programa que leia 100 valores numéricos e os armazene no vetor A . O programa deve calcular e imprimir

$$S = \sum_{i=1}^{100} \frac{i}{a_i}, \text{ onde } a_i \text{ é o } i\text{-ésimo valor armazenado no vetor A}$$

6) Faça um algoritmo que receba um vetor X de 10 elementos e calcule a média dos valores de X. A seguir o programa deve receber um outro vetor Y de 10 elementos e calcular quais os valores de Y, estão acima da média calculada sobre os elementos de X.

7) Faça um programa que leia um vetor vet de 100 valores inteiros e calcule para cada posição i o valor

vet^i

Exemplo

10	5	70								15
----	---	----	--	--	--	--	--	--	--	----

10^0 5^1 70^2

8) O que significa cada uma das constantes a seguir ?

- "A"
- 'A'
- "Curitiba"
- "Curitiba\n"

Declare variáveis para receber cada uma destas constantes

Imprima os valores atribuídos

9) Suponha que você tenha feito um programa com a seguinte declaração

```
char VetCarac[10];
```

e que o usuário deverá fornecer esta string via teclado. Qual o comando para ler este vetor e qual o máximo de caracteres que o usuário deve fornecer ?

10) Fazer um programa para ler os dados de 40 alunos

- Nome de cada aluno
- A média final de cada aluno
- seu número de matrícula

O programa deve imprimir para cada aluno o seu nome e se ele está aprovado, reprovado ou em exame (regras do CEFET)

Calcular a média da turma e quantos ficaram acima da média da turma

Qual o número de matrícula do aluno com a maior nota

11) Implemente um programa que leia uma string (de no máximo 100 caracteres) e conte quantos caracteres a string possui efetivamente.

12) Escreva um programa que leia uma string de no máximo 80 caracteres e

- conte quantos espaços em brancos há nela (espaço em branco → ' ')
- conte quantas vezes aparece a letra 'A' ou 'a'

13) Elabore um programa que leia uma string e conte quantas vogais há nela

14) Elabore um programa que leia string de no máximo 200 caracteres. O programa deve inserir um caractere escolhido pelo usuário em uma posição (também escolhida pelo usuário) do vetor. O programa deve utilizar dois vetores: o original e o modificado e imprimir os dois no final.

15) Elabore um programa que leia dois vetores numéricos A e B de 30 elementos cada e uma string com 30 operadores matemáticos (+, -, *, /). O programa deve efetuar as operações entre cada elemento de A e B seguindo as operações que estão em C.

31	70	125	14	193	15	A
+	-	-	/	+	*	C
5	40	55	7	51	10	B
35	30	70		2	244	150	

Exercícios de Fixação 09 : (prof.^a Myriam)
Estrutura de Dados Homogêneas Multidimensionais em C - **Matrizes**

1) Elabore um programa que declare, inicialize e imprima a seguinte matriz :

MAT

1	0	2	-1
4	3	2	1
1	-2	3	4
8	5	1	3

2) Modifique o programa anterior para que o usuário forneça os valores da matriz MAT

O programa deve ainda calcular e imprimir

- a soma de cada linha
- a soma da diagonal principal

3) Uma matriz M pode ser transposta em uma matriz N onde o número de linhas/colunas da primeira passa a ser o número de colunas/linhas da segunda. Faça um implemente em C um programa que

- leia uma matriz M (o usuário deve definir fornecer a quantidade de linhas e a quantidade de colunas que não deverão exceder 100 linhas e 80 colunas)
- calcule a transposta N da matriz M
- imprima a transposta calculada

4) Elabore um algoritmo que multiplique duas Matrizes M e N fornecidas pelo usuário. O programa deve :

- ler as matrizes M e N
- calcular a multiplicação e armazenar na matriz Mult
- imprimir o resultado

5) A tabela a seguir contém vários itens que estão estocados em vários armazéns de uma companhia. É fornecido também o custo de cada um dos produtos armazenados.

	Produto 1 (R\$7,00 unid)	Produto 2 (R\$8,00 unid)	Produto 3 (R\$3,00 unid)
Armazém 1	1200	3700	3737
Armazém 2	1400	4210	4224
Armazém 3	2000	2240	2444
Armazém 4	1400	3360	1870

a) Faça utilizando vetores unidimensionais e multidimensionais, um algoritmo que :

- leia o estoque inicial (o usuário deve fornecer os valores que estão na tabela acima)
- determine e imprima quantos itens estão guardados em cada armazém
- imprima qual o armazém (1,2,3, ou 4) possui maior quantidade de produtos
- imprima o custo total de
 1. cada produto em cada armazém
 2. estoque em cada armazém
 3. cada produto em todos os armazéns

6) O tempo que um determinado avião leva para percorrer o trecho entre duas localidades distintas está disponível através da seguinte tabela

cidade	0	1	2	3	4	5	6
0	0	2	11	6	15	11	1
1	2	0	7	12	4	2	15
2	11	7	0	11	8	3	13
3	6	12	11	0	10	2	1
4	15	4	8	10	0	5	13
5	11	2	3	2	5	0	14
6	1	15	13	1	13	14	0

Construa um algoritmo e implemente em C um programa que

- inicialize um matriz com os valores ilustrados na tabela acima
- informe ao usuário o tempo necessário para percorrer duas cidades por ele fornecidas, até o momento em que ele fornecer duas cidades iguais (fonte = destino)
- o programa deve ainda permitir calcular o tempo necessário para percorrer um trecho formado por várias cidades. O trecho é fornecido pelo usuário como uma seqüência de cidades até que o valor -1 seja digitado.

7)Elabore um programa que

declare inicialize a seguinte matriz Nomes com o seu nome e o de mais 3 amigos
 imprima os valores atribuídos

8) Modifique o programa anterior 7) para que o usuário forneça os nomes que serão armazenados na matriz

9) Modifique o programa anterior 8) para que ele possa imprimir

- Quantas vogais há em cada nome
- Qual dos nomes é o que tem mais caracteres (mais longo) ?

10) Fazer um algoritmo que inicialize duas matrizes (função strcpy - copia uma string em outra)

NOME (com o nome de 10 pessoas) e NATURAL (seus respectivos locais de nascimento). O programa deve ainda receber como entrada o nome de uma pessoa e localizar o seu local de nascimento; ou ainda o local de nascimento e localizar o nome da pessoa. (para isso utilize a função strcmp(compara duas strings).

11) Elabore um programa que manipule os dados para uma sala de 40 alunos. O programa deve receber como entrada (via leitura)

Os nomes dos alunos

As três parciais

A freqüência de cada aluno

O programa deve calcular e imprimir

O nome

A média final

A freqüência

A maior parcial e a menor parcial

dos alunos com média final acima da média da turma

Exercícios de Fixação 10: (prof.^a Myriam)
Funções: protótipo, definição e chamada – escopo de variáveis

1)

1.1) Cada item abaixo contém um esquema genérico da definição de algumas funções

a) float TESTE1(float a, float b) <pre>{ /*dec var locais*/ Comandos }</pre>	c) long int FAT(long int a) <pre>{ /*dec var locais*/ Comandos ... }</pre>	e) char FUNC(char c1, char c2) <pre>{ /*dec var locais*/ Comandos... }</pre>
--	---	---

b) void FUNC2(int a) <pre>{ /*dec var locais*/ Comandos }</pre>	d) char CONT_LETRA(void) <pre>{ /*dec var locais*/ Comandos }</pre>	f) double TESTE2(int a, float b) <pre>{ /*dec var locais*/ Comandos }</pre>
---	---	---

- Defina o PROTÓTIPO DE CADA UMA DAS FUNÇÕES DESCRITAS ACIMA

1.2) Cada item abaixo contém um esquema genérico da definição de algumas funções do tipo VOID

a) void TESTE1(float a, float b) <pre>{ /*dec var locais*/ Comandos }</pre>	c) void FAT(long int a) <pre>{ /*dec var locais*/ Comandos ... }</pre>	e) void FUNC(char c1, char c2) <pre>{ /*dec var locais*/ Comandos... }</pre>
---	---	---

b) void FUNC2(int a) <pre>{ /*dec var locais*/ Comandos }</pre>	d) void CONT_LETRA(void) <pre>{ /*dec var locais*/ Comandos ... }</pre>	f) void TESTE2(int a, float b) <pre>{ /*dec var locais*/ Comandos... }</pre>
---	--	---

- Faça a estrutura geral de **UM** programa em C contendo os protótipos, as chamadas e as definições de todas as funções descritas anteriormente. Nas chamadas poderão ser usadas constantes ou variáveis.

2) Refaça o programa do cálculo da combinação de N elementos tomados P a P, usando funções (utilize somente variáveis GLOBAIS)

3) Faça um programa que

- leia as coordenadas de um quadrado (x1,y1) (x2,y2) (x3,y3) (x4,y4)
- calcule o perímetro do quadrado formado pelas coordenadas lidas.
- Imprima o perímetro do quadrado

OBS cada tarefa deve ser realizada por uma função
 Utilize apenas variáveis GLOBAIS

2) Faça um programa que

- leia 3 valores fornecidos pelo usuário
- verificar se estes parâmetros formam um triângulo e classificá-lo como equilátero, isósceles ou escaleno (imprimir mensagem caso não formem um triângulo)

OBS cada tarefa deve ser realizada por uma função
 Utilize apenas variáveis GLOBAIS

- 3) Refaça o programa do cálculo da combinação de N elementos tomados P a P, usando função (utilize somente variáveis LOCAIS)
- 4) Escreva uma função que receba como parâmetros os comprimentos dos lados de um triângulo (a,b,c) e retorne os seguintes valores:
- 1 se o triângulo for retângulo
 - 2 se o triângulo for obtusângulo
 - 3 se o triângulo for acutângulo
 - 0 se nenhum triângulo é formado
- 5) Faça um programa que leia os lados do triângulo (a,b,c) e utilize a função anterior imprimir o tipo do triângulo formado. No programa desenvolvedor, defina o escopo das variáveis utilizadas pelo seu programa.
- 8) Faça uma função que receba as quatro coordenadas de um quadrado ($x_1, y_1, x_2, y_2, x_3, y_3, x_4, y_4$), verifique se realmente os 4 lados são iguais retorne o perímetro do quadrado.
- 9) Faça um programa que leia as coordenadas de um quadrado e utilize a função anterior para calcular o perímetro do quadrado formado pelas coordenadas lidas. A função principal deve passar para a função anterior, as coordenadas. O programa deve ainda indicar se o perímetro está ou não acima de determinado limiar fornecido pelo usuário. Defina o escopo das variáveis utilizadas pelo seu programa.
- 10) Escreva um programa que leia um valor e imprima
- a) se este é par ou ímpar
 - b) se é divisível por 5
 - c) seu valor absoluto (módulo)
 - d) seu fatorial
 - e) sua tabuada
- Obs. Utilize uma função para executar cada uma das tarefas anteriores e defina o escopo das variáveis utilizadas no seu programa

11) Escreva uma função que calcule o valor de π através da série

$$S = 1 - \frac{1}{3^3} + \frac{1}{5^3} - \frac{1}{7^3} + \dots, \text{ sendo } \pi = \sqrt[3]{S \times 32}$$

Deverá ser fornecido à função o número de termos da série para o cálculo de π

12) Escrever um programa que fornecendo à função anterior, sucessivamente, o número de termos (1,2,3,...,N), imprima uma tabela com o valor de π e o número de termos utilizados. O valor de N deverá ser fornecido pelo usuário. Defina o escopo das variáveis utilizadas pelo seu programa

13) mínimo múltiplo comum (mmc) de dois inteiros positivos u e v é o menor positivo inteiro que é divisível tanto por u quanto por v. Então, o mmc de 15 e 10, dado por mmc(15,10) é 30 uma vez que 30 é o menor inteiro divisível tanto por 15 quanto por 10. Escreva um programa que calcule o mínimo múltiplo comum de dois inteiros através de uma função chamada mmc que recebe dois argumentos inteiros e retorna o seu mmc. A função mmc deve calcular o mínimo múltiplo comum chamando o máximo divisor comum

$$\text{MinMultCom} = u \cdot v / \text{mdc}(u,v) \quad u,v \geq 0$$

Onde o mdc é definido como mostrado na figura ao lado:

Defina o escopo das variáveis utilizadas pelo seu programa

```
int mdc ( int x, int y)
{
    int temp;
    while( y != 0 )
    {
        temp = x % y;
        x = y;
        y = temp;
    }
    return( x);
}
```

Exercícios de Fixação 11 : (prof.^a Myriam)
funções com passagem de parâmetros por valor e referência

- 1) A função pot(m,n) a seguir eleva um inteiro m à potência inteira positiva n. Por exemplo, o valor de pot(2,5) é 32.

```
int pot( int m, int n)
{ int p;
  for(p=1;n>0;n--)
    p = p*m;
  return(p);
}
```

- Faça um programa que utilizando a função definida anteriormente, calcule e imprima uma table contendo as potências de valores uma seqüência de valores que vão sendo fornecidos pelo usuário.
- Modifique a função anterior para o tipo void (não retorna nenhum valor)
 obs. Para isto é necessário que p seja fornecido e o seu conteúdo possa ser alterado (passagem por referência)
- Escreva um programa que utilize a função modificada no item b) de forma análoga ao item a)

- 2) Faça o mesmo para a função MDC listada a seguir

```
int MDC(int u, int v)
{
  int temp;
  while(v!=0)
  {
    temp = u%v;
    u = v;
    v = temp;
  }
  return(u);
}
```

Defina o escopo das variáveis e a saída na tela do programa a seguir:

```
#include<stdio.h>
void f1(int);
void f2(int *);
void main(void)
{
  int A=2;
  printf("O valor da var. A é = %i e o seu endereco na memória é %x\n",A,&A);
  f1(A);
  printf("O valor de A é %i\n",A);
  f2(&A);
  printf("O valor de A é %i\n",A);
}

void f1(int A)
{
  printf("O valor da var. A é = %i e o seu endereco na memória é %x\n",A,&A);
  A = A+2;
  printf("O valor da var. A é = %i e o seu endereco na memória é %x\n",A,&A);
}

void f2(int *A)
{
  printf("O valor da var. A é = %i e o seu endereco na memória é %x\n",*A,A);
  *A = *A+2;
  printf("O valor da var. A é = %i e o seu endereco na memória é %x\n",*A,A);
}
```

- 4) Refazer os exercícios da lista de fixação 08 (funções), utilizando apenas variáveis locais e funções do tipo VOID (funções que não retornam nada).

5) Elabore um programa que leia os salários de funcionários de uma firma, ate que se digite o valor 0 .O algoritmo deve calcular e imprimir :

- O maior salário da firma e o menor salário da firma
- A média dos salários

ObsA tarefa a) deve ser executada por uma função específica e NÃO é permitido o uso de variáveis globais.

6) Deseja-se fazer uma pesquisa a respeito do consumo mensal de energia elétrica em uma determinada cidade. Para isso são fornecidos os seguintes dados :

- preço do kWh consumido
- numero do consumidor
- quantidade de Kwh consumidos durante o mês
- código do tipo de consumidor (residencial, comercial, industrial)

O número do consumidor igual a zero deve ser usado como condição de parada (flag).

Fazer um algoritmo que calcule e imprima :

- para cada consumidor o total a pagar
- o maior consumo verificado e o menor consumo verificado
- o total do consumo para cada um dos três tipos de consumidores
- a média geral de consumo

Obs. Cada item (exceto o último) deve ser calculado através de uma função, as impressões devem ser feitas na função principal e só é permitido o uso de variáveis locais

7) Numa certa loja de eletrodomésticos, o funcionário encarregado da seção de televisores recebe, mensalmente, um salário fixo mais comissão. Essa comissão é calculada em relação ao tipo e ao número de televisores vendidos por mês, obedecendo à tabela abaixo :

TIPO	Nº. De Televisores vendidos	Comissões
A cores	≥ 10	R\$ 100,00 por televisor vendido
	< 10	R\$ 50,00 por televisor vendido
Preto e branco	≥ 20	R\$ 40,00 por televisor vendido
	< 20	R\$ 20,00 por televisor vendido

Sabe-se ainda que ele tem um desconto de 8% sobre o seu salário fixo para o INSS.

Se o seu salário total (fixo + comissões - INSS) for maior ou igual a R\$3.000,00 ele ainda terá um desconto de 5% sobre o salário total, relativo ao imposto de renda retido na fonte.

Sabendo-se que existem 20 empregados nesta seção, faça um programa que leia

- valor do salário fixado pela empresa e,
- para cada empregado,
- número de sua inscrição,
- número de televisores vendidos (a cores e preto e branco).

Calcule e imprima

- número de inscrição de cada empregado
- seu salário bruto
- seu salário líquido
- o menor e o maior salário do mês entre todos os vendedores

Obs. Cada tarefa de cálculo e execução deve ser feita por uma função específicaExercícios de

Exercícios de Fixação 12: (prof.^a Myriam)
Funções, Vetores, Matrizes e Registros.

- Defina o escopo das variáveis e a saída do programa a seguir

```

#include <stdio.h>
void le_vetor(void);
void imprime_vetor(void);
void imprime_vetor_cont(void);
int DIM=100,VET[100];

void main(void)
{
    le_vetor( );
    imprime_vetor_cont( );
}

void le_vetor(void)
{
    int I;
    for(I=0;I<100;I++)
        scanf("%d",&VET[I]);
}

void imprime_vetor(void)
{
    int I;
    for(I=0;I<100;I++)
        printf("%d ",VET[I]);
    printf("\n");
}

void imprime_vetor_cont(void)
{
    int I,vet2[100];
    for(I=0;I<100;I++)
        vet2[99-I] = VET[I];
    imprime_vetor( );
    for(I=0;I<100;I++)
        printf("%d ",vet2[I]);
    printf("\n");
}

```

2)Escreva um programa que leia um vetor de caracteres e defina

- Quantas vogais existem no vetor
 - Quantas letras C ou c existem no vetor
 - Cada tarefa deve ser executada por uma função que recebe o caractere e retorna 1 ou 0
-

3)Escreva um programa que leia um vetor de caracteres e

- Coloque um asterisco nas letras que forem vogais
 - Coloque um sinal de exclamação nas letras C ou c
 - Cada tarefa deve ser executada por uma função que recebe o caractere para verificação e retorna o caractere a ser colocado
-

4) Refaça o programa anterior para que o elemento a ser verificado possa ser alterado dentro da própria função (não passar o vetor todo como argumento).

5)Escreva uma função chamada soma_vet que toma dois argumentos: um vetor de inteiros e o total de elementos no vetor. A função deve retornar a soma dos elementos no vetor de elementos.

6)Escreva um programa que utilize a função implementada no exercício anterior .

- defina o escopo das variáveis
- reescreva o programa utilizando somente variáveis globais

7)Escreva uma função chamada max_vet que toma dois argumentos: um vetor de inteiros e o total de elementos no vetor. A função deve retornar o maior elemento do vetor.

8)Escreva um programa que utilize a função implementada no exercício anterior.

- defina o escopo das variáveis
 - reescreva o programa utilizando somente variáveis globais
-

9)Escreva uma função chamada troca_par que toma dois argumentos : um vetor de inteiros e o total de elementos no vetor. A função deve alterar os elementos do vetor que forem números pares. Neste caso o novo valor deve ser 0.

10)Escreva um programa que

- leia um vetor de caracteres
- Coloque um asterisco nas letras que forem vogais
- Coloque um sinal de exclamação nas letras C ou c
- Imprima o vetor resultante

Cada tarefa deve ser executada por uma função

11)Refazer os problemas da lista de fixação 10 (vetores) utilizando uma função para cada tarefa pedida.

12) Uma matriz M com I linhas e j colunas pode ser transposta em uma matriz N tendo J linhas e I colunas, simplesmente fazendo o valor N(a,b) igual a M(b,a) para todos os valores de a e b. Escreva um programa que:

- leia a matriz
- encontre a transposta
- imprima a matriz e a transposta

Cada tarefa deve ser feita por uma função (utilize variáveis globais)

13) Refaça o exercício anterior utilizando apenas variáveis locais (as matrizes devem ser passadas como argumentos)

14) Faça um programa que

- Leia um conjunto de nomes
- Defina o total de vogais em cada um
- Defina o total de letras em cada um
- Qual o nome mais longo
- Imprima os nomes de trás para frente

Cada tarefa deve ser executada por uma função específica (utilize apenas variáveis locais)

15)Refazer os problemas da lista de fixação 10(vetores) e 11(matrizes) utilizando uma função para cada tarefa pedida.

16) Refazer os exercícios 06) e 07) da lista de fixação 12) (registros) utilizando funções para cada tarefa pedida.