

## 2ª LISTA DE EXERCÍCIOS DE LINGUAGEM C

1. Fazer um programa ler um vetor de inteiros e positivos e imprimir quantas vezes aparece o número 1, 3 e 4, nesta ordem. O vetor terá *no máximo* 100 posições. Sair do programa quando for digitado -1.
2. Fazer um programa em C para ler uma quantidade N de alunos. Ler a nota de cada um dos N alunos e calcular a média aritmética das notas. Contar quantos alunos estão com a nota acima de 5.0. Obs.: Se nenhum aluno tirou nota acima de 5.0, imprimir mensagem: *Não há nenhum aluno com nota acima de 5.*
3. Seja N um número quadrado perfeito. Se somarmos os números ímpares consecutivos (1+3+5+7+9+...) até que esta soma seja igual a N, o número M de termos somados será igual a raiz quadrada de N.  
Exemplo: N = 16      16 = 1 + 3 + 5 + 7      M = 4 termos.  
Logo, a raiz quadrada de 16 é 4.  
Fazer um programa em C para ler um número inteiro e positivo N e responder se N é quadrado perfeito.
4. Fazer um programa em C para ler o raio R de uma esfera e calcular o volume e a área de uma esfera.  
$$V = \frac{4}{3} * \pi * R^3 \quad \text{e} \quad A = 4 * \pi * R^2$$
5. Durante uma corrida de automóveis com N voltas de duração foram anotados para um piloto, na ordem, *os tempos registrados em cada volta*. Fazer um programa em C para ler os tempos das N voltas, calcular e imprimir:
  - melhor tempo;
  - a volta em que o melhor tempo ocorreu;
  - tempo médio das N voltas;
6. Fazer um programa em C para calcular a soma dos N primeiros múltiplos de um inteiro K, onde N e K são lidos e são números inteiros e positivos.
7. Dado dois vetores, A (5 elementos) e B (8 elementos), faça um programa em C que imprima todos os elementos comuns aos dois vetores.
8. Faça um programa que determina o máximo e o mínimo de um conjunto de n números inteiros armazenados num vetor A de 10 elementos.
9. Suponha um vetor N com 10 elementos e outro vetor M com 10 elementos. Faça um programa em C que calcule o produto escalar P de A por B. (Isto é,  $P = A[1]*B[1] + A[2]*B[2] + \dots + A[N]*B[N]$ ).
10. Fazer um programa em C que leia uma frase de até 50 caracteres(utilizar o comando gets) e imprima a frase sem os espaços em branco. Imprimir também a quantidade de espaços em branco da frase.
11. Fazer um programa em C para ler um vetor de inteiros positivos de 50 posições. Imprimir a quantidade de números pares e de múltiplos de 5.

12. Fazer um programa que leia vários números(um por vez) e imprima 0(zero) se o número não for múltiplo de 2,3, 5 nem 7. Imprimir 1(um) se o número for múltiplo de 2 ou 3 e 2(dois) se o número for múltiplo de 5 e 7. Sair do programa quando for digitado o valor -1.(Não é necessário nenhuma consistência ou crítica dos dados).
13. Fazer um programa em C que leia uma frase e imprima somente as vogais.
14. Fazer um programa em C para ler vários números inteiros e positivos e calcular a média. Imprimir também o maior.
15. Fazer um programa em C que leia vários números inteiros e positivos e calcule o fatorial se o número é menor que 7 e a soma de 1 até o número lido se o número é maior ou igual a 7. Sair quando for digitado -1.
16. Escreva um programa que solicite dois caracteres de a a z ao usuário e imprima o número de caracteres existente entre eles. Assuma que o usuário digite os 2 caracteres em ordem alfabética. Caso não o estejam, emitir mensagem de erro. Exemplo:

*Digite 2 caracteres : j t*  
*O numero de caracteres entre eles e': 9*

17. Para os programas prontos das páginas seguintes, utilizar a seguinte estratégia:
  - Verificar os conceitos envolvidos no programa;
  - Fazer um teste de mesa do programa, exaustivamente;
  - Processar o programa e comparar o resultado com o seu;

```
#include <stdio.h>
void main()
{
int i,j, num, cont;
char flag;
printf("Informe um n° inteiro:");
scanf("%d",&num);
i=2;
flag='V';
while((i < num/2) && (flag=='V'))
{
if((num%i) == 0)
flag='F';
else
++i;
}
if (flag=='V')
printf("%d ",num);
}
```

Após testar para os números:  
8,10,12, 13,15,17,18,20,27,12  
responder quais os números serão  
impressos e o que faz o programa  
acima?

```
#include <stdio.h>
#define N 10
void main()
{
int v[N],i,aux;
char para;
for(i=0; i<N; ++i)
v[i]=0;
para = 'f';
i=1;
do
printf("Informe um n° Inteiro:");
scanf("%d",&aux);
fflush(stdin);
if(aux==0)
para='v';
else
{
v[i]=aux;
i++;
}
while (para=='f');
for(j=0; j<i; ++j)
printf("%d ",v[j]);
}
```

Ler vários números quaisquer à sua  
escolha e responder o que será impresso.

```
#include <stdio.h>
void main()
{
int n, s, auxiliar;
printf("Informe um n° Inteiro:");
scanf("%d",&n);
s=0;
auxiliar = 1;
while (n >= auxiliar)
{
s+=auxiliar;
auxiliar++;
}
printf("Valor final da Soma: %d", s);
}
```

```
#include <stdio.h>
void main()
{
int i, n, s;
printf("Informe um n° Inteiro:");
scanf("%d",&n);
s = 0;
for(i=1;i<=n;i++)
s+=i;
printf("Valor final da Soma: %d", s);
}
=> Responder o que faz os programas  
Calculo1 e Calculo2 para N=3, 5 e 8.  
Os dois programas fazem a mesma coisa?
```

```
#include <stdio.h>
void main()
{
int x, y, produto, auxiliar;
printf("Informe dois números:");
scanf("%d %d",&x, &y);
produto=0;
auxiliar= y;
while(auxiliar > 0)
{
produto+=x;
auxiliar--;
}
printf("\nResultado:%d", produto);
}
```

Fazer os testes com os valores:

x	y
4	3
3	4
5	2
7	5

O que será impresso?  
O que faz o algoritmo  
com os números testados?

<pre> #include &lt;stdio.h&gt; void main() { float v[5], aux, *p, *q; int i, j; printf("Informe os valores:"); for(i=0; i&lt;5; i++) { scanf("%f", v+i); fflush(stdin); } for(i=0; i&lt;4; ++i) for(j=(i+1); j&lt;5; j++) { p=(v+i); q=(v+j); if (*p &lt; *q) { aux=*q; *q=*p; *p=aux; } } printf("Resultado final:"); for(i=0; i&lt;5; i++) printf("%.2f  ",*(v+i)); } //Testar com valores à sua escolha </pre>	<pre> #include &lt;stdio.h&gt; void main() { int a, b, c=5; printf("Informe dois n°s inteiros:"); scanf("%d %d",&amp;a, &amp;b); while(c--) { if(--a&lt;=5) { printf(" Valor de a:%d\n",a); b++; } else { printf(" Valor de b:%d\n", b); a-=b; } c=(a&lt;b)?a:b; } printf(" Valor de a:%d\n \ Valor de b:%d",a, b); }  // Testar para a=2 e b=-2. </pre>
<pre> #include &lt;stdio.h&gt; void main() { int val, cont1,cont2; printf("Informe dois n°s inteiros:"); scanf("%d %d",&amp;val,&amp;cont2); while(++val&lt;=12) { for(cont1=4;cont1&lt;=6;cont1++) { if (cont1%2==cont2%3) printf("cont1: %d &amp;&amp; \ val: %d\n",cont1, val); else { printf("cont1:%d    \ cont2:%d\n",cont1-1,cont2+1); cont2--; } } } } // Testar para val=10 e cont2=20 </pre>	<pre> #include &lt;stdio.h&gt; void main() { int a, b, c, d; printf("Informe 3 n°s inteiros:"); scanf("%d %d %d",&amp;a,&amp;b,&amp;c); while(c--) { d=(a&lt;b)?(a%2):(b%3); if((--a%2)==0) { printf("Valor de a: %d e valor de \ d: %d\n",a,d); b++; } else { printf("Valor de b:%d e valor de \ d:%d\n",b,d); a-=b; } } } // Testar para a=10, b=4 e c=3 </pre>

<pre> #include&lt;stdio.h&gt; void main() { float vet1[5]={3,-76,0.22,3.00,1.50}, vet2[]={5,4,7.78,-9,12},*p, *q; int j=4,a=0; p=vet1; q=vet2; while(--j!=0) { if(a++&lt;1) *(p+j)+=*(q+j); else *(p+j)=*(q+j)*2; printf("%.2f\n",*(q+j+1)); } } </pre>	<pre> #include &lt;stdio.h&gt; int funcao(int num) { int cont1, cont2=2; for(cont1=2; cont1&lt;4; ++cont1) { if((cont1%2)==(cont1%3)) { ++cont2; continue; } else cont2+=num; } return(cont2); }  void main() { int val=10; while(++val!=16) { printf("%d &amp;&amp; %d\n",val,\ funcao(val)); } } </pre>
<pre> #include &lt;stdio.h&gt; #include &lt;string.h&gt; void main() { char nome1[5]={'j','o','a','o','\0'}, nome2[8]="pedro", *p, *q; int prim, seg,num, cont; prim=strlen(nome1); seg=strlen(nome2); for(num=0;num&lt;prim;++num) { for(cont=0; cont&lt;seg; cont++) { p=nome1+num; q=nome2+cont; if ((*p!=*q)&amp;&amp;(cont%2==0)) printf("%c %c\n",*p, *q); else break; } } } </pre>	<pre> #include &lt;stdio.h&gt; void main() { int j, a, b; printf("Informe dois n°s:"); scanf("%d %d",&amp;a,&amp;b); for (j=1 ; j&lt;10 ; j++) { if ((j%a) != 0) { if ((j%b) !=0) printf("%d - %d\n", j, a); } else printf("%d - %d\n", j, b); }  //Testar para a=3 e b=2 </pre>

<pre> #include &lt;stdio.h&gt; void main() { int a,*b; printf("Informe um n° inteiro:"); scanf("%d",&amp;a); b=&amp;a; while(*b &gt; 10) { (*b)--; (*b)--=2; printf("%d %d\n",*b,a); } } //Testar para a=25 </pre>	<pre> #include &lt;stdio.h&gt; void main() { int a=0,*b; b=&amp;a; if ((*b)==0) { while(a&lt;10) { (*b)++; printf("%d \n", a); } } else { while(a &gt; -5) { (*b)--; printf("%d \n",*b); } } } </pre>
<pre> #include &lt;stdio.h&gt; void main() { int j,a=2,b=3; for (j=1;j&lt;10;j++) { if (((j%a)&gt;b)  ((j % b)&lt;a)) if ((a&gt;b)   (a&lt;(b/2))) { printf("%d # %d\n",a,b); a +=2; } else { b++; printf("%d ## %d\n",j,a); } } } // Testar para a=2 e b=3 </pre>	<pre> #include &lt;stdio.h&gt; void main() { int a,b,c; printf("Informe 3 n°s inteiros:"); scanf("%d %d %d",&amp;a,&amp;b,&amp;c); if((a==0)   (b==3)   (c==3)) { while((b+c) &lt; 15) { if (b &gt;= c) b=c++; else b=++c; printf("%d - %d \n",b,c); } } else { while((a+b) &gt; -5) { if (a &gt;= b) a=b++; else a=++b; b=a--; printf("%d - %d \n",a,b); } // Testar para a=1, b=2 e c=3 } } </pre>

<pre> #include &lt;stdio.h&gt; void main() { int a,c,*b,*d; printf("Informe dois n°s inteiros:"); scanf("%d %d",&amp;a,&amp;c); b=&amp;a; d=&amp;c; while(*d&lt;6) { (*d)=(*b)++; (*d)=(*d)+(++*b); if ((c%a) &lt;=2) { c=(*b)*2+3; } else c=(*b)--; printf("%d %d \n",*d, *b); } } //Testar para a=3 e c=1 </pre>	<pre> #include&lt;stdio.h&gt; #include&lt;string.h&gt; void main() { char frasel[6],frase2[6],*f,*g; int a; printf("Informe uma frase :"); gets(frasel); f=frasel; g=frase2; for(a=0;a&lt;strlen(frasel);++a) { *(g+a)=*(f+a); if(a%2==0) *(g+a)=*(f+a)+1; else if(a%3!=0) (*(f+a))++; } *(frasel+5)='\0'; *(frase2+5)='\0'; printf("frasel=%s\n\ frase2=%s",frasel,frase2); } // Testar para frasel="aeiou" </pre>
<pre> #include&lt;stdio.h&gt; void main() { float *p,*q, vet1[]={3.23,-7.89,6.00,0.22,2.21}, vet2[]={98.01,14.89,0.78,-9.98}; int j=4,a=0,b=0; p=vet1; q=vet2; while(j!=0) { a--j; b=(++a)+1; /* Comando trocado */ if((a+b)%2!=1) { *(p+j)=*(q+j)-2.99; printf("%6.2f\n",*(p+j)); } else if((a+j)%3==0) *(q+j)=*(p+j)-*(q+j); printf("%6.2f\n", *(q+j)); } } /* Após responder o que vai ser impresso, trocar o comando acima por b=(a++)+1 e fazer o exercício novamente */ </pre>	<pre> #include&lt;stdio.h&gt; #include&lt;string.h&gt; #include&lt;ctype.h&gt; void main() { char frasel[7]="1a2c3e", frase2[7], *p,*q; int i,j=0,tam; p=frasel; q=frase2; tam=strlen(p); for(i=0;i&lt;tam;i++) { if(i%2!=0  i%5!=0) { if(*(p+i)&gt;='a'&amp;&amp;*(p+i)&lt;='z') *(q+j)=toupper(*(p+i)); else *(q+j)=(*(p+i))++; } else *(q+j)=*(p+i)+2; j++; } *(q+j)='\0'; printf("\nFrase final:%s",q); } </pre>

## Programas resolvidos e comentados

a) O que faz o seguinte programa?

```
#include <stdio.h>
void main()
{
    int x;
    scanf("%d",&x);
    printf("%d",x);
}
```

### Solução

A primeira linha do programa inclui a biblioteca padrão stdio.h. Esta biblioteca será usada para as funções 'scanf' e 'printf'. No início do programa é declarada a variável 'x', inteira. 'scanf' lê x pelo teclado e printf imprime o valor lido. Em tempo de execução o programa irá esperar até que o usuário entre com um número, e exibirá em seguida este número.

b) Escreva uma função que some dois inteiros e retorne o valor da soma. O objetivo neste exercício é apenas escrever a função. Logo, presume-se que já existe um programa que realizará a sua chamada.

### Solução

A função soma2 fica:

```
int soma2(int a, int b) /* Retorna a soma dos inteiros a e b */
{
    int resultado;
    resultado = a+b;
    return resultado;
}
```

ou de uma forma mais pratica..

```
int soma2(int a, int b) /* Retorna a soma dos inteiros a e b */
{
    /* Versao 2 */
    return(a+b);
}
```

c) Escreva um programa que leia uma string (use gets()) e retire todos os caracteres 'c' da string lida. Informe na tela quantos caracteres foram retirados.

Solução: Uma primeira alternativa para o problema seria substituir toda letra 'c' encontrada por um outro caracter...



```

#include <stdio.h>
#define TAMANHO 50
#define OUTRO ' '

/* Retira todas as letras 'c' de uma frase e substitui por um espaço em
branco */
void main()
{
    char original[TAMANHO];
    int i;
    printf("\n\nDigite uma frase: ");
    gets(original);
    printf("\n\nFrase digitada:\n%s", original);
    for (i=0; original[i] != '\0'; i++)
    {
        if ((original[i] == 'c')||(original[i] == 'C')) /* Se for letra
        'c' */
            original[i] = OUTRO; /* Substitui por outro
            caracter..*/
    }
    printf("\n\nFrase alterada:\n%s\n", original);
}

```

```

#include <stdio.h>
#define TAMANHO 50
/* RETIRA todas as letras 'c' de uma string */
main()
{
    char original[TAMANHO];
    int i, j;
    printf("\n\nDigite uma frase: ");
    gets(original);
    printf("\n\nFrase digitada:\n%s", original);
    for (i=0; original[i] != '\0'; i++)
    {
        if ((original[i] == 'c')||(original[i] == 'C'))
        {
            for (j=i; original[j]!='\0'; j++)
                original[j] = original[j+1];
            i--;
        }
    }
    printf("\n\nFrase alterada:\n%s\n", original);
}

```

<p>Uma outra alternativa é a de realmente RETIRAR todas as letras 'c' do conjunto..</p>
---

d) Diga o resultado das variáveis x, y e z depois da seguinte sequência de operações:

```
int x,y,z;  
x=y=10;  
z=++x;  
x=-x;  
y++;  
x=x+y-(z--);
```

Comentários:

- Em `z = ++x`, x é incrementado e então o seu valor é atribuído a z
- Em `x = x+y-(z--)`, z é atribuído a x e então acontece o decremento

Solução:

```
x = -11  
y = 11  
z = 10
```

e) Faça um programa que inverta uma string (uma sequência de caracteres). Use o comando for para varrer a string até o seu final. Lembre-se que a string sempre termina com o caracter '\0'.

Solução:

Neste exercício existem alguns detalhes interessantes. Muitas pessoas não conseguiram chegar ao resultado por causa de um pequeno detalhe na 'linha em destaque'. Lembre-se sempre que se você tem um string de 10 posições, o seu índice pode variar de 0 a 9, e o '\0' vai estar na posição 9. Logo, você deve inverter a posição 8 com a 0, 7 com a 1, e assim por diante. É por isto que a expressão fica:

```
copia[i] = str[strlen(str)-i-1]
```

Não podemos também deixar de colocar o '\0' no final da nova string.

```
#include <stdio.h>  
#include <string.h>  
#define MAX 15  
  
void main()  
{  
    char str[MAX], copia[MAX];  
    int i;  
    printf("\n\nEntre com uma palavra: ");  
    gets(str);  
    for(i=0; str[i] != '\0'; i++)  
    {  
        copia[i] = str[strlen(str)-i-1];    /* Linha em destaque */  
    }  
    copia[i] = '\0';  
    printf("\n\nString Invertida: %s\n", copia);  
}
```

- f) Faça um programa que leia várias palavras pelo teclado, e armazene cada palavra em uma string. Depois, concatene todas as strings lidas numa única string. Por fim apresente esta como resultado ao final do programa.

**Solução:**

Aí segue o programa segundo o enunciado requiere.

```
/* Le 5 strings: -----*/
#include <stdio.h>
#include <string.h>
#define TAM 15
#define MAX 5
void main()
{
    char palavra[TAM], frase[TAM*MAX];
    int i;
    /* Inicializa a string frase */
    frase[0] = '\\0';

    printf("\\n\\n Digite 5 palavras, teclando <ENTER> ao final de cada
uma:\\n");

    /* Le todas as strings */
    for (i=0; i<MAX; i++)
    {
        gets(palavra);
        strcat(frase, palavra);
    }

    /* Apresenta o resultado */
    printf("\\n\\n Resultado: %s\\n",frase);
}
```

- g) O que o programa a seguir faz? Qual é o resultado obtido se a string fornecida for:

a) "Ah! Eu to maluco!"

b) "5\*4+(3^3)+4\*5"

```
#include <stdio.h>
#include <string.h>
#define TAM 20
void main ()
{
    char s[TAM];
    int c, i, j;
    for (i=0, j=strlen(s)-1; i<j; i++, j--)
    {
        c = s[i];
        s[i] = s[j];
        s[j] = c;
    }
}
```

Solução:

A função inverte a string fornecida. Assim, obtém-se:

a) "Ah! Eu to maluco!" => "!oculam ot uE !hA"  
b) "5\*4+(3^3)+4\*5" => "5\*4+)3^3(+4\*5"

Comentários:

- a função começa com i no início e j no final da string. A medida que i incrementa e j decrementa, o conteúdo das posições i e j é trocado, resultando na string invertida.

- repare que a função é válida para strings de comprimento par ou ímpar: Se for par, existirão 2 elementos medianos, que serão trocados; se for ímpar, ao atingir o elemento mediano, a string já estará toda invertida.

- todo string é terminado com \0 (caracter nulo), que não é contado pela função strlen(). Se um string s possui n caracteres (descontando o \0), strlen(s) = n, e existirão elementos no string de s[0] ate s[n-1].

h) Aprendemos, pelo curso, que o valor de uma variável ou expressão do tipo vetor é o endereço do elemento zero do vetor. Seja a[] um vetor qualquer, independente de tipo e tamanho, e pa um ponteiro para o mesmo tipo de a[]. Responda V ou F, justificando as que forem falsas:

( ) Qualquer expressão de vetor e índice é equivalente a uma outra expressão escrita com um apontador e um deslocamento;

( ) Após a atribuição pa=&a[0]; pa e a possuem valores idênticos;

( ) A atribuição pa=&a[0]; pode ser escrita como pa=a;

( ) Uma referencia a a[i] pode ser escrita como \*(a+i)

( ) &a[i] e a+i são idênticos

( ) a+i e' o endereço do i-ésimo elemento após a

( ) pa[i] e' idêntico a \*(pa+i)

( ) pa=a e' uma operação valida

( ) pa++ e' uma operação valida

( ) a=pa e' uma operação valida

( ) a++ e' uma operação valida

RESPOSTAS COMENTADAS:

(V) Qualquer expressão de vetor e índice é equivalente a uma outra expressão escrita com um apontador e um deslocamento

(V) Após a atribuição pa=&a[0]; pa e a possuem valores idênticos

(V) A atribuição pa=&a[0]; pode ser escrita como pa=a;

(V) Uma referência a a[i] pode ser escrita como \*(a+i)

(V) &a[i] e a+i são idênticos

(V) a+i é o endereço do i-ésimo elemento após a

(V) pa[i] é idêntico a \*(pa+i)

(V) pa=a é uma operação válida

(V) pa++ é uma operação válida

(F) a=pa é uma operação válida= o nome de um vetor não pode ser usado como uma variável

(F) a++ é uma operação válida= o nome de um vetor não pode ser usado como uma variável

## Comentários:

- Se `pa` aponta para um elemento particular de um vetor, então, por definição, `pa+1` aponta para o próximo elemento, `pa+i` aponta para  $i$  elementos após `pa` e `pa-i` aponta para  $i$  elementos antes de `pa`. Assim, se `pa` aponta para `a[0]`, `*(pa+1)` refere-se ao conteúdo de `a[1]`, `pa+i` é o endereço de `a[i]` e `*(pa+i)` é o conteúdo de `a[i]`.

- Estas observações aplicam-se independentemente do tipo ou tamanho das variáveis no vetor `a`.

- A correspondência entre indexação e aritmética com ponteiros é evidentemente muito estreita. Por definição, o valor de uma variável ou expressão do tipo vetor é o endereço do elemento zero do vetor. Assim, após a atribuição `pa=&a[0]`; `pa` e `a` possuem valores idênticos. Como o nome de um vetor é sinônimo para o local do elemento inicial, a atribuição `pa=&a[0]` também pode ser escrita como `pa=a`;

- `a[i]` pode ser escrita como `*(a+i)`. Na avaliação de `a[i]`, o C a converte para `*(a+i)` imediatamente; as duas formas são equivalentes. Aplicando-se o operador `&` a ambas as partes dessa equivalência, segue-se que `&a[i]` e `a+i` também são idênticos: `a+i` é o endereço do  $i$ -ésimo elemento após `a`. Por outro lado, se `pa` é um apontador, expressões podem usá-lo como um subscripto; `pa[i]` é idêntico a `*(pa+i)`. Em suma, qualquer expressão de vetor e índice é equivalente a uma escrita com um apontador e um deslocamento.

- Existe uma diferença importante entre o nome de um vetor e um apontador que deve ser sempre lembrada: um apontador é uma variável, de forma que `pa=a` e `pa++` são operações válidas. Mas o nome de um vetor não é uma variável (poderia ser visto como um ponteiro constante, mas não uma variável); construções como `a=pa` e `a++` são ilegais.

i) O programa abaixo está errado. Por quê?

```
void main(void) /* esse programa esta errado */
{
    int x, *p;
    x = 10;
    p = x;
    printf ("%d", *p);
}
```

### Solução comentada:

- O erro aqui apresentado é provocado por um simples equívoco sobre como usar um ponteiro. A chamada de `printf()` não imprime o valor de `x`, que é 10. Imprime um valor desconhecido porque a atribuição `p=x`; está errada. Esse comando atribui o valor 10 ao ponteiro `p`, que se supõe conter um endereço, não um valor. Para corrigí-lo, basta escrever `p=&x`;