



Lista de exercícios 8 - Ponteiros

1) Qual das instruções abaixo é correta para declarar um ponteiro para inteiro?

- a) `*int pti;` c) `&pti;` e) `int *pti;`
b) `* pti;` d) `int _pti;`

2) Seja a seguinte sequência de instruções em um programa C:

```
int *pti;  
int i = 10;  
pti = &i;
```

Qual afirmativa é *falsa*?

- a) `pti` armazena o endereço de `i`.
b) `*pti` é igual a 10.
c) ao se executar `*pti = 20;` `i` passará a ter o valor 20.
d) ao se alterar o valor de `i`, `*pti` será modificado.
e) `pti` é igual a 10.

3) Escreva uma função que declara variáveis do tipo inteiro, real e caractere e ponteiros para esses tipos. Associe os ponteiros às variáveis e altere seus valores através dos ponteiros. Por fim, imprima os valores das variáveis.

4) Responda as perguntas abaixo. Se achar necessário, faça testes em algum programa em C.

- a) Se a variável `p` é um ponteiro para inteiro. Explique a diferença entre `p++` / `(*p)++` / `*(p++)`.
b) O que quer dizer `*(p+10)`?

5) Qual o valor de `y` no final do programa? Tente primeiro descobrir e depois verifique no computador o resultado. A seguir, escreva um comentário em cada comando de atribuição explicando o que ele fez e o valor da variável à esquerda do `=` após sua execução.

```
#include <stdio.h>
```

```
int main(){  
    int y, *p, x;  
    y = 0;  
    p = &y;  
    x = 4;  
    (*p)++;  
    x--;  
    (*p) += x;  
    printf("y = %d\n", y);  
    return 0;  
}
```

```
#include <stdio.h>
```

```
int main(){
    int vec[] = {4,9,13};
    int i;
    for(i = 0 ; i < 3 ; i++)
        printf("%d,",*(vet+i));
    return 0;
}
```

```
01. #include <stdio.h>
02.
03. int main(){
04.     int a, b, *p1, *p2;
05.     a = 4;
06.     b = 3;
07.     p1 = &a;
08.     p2 = p1;
09.     *p2 = *p1 + 3;
10.     b = b * (*p1);
11.     (*p2)++;
12.     p1 = &b;
13.     printf("%d %d\n",*p1,*p2);
14.     printf("%d %d\n",a,b);
15. }
```

[illegible]

```
#include <stdio.h>

void func(int *a, int *b){
    int c = *a;
    *a = *b;
    *b = c;
}

int main(){
    int x, y;
    x = 4;
    y = 3;
    func(&x,&y);
    printf("x=%d, y=%d\n",x,y);
}
```

[illegible]

9) Mostre na tabela abaixo todos os passos (teste de mesa) e identifique a saída do programa em C.

```
#include <stdio.h>
```

```
int main(){
    int vet[4], i, *p, *q;
    for(i = 0, i < 4 , i++)
        vet[i] = 3*i+1;
    p = vet;
    p++;
    q = p+2;
    *p = (*p) + 1;
    *q = (*q) - 1;
    for(i = 0 ; i < 4 ; i++)
        printf("%d, ",vet[i]);
}
```

Valores das variáveis					
vet[0]	vet[1]	vet[2]	vet[3]	p	q

10) Crie um programa que contenha um vetor de inteiros contendo 5 elementos. Utilizando apenas aritmética de ponteiros, leia esse vetor do teclado e imprima o dobro de cada valor lido.

11) Escreva uma função com a seguinte assinatura:

```
void limites(int vet[], int n, int *min, int *max);
```

onde **vet** é um vetor de inteiros e **n** indica a quantidade de elementos nesse vetor. A função deve armazenar nas variáveis passadas por referência, **min** e **max**, o valor mínimo e máximo do vetor, respectivamente. Escreva também uma função **main** que use essa função.

12) Faça um programa que leia três valores inteiros e chame uma função que receba esses três valores de entrada e retorne eles ordenados, ou seja, o menor valor na primeira variável, o segundo menor valor na segunda variável do meio, e o maior valor na última variável. A função deve retornar 1 se os três valores forem iguais, 2 se dois valores forem iguais e 3 se todos os valores forem diferentes. Exibir os valores ordenados na tela.

13) Implemente uma função que calcule as raízes de uma equação do segundo grau do tipo $Ax^2+Bx+C=0$. Lembrando que:

$$X = \frac{-B \pm \sqrt{\Delta}}{2A}$$

Onde

$$\Delta = B^2 - 4AC$$

A variável A tem que ser diferente de zero.

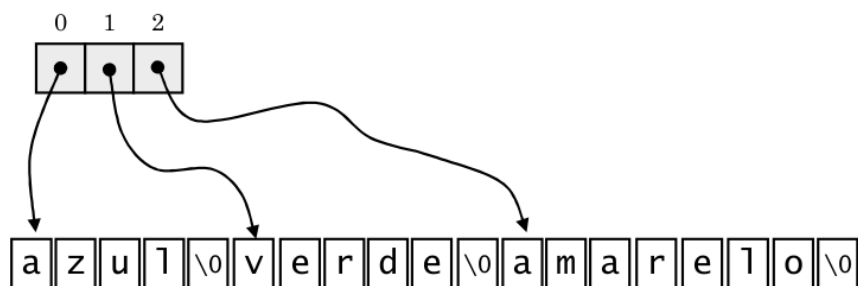
- Se $\Delta < 0$, não existe raiz real.
- Se $\Delta = 0$, existe uma raiz real.
- Se $\Delta \geq 0$, existem duas raízes reais.

Essa função deve obedecer à seguinte assinatura:

```
int raizes(float A, float B, float C, float *X1, float *X2);
```

Essa função deve ter como valor de retorno o número de raízes reais e distintas da equação. Se existirem raízes reais, seus valores devem ser armazenados nas variáveis apontadas por X1 e X2.

- 14) Usando ponteiros, codifique a função **void puts(char *s)**, que exibe uma string **s** na tela e, depois, muda o cursor de linha. Imprima um caractere por vez usando a função **putc(c)**.
- 15) Usando ponteiros, codifique a função **int strlen(char *s)**, que retorna a quantidade de caracteres de uma string.
- 16) Usando ponteiros, codifique a função **void strcpy(char *origem, char *destino)**, que copia os valores da string passada para **origem** para a string passada para **destino**.
- 17) Usando as funções das questões 15 e 16, crie um programa que possui um vetor de ponteiros para caractere (**char *nomes[5]**), leia do teclado strings dadas pelo usuário, copiando elas para strings de tamanho exato às strings que o usuário entrou e amarrando-as aos ponteiros do vetor **nomes**.



- 18) Construa um registro aluno com nome, número de matrícula e curso. Construa um vetor alocado dinamicamente para armazenar os registros de N alunos. Esse valor N deve ser pedido do usuário.