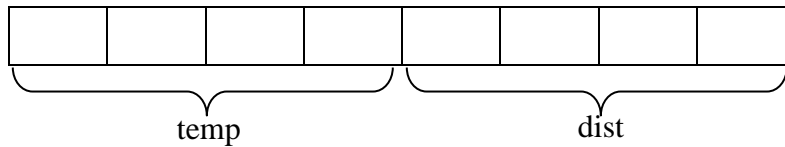


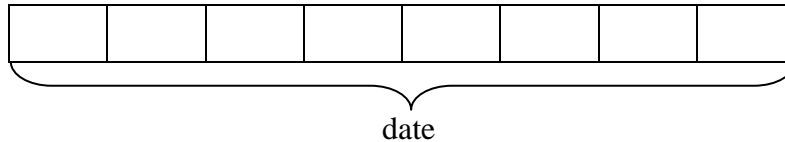
Lista de Exercícios de Programação Estruturada - Ponteiros

1. Se uma variável for declarada como sendo um ponteiro, o que deve ser armazenado na variável?
2. Se **soma** é uma variável, o que significa **&soma**?
3. Considerando as variáveis e os endereços ilustrados na figura abaixo, determine quem é **&temp**, **&dist**, **&date** e **&miles**.

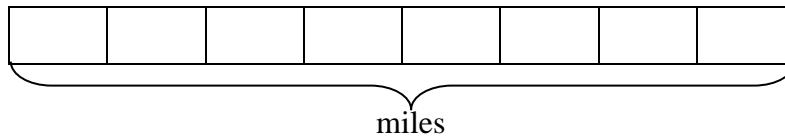
Endereços: 16892 16893 16894 16895 16896 16897 16898 16899



Endereços: 16900 16901 16902 16903 16904 16905 16906 16907



Endereços: 16908 16909 16910 16911 16912 16913 16914 16915



4. Implemente em C um programa incluindo as declarações abaixo. O programa deve exibir os endereços respectivos de cada variável.

```
char key, choice;  
int num, count;  
float yield, price;
```

5. Para as variáveis e os endereços ilustrados na figura, preencha os dados conforme determinados pelos comandos abaixo:

- a. `pt_num = &m;`
- b. `amt_addr = &amt;`
- c. `*z_addr = 25;`
- d. `k = *num_addr;`
- e. `pt_day = z_addr;`
- f. `*pt_yr = 1987;`
- g. `*amt_addr = *num_addr;`

Variável: pt_num
Endereço: 500

Variável: amt_addr
Endereço: 564

Variável: z_addr
Endereço: 8024

Variável: num_addr
Endereço: 10132

Variável: pt_day
Endereço: 14862

Variável: pt_yr
Endereço: 15010

Variável: years
Endereço: 694

Variável: m
Endereço: 8096

Variável: amt
Endereço: 16256

Variável: firstnum
Endereço: 18938

Variável: balz
Endereço: 20492

Variável: k
Endereço: 24608

6. O código abaixo está correto? Justifique sua resposta.

```
void troca (int *x, int *y) {  
    int *temp;  
    *temp = *x; *x = *y; *y = *temp;  
}
```

7. Um ponteiro pode ser usado para dizer a um procedimento onde ele deve depositar o resultado de seus cálculos. Implemente em C um procedimento `xxx` que converta minutos em horas-e-minutos. O procedimento recebe um inteiro `mnts` e os endereços de duas variáveis inteiras, digamos `hora` e `minutos`, e atribui valores a essas variáveis de modo que `minutos` seja menor que 60 e que $60 \cdot \text{hora} + \text{minutos}$ seja igual a `mnts`. Implemente também uma função `main` que use o procedimento `xxx`.

8. As raízes de uma equação do segundo grau, da forma $ax^2 + bx + c = 0$, podem ser calculadas através da expressão $x = \frac{-b \pm \sqrt{\Delta}}{2a}$, onde $\Delta = b^2 - 4ac$. Implemente um programa em C que solicite ao usuário os valores dos coeficientes **a**, **b** e **c** e chame um **procedimento** que calcule as raízes reais da equação. O procedimento deverá receber como parâmetros de entrada os coeficientes **a**, **b** e **c** (passagem por valor) e como parâmetros de saída as variáveis **existe**, **x1** e **x2** (passagem por referência). Se existirem raízes reais, a variável **existe** deve receber o valor 1, caso contrário, ela deve receber o valor 0. Caso existam raízes reais, o programa principal (main) deverá se encarregar de exibi-las na tela. Caso contrário, o programa (main) exibirá a mensagem “*Não existem raízes reais*”.

Notas:

- a) Se $\Delta > 0$, x1 e x2 conterão duas raízes reais e distintas;
- b) Se $\Delta = 0$, x1 e x2 conterão duas raízes reais e iguais;
- c) Se $\Delta < 0$, x1 e x2 não deverão ser alteradas pelo procedimento.

9. Implemente um programa em C que solicite ao usuário 3 números reais e chame um **procedimento** que receba, como parâmetros passados por referência, as três variáveis que armazenam os valores informados pelo usuário. O procedimento deverá ordenar os valores de forma que o menor entre esses três valores fique na primeira variável, o segundo valor fique na segunda variável e o maior fique na terceira variável. O programa principal (main) deve exibir os valores ordenados.

Exemplo:

Antes da ordenação:

A = 6

B = 3

C = 1

Depois da ordenação:

A = 1

B = 3

C = 6