



**1) (3,0 PONTOS)** Fazer um programa que leia o nome de um arquivo em uma string e separe em duas novas strings:

- na primeira string o diretório ou pasta (tudo o que estiver até a última barra '/'): caso não houver nenhuma barra, retorne "./"
- na segunda string o nome (tudo o que estiver após a última barra até o final: caso não houver nenhuma barra, considerar tudo como nome do arquivo).

**Exemplos:**

| Nome Completo           | Diretório        | Arquivo     |
|-------------------------|------------------|-------------|
| /usr/local/meus/teste.c | /usr/local/meus/ | teste.c     |
| /usr/local/meus/teste   | /usr/local/meus/ | teste       |
| /arq.doc.tmp            | /                | arq.doc.tmp |
| /tmp/arq 3              | /tmp/            | arq 3       |
| recados.txt             | ./               | recados.txt |

**2) (3,5 PONTOS)** Faça um programa que leia uma matriz de 5 linhas x 5 colunas. Troque as linhas desta matriz até que na primeira linha fique a linha que contém maior quantidade de números zeros, na segunda linha a que contém a segunda maior quantidade de números zeros, ... , e finalmente na última a linha que contém a menor quantidade de números zeros. Por fim, escreva a matriz resultante.

Por exemplo, dada a matriz

$$\begin{bmatrix} 0.0 & 0.0 & 2.3 & 3.5 & 1.5 \\ 4.9 & 0.0 & 0.0 & 0.0 & 3.6 \\ 2.9 & 0.0 & 8.5 & 6.4 & 4.5 \\ 4.3 & 0.0 & 0.0 & 0.0 & 0.0 \\ 3.4 & 7.3 & 2.8 & 9.5 & 9.2 \end{bmatrix}$$

o programa deve produzir a seguinte matriz (a linha 3 possui o maior número de valores zero em sua linha, depois a linha 1, linha 0 e linha 2, nesta ordem)

$$\begin{bmatrix} 4.3 & 0.0 & 0.0 & 0.0 & 0.0 \\ 4.9 & 0.0 & 0.0 & 0.0 & 3.6 \\ 0.0 & 0.0 & 2.3 & 3.5 & 1.5 \\ 2.9 & 0.0 & 8.5 & 6.4 & 4.5 \\ 3.4 & 7.3 & 2.8 & 9.5 & 9.2 \end{bmatrix}$$

**3) (3,5 PONTOS)** Faça um programa que leia uma string e calcule a operação, escrevendo o resultado. As operações possíveis são: '+', '-', '\*' e '/'. Para facilitar, considere que a) não existem espaços em branco na string; b) na string existe somente uma operação; e c) sempre existem dois números inteiros na string. A seguir, são mostrados alguns exemplo do que o usuário digita e o que o programa deve escrever

| Leitura | Escrita |
|---------|---------|
| 10+16   | 26      |
| 1000-54 | 946     |
| 12*4    | 48      |
| 50/8    | 6,25    |