

Nomes: _____

1. Escreva uma função que lê uma sequência de números e conte o número de elementos maior que a média.
2. Escreva um programa que leia uma lista de nomes separados por espaço e imprima esses nomes em ordem alfabética. Implemente duas versões mais eficientes para os casos:
 - em que se presume que a quantidade de nomes é no máximo 10000;
 - que não se faz hipóteses sobre a quantidade de nomes.
3. Descreva as diferenças de implementação e uso de memória dos programas do item anterior.
4. Um cadastro de frutas tem as seguintes informações: *nome e peso médio da uma unidade em gramas*. Cada fruta tem também uma lista de nutrientes. Um nutriente é composto pelo nome e um percentual; esse percentual representa a fração do total diário recomendado para uma pessoa que é suprida por uma grama de fruta.
 - (a) Escreva um programa que leia os dados do cadastro e armazene-os na memória.
 - (b) Escreva uma função que, dado o nome de um nutriente, encontre a fruta cadastrada que, com o menor número de unidades, poderia suprir completamente a necessidade diária desse nutriente.

Matriz dinâmica

Uma maneira de alocar uma matriz $m \times n$ de inteiros dinamicamente é:

```
int **mat;
int i;
// crie um vetor de m vetores
mat = malloc(sizeof(*int) * m);
// crie m vetores de n inteiros
for (i = 0; i < m; i++)
    mat[i] = malloc(sizeof(int) * n);
```

Outra maneira é criar uma matriz *linearizada*:

```
int *mat;
// crie um vetor de m*n inteiros}
mat = malloc(sizeof(int) * m * n);
```

5. Escreva uma função que recebe uma matriz linearizada que representa uma imagem em preto e branco (0 ou 1) e “desenhe” duas faixas de 20 pixels no topo e na base.
6. Suponha que você precise ler uma sequência de vetores n -dimensionais e ordená-los pelo módulo. Qual representação mais adequada? Explique por que e implemente.