

Nomes: _____

1. Escreva uma função que lê uma matriz $m \times n$ do teclado alocada dinamicamente como vetor de vetores.
2. Escreva uma função que receba dois vetores de mesmo tamanho e devolva 0 se são iguais, -1 se o primeiro vier antes lexicograficamente, e +1 caso contrário.
3. Descreva uma estrutura de dados que representa um polinômio de grau arbitrário.
4. Escreva uma função recursiva que calcula o valor do polinômio $p(x)$ para um dado x .
5. Implemente o método da bissecção recursivamente para encontrar a raiz de um polinômio.
6. Descreva uma estrutura que representa um número que, ou é a única raiz de um polinômio entre um intervalo real $[a, b]$, ou é a uma função transcendental sen, tan ou cos de um certo número real.
7. Descreva o que a função a seguir faz. Desenhe a pilha de chamadas.

```
int calcula(int a, int b) {
    if (b == 1)
        return a;
    else
        return a * calcula(a, b - 1)
}
```

8. Um sinal sonoro digital nada mais é do que uma sequência de amostras no intervalo $[-1, 0, 1, 0]$ (ou outro intervalo, como na imagem) de intensidades de uma onda sonora. Essas amostras são proporcionais à amplitude da onda em determinado instante. Assim, um “som” de 44,1 kHz como um CD tem 44.100 amostras em cada segundo e tem qualidade muito melhor do algumas ligações telefônicas com 8 kHz.
 - (a) Um som estéreo nada mais é do que dois canais de som (esquerda e direita). Descreva uma estrutura de dados para representar um som estéreo.
 - (b) Escreva uma função para ler do teclado um som estéreo (as amostras são apresentadas de forma alternada, esquerda, direita, esquerda, direita, ...).
 - (c) Pelo princípio de interposição das ondas, quando duas pessoas falam ao mesmo tempo, as amplitudes se somam. Pensando nisso, escreva uma função que faça a mixagem (mistura) de dois “sons” estéreos.
 - (d) Fisicamente, os auto-falantes têm limitações de pressão, por isso as amostras são limitadas. Assim, se tivermos uma sequência 0,7; 0,8; 1,0; 1,2; 1,3; 1,2; 1,1; 1,0; 0,8, o auto-falante tocará como se fosse 0,7; 0,8; 1,0; 1,0; 1,0; 1,0; 1,0; 0,8. Isso explica as distorções quando aumentamos o volume. Uma maneira (bem rudimentar) de corrigir isso é normalizando o sinal, isso é, dividindo cada valor pelo maior valor encontrado na sequência. Implemente uma função que normalize um som.

