

Prova 2
MAC-115 – Introdução à Computação
1º semestre de 2017 - IAG

nome (em letras de forma **LEGÍVEIS**):

assinatura:

professor:

No. USP:

Instruções

1. Não destaque as folhas deste caderno.
2. A prova pode ser feita a lápis.
3. A legibilidade também faz parte da nota !
4. A prova consta de 3 questões. Verifique antes de começar a prova se o seu caderno de questões está completo.
5. Não é permitido o uso de folhas avulsas para rascunho.
6. Não é necessário apagar rascunhos no caderno de questão mas especifique qual é a resposta e qual é o rascunho.
7. Só é permitido usar os recursos dados nas aulas até o dia desta prova e deve-se seguir todas as restrições dadas também.
8. A prova é sem consulta.

Não escrever nesta parte da folha

Questão	Nota	Observação
1		
2		
3		
Total		

Boa Sorte !

Questão 1 (valor=3.0)

O **método de Horner** permite o cálculo eficiente de valores de funções polinomiais da forma

$$p(x) = \sum_{k=0}^N a_k x^k$$

explorando a identidade

$$p(x) = a_0 + x(a_1 + x(a_2 + x(\cdots + x(a_{N-1} + xa_N)))) .$$

Especificamente, esse algoritmo calcula os valores das expressões acima entre parênteses, de dentro para fora,

$$\begin{aligned} b_N &= a_N \\ b_{N-1} &= a_{N-1} + b_N x \\ \vdots &\quad \quad \quad \vdots \\ b_0 &= a_0 + b_1 x \end{aligned}$$

de tal forma que $p(x) = b_0$.

a) Escreva uma função **MetodoHorner** em Python que recebe uma lista de coeficientes **a** e um valor de **x**, e devolve o valor de $p(x)$ calculado pelo método de Horner.

b) Considere *dada* uma função **CalculaPolinomio** com a mesma descrição da função **MetodoHorner**, mas que calcula a expressão $p(x) = \sum_{k=0}^N a_k x^k$ fazendo uma soma simples dos termos $a_k * (x^{**}k)$. Escreva uma função **TestaHorner** que recebe um inteiro M e testa a função **MetodoHorner** com $M + 1$ entradas aleatórias (a, x) . Use `import random` no início do código e `random.random()` para gerar cada valor aleatório. As $M + 1$ entradas aleatórias geradas devem corresponder respectivamente a polinômios de graus $0, 1, 2, \dots, M$. Sua função deve produzir um erro de execução (usando `assert`) caso o valor absoluto da diferença entre os resultados produzidos pelas duas implementações seja maior que 10^{-6} . **Atenção:** não implemente a função **CalculaPolinomio**! Você pode fazer o item (b) mesmo sem ter feito o item (a).

Questão 2 (valor=4.0)

Implemente uma classe para representar figuras através de matrizes de caracteres (strings de tamanho 1) usando ' ' (espaço) e 'X'. Sua classe deve conter os seguintes métodos:

- a) um construtor (`__init__`) que recebe as dimensões da matriz e inicializa a figura com espaços.
- b) um método `imprime()` que mostra a figura na tela.
- c) um método `muda(i,j)` que troca o caractere da posição `[i][j]` de ' ' para 'X' e vice-versa. Atenção: seu método deve verificar se os valores de `i` e `j` estão na faixa correta, e ignorar a chamada se esta condição não se verificar.
- d) um método `inverte()` que troca todos os ' ' por 'X' e vice-versa.
- e) um método `espelha(eixo)` que espelha a figura na horizontal se (`eixo=='H'`) e na vertical se `eixo=='V'` (a mensagem não deve fazer nada se o valor de `eixo` for outro).

Exemplo: figura original e sequência obtida após `espelha('V')`, `espelha('H')` e `inverte()`:

		X	X	X	X			
		X	X			X	X	
	X	X					X	X
	X	X					X	X
	X	X					X	X
						X	X	
					X	X		
				X	X			
				X	X			
						X	X	
						X	X	

		X	X	X	X			
		X	X			X	X	
	X	X					X	X
	X	X					X	X
	X	X					X	X
		X	X					
			X	X				
				X	X			
					X	X		
					X	X		

			XX		
			XX		
			XX		
			XX		
			XX		
		XX			
XX					XX
XX					XX
XX					XX
	XX			XX	
		XXXX			

[illegible]

Questão 3 (valor=3.0)

Escreva uma função que recebe uma matriz de números inteiros e devolve dois valores, o número de linhas identicamente nulas e o número de colunas identicamente nulas. Como exemplo, a matriz abaixo

$$\begin{bmatrix} 1 & 0 & 2 & 3 \\ 4 & 0 & 5 & 6 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

possui 2 linhas nulas e 1 coluna nula.

