

MAC0110 Introdução à Computação

BACHARELADO EM ESTATÍSTICA, MATEMÁTICA E MATEMÁTICA APLICADA
Segunda Prova – 18 de maio de 2017

Nome: _____

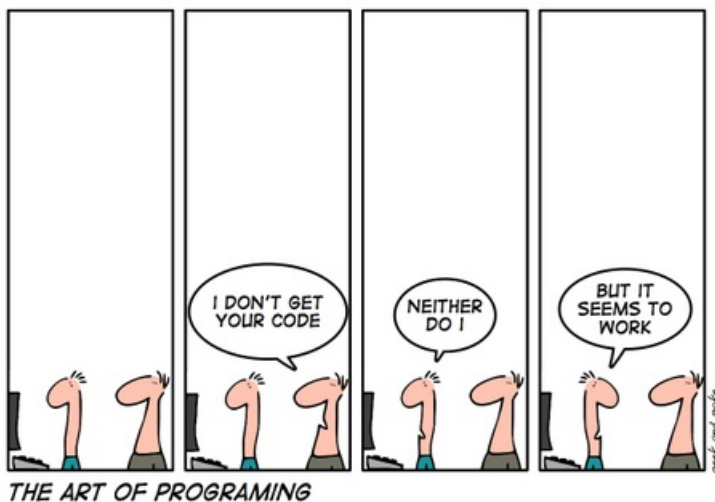
Assinatura: _____

Nº USP: _____

Instruções:

1. Não destaque as folhas deste caderno. A prova pode ser feita a lápis.
2. A prova consta de 2 questões cada uma com 2 itens. Verifique antes de começar a prova se o seu caderno está completo.
3. As questões podem ser resolvidas em qualquer página. Ao escrever uma solução (ou parte dela) em página diferente do enunciado, escreva QUESTÃO X em letras ENORMES junto da solução.
4. As soluções devem ser em Python. Você pode usar apenas recursos de Python vistos em aula. Cuidado com a legibilidade e, principalmente, com a TABULAÇÃO.
5. As soluções não precisam verificar consistência de dados.
6. Não é permitido o uso de folhas avulsas para rascunho, a consulta a livros, apontamentos, colegas ou equipamentos eletrônicos. Desligue o seu celular e qualquer equipamento que possa perturbar o andamento da prova.

DURAÇÃO DA PROVA: 100 minutos



Questão	Valor	Nota
1(a)	2,5	
1(b)	2,5	
2(a)	2,0	
2(b)	3,0	
Total	10,0	

Questão 1 (vale 5,0 pontos)

Item 1(a) (vale 2,5 pontos)

Dado um número real x , uma aproximação da raiz cúbica $\sqrt[3]{x}$ de x pode ser calculada pelo método de Newton através seguinte sequência:

$$r_0 = x,$$
$$r_{n+1} = \frac{1}{3} \left(\frac{x}{r_n^2} + 2r_n \right).$$

Para um número real $\text{eps} > 0$ a **eps-aproximação** de $\sqrt[3]{x}$ é o primeiro valor r_{n+1} tal que $|r_{n+1} - r_n| < \text{eps}$. Por exemplo, a **0.1-aproximação** de $\sqrt[3]{8}$ é 2.00004628653597 pois

```
r0 = 8
r1 = 5.375
r2 = 3.6756354786371013
r3 = 2.6478039787146694
r4 = 2.145564607905976
r5 = 2.0096524096938087
r6 = 2.00004628653597
```

e $|r_6 - r_5| \approx |-0.00960612315783882| < 0.1$. Escreva uma função `raiz_cubica()` que recebe dois números reais x e $\text{eps} > 0$ e calcula e retorna a **eps-aproximação** de $\sqrt[3]{x}$ através do método de Newton.

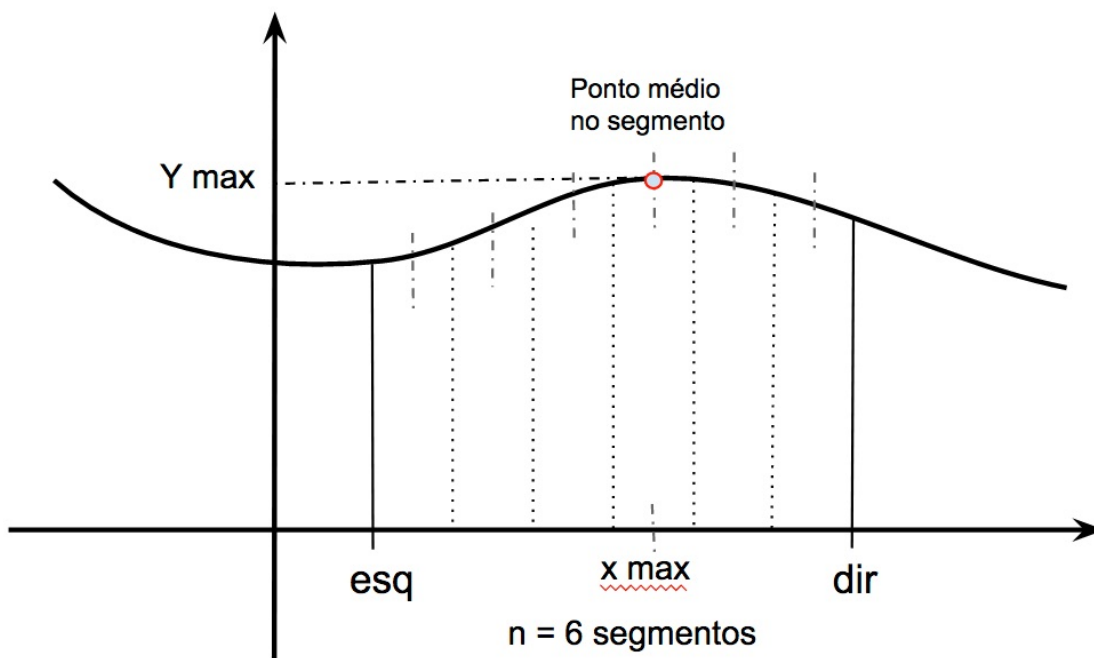
Item 1(b) (vale 2,5 pontos)

Considere a função $y^3 = x^2 + ax + b$ onde a e b são números reais. Escreva um programa (função `main()`) que leia:

- dois números reais a e b ;
- dois números reais `esq` e `dir` que definem o intervalo `[esq, dir]`;
- um inteiro $n > 1$.

O seu programa deve dividir o intervalo `[esq, dir]` em n segmentos, como ilustra a figura a seguir. Para cada segmento, o programa deve calcular o valor de y correspondente ao seu ponto médio x . Finalmente, o programa deve imprimir as coordenadas (x_{\max}, y_{\max}) , onde y_{\max} é o maior dos valores de y calculados e x_{\max} é o correspondente ponto médio x . O ponto (x_{\max}, y_{\max}) é ilustrado na figura a seguir. Em caso de haver mais de um ponto médio x correspondente a y_{\max} , o programa deve imprimir o menor.

O seu programa deve usar **obrigatoriamente** a função `raiz_cubica()` da questão anterior para calcular uma 0.001-aproximação de uma raiz cúbica. Você pode utilizar a função `raiz_cubica()` mesmo que não a tenha feito.



Questão 2 (vale 5,0 pontos)

Item 2(a) (vale 2,0 pontos)

Escreva uma função `conte()` que respeite a seguinte especificação.

```
def conte(k, seq):  
    ''' (int, list) -> int
```

Recebe um inteiro `k` e uma lista `seq` de inteiros
e retorna o número de vezes que `k` aparece na lista `seq`.

Exemplos:

```
>>> conte(-1, [])  
0  
>>> conte(21, [10, 38, 10, -523])  
0  
>>> conte(10, [375, 10, 34, 10, 0, 10])  
3  
'''
```


Item 2(b) (vale 3,0 pontos)

Uma sequência com $n > 0$ números inteiros é **mágica** se o i -ésimo número da sequência, denotado por x_i , é o número de vezes que i aparece na sequência, para $i = 0, \dots, n - 1$.

Exemplos:

A sequência $[2, 1, 2, 0, 0]$ é mágica, pois:

- 0 aparece 2 vezes na sequência e $x_0 = 2$,
- 1 aparece 1 vez na sequência e $x_1 = 1$,
- 2 aparece 2 vezes na sequência e $x_2 = 2$,
- 3 aparece 0 vezes na sequência e $x_3 = 0$, e
- 4 aparece 0 vezes na sequência e $x_4 = 0$.

A sequência $[3, 2, 1, 0, 1, 0, 0]$ não é mágica, pois:

- pois 3 aparece 1 vez na sequência mas $x_3 = 0$

Escreva um programa (função `main()`) que lê um inteiro $n > 0$ e uma sequência com n números inteiros e imprime “A sequência é mágica :-)”, se a sequência dada for mágica, em caso contrário imprime “A sequência não é mágica :- (“,

O seu programa deve usar **obrigatoriamente** a função `conte()` da questão anterior. Você pode utilizar a função `conte()` mesmo que não a tenha feito.

