

Prova 1
MAC0110 – Introdução à Computação
19/04/2018 - BCC

NOME (EM LETRA DE FORMA LEGÍVEL):

ASSINATURA:

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 Prova!

Questão 1 (valor=3.0)

Escreva um programa em Python que lê do teclado valores inteiros $N > 0$, $\text{dist} > 0$ com N dígitos e uma lista de inteiros entre 0 e $N - 1$ terminada por -1, e calcula a probabilidade dessa sequência ser sorteada pela roleta enviesada representada por (N, dist) (como no EP1). Para ajudar a memória, no EP1 a roleta enviesada é representada pelo número de códigos sorteados N e por uma lista de pesos $p_0, p_1, \dots, p_{N-1} \in \{0, \dots, 9\}$ codificados em um inteiro $\text{dist} = p_{N-1} * 10^{N-1} + \dots + p_1 * 10^1 + p_0 * 10^0$, onde os pesos definem a probabilidade de sorteio de cada valor $n \in \{0, \dots, N - 1\}$ através da expressão $\text{prob}(x = n) = \frac{p_n}{S}$ e $S = p_0 + p_1 + \dots + p_{N-1}$. Lembre que a probabilidade de uma sequência de eventos independentes é o produto das probabilidades individuais, ou seja, a probabilidade da roleta sortear a sequência de números x_1, x_2, \dots, x_K será igual a $\prod_{i=1}^k \text{prob}(x = x_i) = \frac{p_{x_i}}{S}$. Por exemplo, se $N = 3$, $\text{dist} = 735$ e a sequência for 0, 2, 2, 1, 1, 1, 0 então $S = 7 + 3 + 5 = 15$ e a probabilidade desse sorteio será

$$\frac{p_0}{S} \frac{p_2}{S} \frac{p_2}{S} \frac{p_1}{S} \frac{p_1}{S} \frac{p_1}{S} \frac{p_0}{S} = \frac{5}{15} \frac{7}{15} \frac{7}{15} \frac{3}{15} \frac{3}{15} \frac{3}{15} \frac{5}{15} = 0.0001935802469135803.$$

*Como no EP1, não está permitido o uso de exponenciais (operador ** do Python ou reimplementações dele). DÚVIDA: SERÁ QUE FAZEMOS ISSO?? TALVEZ A RESPOSTA SEM ISSO JÁ SEJA GRANDE...*

Dicas: (1) Você não precisa ler os pesos p_n e construir o inteiro dist , pois ele será digitado todo de uma vez pelo usuário. (2) Para simplificar o problema, escreva separadamente uma função `def probabilidade(x,N,dist)` para calcular a probabilidade associada ao sorteio de um único valor $x \in \{0, \dots, N - 1\}$.

DÚVIDA: poderíamos tornar a dica (2) obrigatória dividindo a questão em 2 itens (parte a = função probabilidade, parte b = sequência + produtória). Acho que pode simplificar tanto pra eles quanto pra gente...

Questão 2 (valor=3.0)

Escreva uma função em Python que ajude a gerenciar um sistema de elevadores de um prédio conforme a descrição a seguir. O prédio possui 3 elevadores representados por 3 valores inteiros e_1 , e_2 e e_3 ; se $e_n > 0$ isso significa que o elevador n está parado no andar e_n , e se $e_n < 0$ isso significa que ele está se dirigindo ao andar $\text{abs}(e_n)$. Considere que os andares do prédio começam no 1, e que não se sabe a altura do prédio (o sistema será usado em muitos prédios diferentes). Sua função será acionada cada vez que um usuário chamar um elevador no andar X , devendo devolver o número do elevador (1, 2 ou 3) que irá atender aquele usuário, de acordo com o protótipo

```
def chamaelevador(X,e1,e2,e3):
    """ Calcula o elevador que irá atender o usuário no andar X.
    """
    # seu código aqui...
    return elevador # 1, 2 ou 3
```

A política de atendimento deve priorizar o elevador que estiver mais perto, dando preferência para os elevadores que estão parados; apenas se todos os elevadores estiverem em movimento, deve-se dar preferência para aquele vai parar mais perto.

Questão 3 (valor=3.0+1.0)

a) Escreva uma função com protótipo

```
def triplaPitagoreana(N):  
    """ Tenta encontrar a,b,c tais que  
        0<a<b<c, N == a+b+c, e a**2+b**2 == c**2.  
        """  
    # seu código aqui...  
    return encontrei, a, b, c
```

que recebe um inteiro $N > 0$ e procura alguma tripla de valores inteiros (a, b, c) com $0 < a < b < c$, $N = a + b + c$ e $a^2 + b^2 = c^2$. Se isso ocorrer, sua função deve devolver `encontrei=True` e os valores encontrados de a, b e c ; do contrário, ela devolverá `encontrei=False` e $a = b = c = 0$.

b) Escreva um programa que pede ao usuário um valor inteiro $K > 0$ e produz a lista de todos os valores de $N \in \{0, 1, \dots, K\}$ que possuem triplas Pitagoreanas associadas. Dica: você pode resolver esse item mesmo sem ter feito o item (a), basta usar a função.

