UNIVERSIDADE FEDERAL DE VIÇOSA — DEPARTAMENTO DE INFORMÁTICA
INF 100 – Introdução à Programação I — SEMESTRE II / 2016

Prova 3 **10/12/2016** **Solução**

OBSERVAÇÕES:

- As questões podem ser resolvidas a lápis.
- Entende-se por algoritmo refinado completo um algoritmo contendo a representação do código em português, de forma clara, não ambígua, de modo que cada instrução do algoritmo possa ser traduzida em uma instrução da linguagem de programação.
- Para a leitura de dados, basta usar instruções em formato algorítmico como "**`Leia n`**", "**`Leia A, B, C`**" etc. Não precisa se preocupar com as mensagens escritas antes dessas leituras.

A tabela abaixo apresenta a ordem de prioridade dos operadores e comandos Python mais comuns. Essa informação pode ser útil para você na resolução das questões da prova.

Prioridade	Operador(es) e comando =	Exemplo
1	**	x ** 3
2	- (unário)	-x
3	* / // %	x / y
4	+ -	x - y
5	< <= > >= == !=	x < y
6	not	not x > 0
7	and	x < y and x > 0
8	or	x < y or x == 0
9	=	x = 2

Questão 1 **(20%)**

Na UFI (Universidade Federal de Inficentilândia), em uma área de aproximadamente 0,41km^2, existem 5 lagos onde vivem bandos de capivaras. No ano passado, uma epidemia causada por um parasita quase dizimou a população toda dos roedores. Uma equipe de zoólogos e veterinários que monitora e estuda os animais há tempos conseguiu salvar apenas 6 indivíduos após o desastre causado pelo parasita. No entanto, a população reduzida não chegou a ser motivo de preocupações para os estudiosos, uma vez que o ambiente onde as capivaras vivem é bastante favorável, além de não existirem seus predadores naturais dentro do Campus. Desta forma, espera-se que a população dos simpáticos animais rapidamente cresça e se estabilize no patamar anterior, que era a população máxima dos indivíduos que não perturba o equilíbrio do ecossistema da região onde vivem. No caso, cerca de 80 capivaras.

Para fazer as projeções de crescimento da população dos roedores e saber em quanto tempo a população destes será novamente estabilizada, os cientistas analisaram os dados de controle populacional de levantamentos passados e chegaram à conclusão que a taxa de crescimento populacional das capivaras naquele ecossistema pode ser descrita, de forma satisfatória, pela função:

$$C(t) = 0{,}75 \cdot P(t) \cdot \left(1 - \frac{P(t)}{80}\right)$$

onde t é a variável do tempo, medido em meses e $P(t)$ é a função que calcula a população das capivaras, também em função do tempo e da população inicial. Para o caso da população inicial com 6 indivíduos, a função é definida apenas em função do tempo como sendo:

$$P(t) = \frac{480 \cdot e^{0{,}75t}}{80 + 6 \cdot (e^{0{,}75t} - 1)}$$

a) Escreva uma função em Python para calcular $P(t)$, onde t é o número de meses que se passou desde o "salvamento" das capivaras e deve ser informado como parâmetro.
b) Escreva uma função em Python para calcular $C(t)$, onde t é o número de meses que se passou desde o "salvamento" das capivaras e que deve ser informado como parâmetro. A função que implementa $C(t)$ deve utilizar a função que implementa $P(t)$, desenvolvida no item anterior.

OBS: suponha que a função `exp()` da biblioteca *math* já foi adequadamente incluída no programa. Com isso, a função que calcula e^x, em será a função `exp(x)`. Por exemplo, para calcular $e^{2,3}$, você deve usar `exp(2.3)`.

SOLUÇÃO:

```
def P( t ):
    e = exp( 0.75*t )
    return 480 * e / (80 + 6 * (e - 1))

def C( t ):
    Pt = P( t )
    return 0.75 * Pt * (1 - Pt/80)
```

Questão 2 **(30%)**

Após a tragédia que se abateu sobre as capivaras da UFI, descrita na questão anterior, resolveu-se fazer um controle mais rígido da população destes animais na Universidade. Para isto, cada um dos 5 lagos da instituição foi identificado por um número diferente, pertencentes ao intervalo [1,5].

A cada mês, a equipe de pesquisadores que trabalha com os roedores observa cada lago e anota quantos indivíduos "habitam" (isto é, passam a maior parte do dia) cada um deles. A ideia dos pesquisadores é mapear não só a evolução da população dos bichos, mas também monitorar aspectos diferentes como, por exemplo, qual lago é mais utilizado pelas capivaras em determinada época do ano.

Pede-se que se faça um programa em Python ou um algoritmo refinado completo que, a partir das contagens de capivaras informadas pelos pesquisadores para cada lago, calcule o total de indivíduos pertencentes à região, bem como o percentual da população dos animais que está presente em cada lago. A seguir, tem-se um exemplo de tela ilustrando o comportamento do programa.

Obs.:

1. Assuma que o usuário digitará somente valores positivos;
2. **Não** é necessário fazer saída formatada para precisão de 2 casas decimais e com alinhamento elaborado, conforme apresentado no exemplo de tela abaixo. O exemplo traz a saída desta forma apenas para facilitar a leitura e análise por parte do aluno.

Exemplo de tela:

```
Informe o número de capivaras do lago 1: 5
Informe o número de capivaras do lago 2: 12
Informe o número de capivaras do lago 3: 51
Informe o número de capivaras do lago 4: 6
Informe o número de capivaras do lago 5: 1

População total: 75 capivaras
Lago 1:  6.67% dos indivíduos
Lago 2: 16.00% dos indivíduos
Lago 3: 68.00% dos indivíduos
Lago 4:  8.00% dos indivíduos
Lago 5:  1.33% dos indivíduos
```

SOLUÇÃO:

```
import numpy as np

ncap = np.empty( 5, dtype=int )

ntot = 0
for i in range( 0, 5 ):
    Leia ncap[i]
    ntot = ntot + ncap[i]

print( ntot )
for i in range( 0, 5 ):
    print( i+1, ncap[i]/ntot*100 )
```

Questão 3 **(30%)**

Deseja-se implementar um programa que:

1. Leia dois valores inteiros *m* e *n*
2. Crie uma matriz numérica de *m* linhas e *n* colunas;
3. Leia do teclado todos os valores da matriz;
4. Escreva na tela os elementos da 1ª linha da matriz;
5. Escreva na tela os elementos da última coluna da matriz;
6. Escreva na tela o produto de todos os elementos da matriz.

Segue um exemplo de execução do programa:

```
Entre com o número de linhas:  4
Entre com o número de colunas: 3
Entre com os elementos da matriz 4 x 3:
1 2 3
4 5 6
7 8 9
10 11 12

Resultado:
1 2 3
3 6 9 12
479001600
```

Considere que os itens 1 e 2 do programa já estão implementados a seguir. Complete o programa usando código em Python ou usando algoritmo refinado completo que implemente os passos 3 a 6.

Obs:
Para escrever alguma coisa (por exemplo, uma variável ***x***) na tela e continuar na mesma linha, use:

```
print( x, end=' ')          ou:   Escreva x, ...
```

Indica que o próximo comando "Escreva" continuará a escrever na mesma linha.

Para forçar o "pulo de linha", use:

```
print()                     ou:   Pule para a próxima linha
```

```
import numpy as np
m = int( input('Entre com o número de linhas:  '))
n = int( input('Entre com o número de colunas: '))
A = np.empty( (m,n) )
print('Entre com os elementos da matriz ', m, ' x ', n, ':', sep='')

# Início da solução:

produto = 1
Para i = 0 até m-1:
    Para j = 0 até n-1:
        Leia A[i][j]
        Produto = produto * A[i][j]

# 1a linha da matriz:

Para j = 0 até n-1:
    print( A[0][j], end=' ')
print()

# Última coluna da matriz:

Para i = 0 até m-1:
    print( A[i][n-1], end=' ')
print()

print( produto )
```

Questão 4 **(20%)**

Escreva um algoritmo refinado completo ou um programa em Python que:

1. Leia um valor inteiro *n* maior que 0 (zero). O programa não precisa verificar se o valor é válido ou não. Ou seja, ele deve assumir que o usuário irá digitar um valor maior que zero.
2. Calcule e escreva o valor da constante π usando os *n* primeiros termos da soma de Nilakantha:

$$\pi = 3 + \frac{4}{2\times3\times4} - \frac{4}{4\times5\times6} + \frac{4}{6\times7\times8} - \ldots$$

Seguem três exemplos de execução do programa:

```
n = 1
pi( 1 ) = 3
```

```
n = 2
pi( 2 ) = 3.1666666666666665
```

```
n = 50
pi( 50 ) = 3.1415946525910106
```

SOLUÇÃO:

```
n = int( input('n = '))

pi = 3
sinal = 1
t = 2
for i in range( 1, n ):
    pi = pi + sinal*4/(t*(t+1)*(t+2))
    sinal = -sinal
    t = t + 2

print( pi )
```