

MAC2166 – Introdução à Computação

ESCOLA POLITÉCNICA

Segunda Prova – 13 de maio de 2014

Nome: _____

Assinatura: _____

Nº USP: _____ Turma: _____

Professor: _____

Instruções:

1. Não destaque as folhas deste caderno.
2. A prova consta de 3 questões. Verifique antes de começar a prova se o seu caderno de questões está completo.
3. A prova pode ser feita a lápis. Cuidado com a legibilidade e, principalmente, com a TABULAÇÃO.
4. Qualquer questão pode ser resolvida em qualquer página. Se a questão não está na página correspondente ao enunciado basta indicar isto na página e escrever QUESTÃO X em letras ENORMES antes da solução.
5. Não é necessário apagar rascunhos no caderno de questões.
6. Não é permitido o uso de folhas avulsas para rascunho.
7. Não é permitido o uso de equipamentos eletrônicos.
8. Não é permitido a consulta a livros, apontamentos ou colegas.
9. Você pode definir funções e usá-las à vontade.

DURAÇÃO DA PROVA: 2 horas

Questão	Valor	Nota
1	2,5	
2	3,5	
3	4,0	
Total	10,0	

QUESTÃO 1 (vale 2,5 pontos)

Escreva uma função

```
def substitui(str1, str2, coringa):  
    ''' (str,str,str) -> str '''
```

que recebe três sequências de caracteres (strings) não vazias, sendo que a terceira é um caractere coringa. A função deve devolver uma sequência que é uma cópia da primeira sequência, porém substituindo-se as ocorrências dos caracteres da segunda sequência pelo caractere coringa, obedecendo a ordem de ocorrência em ambas as sequências.

Exemplos:

```
Seq 1: Lá vem o pato pato aqui pato acolá  
Seq 2: papa  
Coringa: X  
Saída: Lá vem o XXto XXto aqui pato acolá
```

```
Seq 1: Era uma casa muito engraçada não tinha teto, não tinha nada  
Seq 2: cratera  
Coringa: *  
Saída: Era uma *asa muito eng**çada não *inha t*to, não tinha nada
```

```
Seq 1: 5248942  
Seq 2: 130  
Coringa: X  
Saída: 5248942
```

```
Seq 1: ABCEFG  
Seq 2: ABCDEFGHIJK  
Coringa: ?  
Saída: ???EFG
```


QUESTÃO 2 (vale 3,5 pontos)

Dadas duas sequências x_1, x_2, \dots, x_n e y_1, y_2, \dots, y_n , $n > 0$, de números reais, os respectivos valores médios são dados por

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i \quad \text{e} \quad \bar{y} = \frac{1}{n} \sum_{i=1}^n y_i$$

A diferença $x_i - \bar{x}$ (e, similarmente, $y_i - \bar{y}$), $i = 1, 2, \dots, n$, é denominada **variação**.

O **coeficiente de correlação** entre os valores das sequências x_1, x_2, \dots, x_n e y_1, y_2, \dots, y_n é definido por:

$$r = \frac{\sum_{i=1}^n (x_i - \bar{x}) * (y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2} * \sqrt{\sum_{i=1}^n (y_i - \bar{y})^2}}$$

Dada uma tabela com as notas de um turma, gostaríamos de verificar se há alguma correlação entre as notas de EP e notas de prova. Em estatística, o coeficiente de correlação pode ser utilizado para tal fim. Suponha que a primeira coluna da tabela contém o nome dos alunos e as demais colunas contém as notas de EPs e Provas. Por exemplo, a tabela abaixo é de uma turma com 5 alunos e contém as notas do EP1, P1, EP2 e P2. O traço indica uma nota que não está disponível.

NOME	EP1	P1	EP2	P2
Fulano	5.0	5.0	3.0	-
Beltrano	0.0	10.0	9.0	8.0
Sicrano	10.0	10.0	10.0	10.0
Batman	9.0	9.0	-	8.0
Pinguin	1.0	2.0	3.0	4.0

Para resolver o problema acima, foi escrito o seguinte programa que lê e imprime uma tabela, lê o índice das colunas que correspondem às sequências para as quais deseja-se calcular o coeficiente de correlação, e então calcula e imprime o coeficiente de correlação. A exemplo do EP3, um valor que não está disponível é representado pelo valor `None` e deve ser ignorado nos cálculos.

```
import math

def main():
    tabela = le_tabela()
    imprime_tabela(tabela)
    cx = int(input("Coluna da primeira seq: "))
    cy = int(input("Coluna da segunda seq: "))
    corr = coeficiente_correlacao(tabela, cx, cy)
    if corr is None:
        print("O coeficiente não pode ser calculado")
    else:
        print("A correlação é %f" %corr)
```

Suponha também que as funções abaixo já estão prontas:

```
def le_tabela():
    ''' Lê, de um arquivo cujo nome é fornecido pelo usuário, uma tabela como
        a especificada acima. Devolve a tabela em uma lista de listas, correspondendo
        às linhas da tabela. Um valor não disponível é representado por None. '''

def imprime_tabela(tabela):
    ''' Recebe uma tabela e imprime-a. '''
```

Você deverá completar o programa escrevendo duas funções especificadas nos itens (a) e (b) a seguir.

a) (vale 1,5 pontos) Escreva uma função

```
def adiciona_coluna_variação(tabela, col):
```

que recebe uma tabela (**tabela**) como a definida anteriormente e adiciona uma coluna à tabela. A coluna a ser adicionada corresponde à variação de cada valor da coluna de índice **col**. A variação de um valor **None** é **None**. Para o cálculo da média dos valores na coluna de índice **col**, desconsidere os valores **None**.

b) (vale 2,0 pontos) Escreva uma função

```
def coeficiente_correlacao(tabela, col1, col2):
```

que recebe uma tabela (**tabela**) e os índices **col1** e **col2** de duas colunas da tabela. A função deve:

- acrescentar duas colunas à tabela: uma com as variações de cada valor na coluna de índice **col1** e outra com as variações de cada valor na coluna de índice **col2**,
- imprimir a tabela (usando a função dada), e
- calcular e devolver o coeficiente de correlação entre as sequências de valores nas colunas de índices **col1** e **col2**.

OBS.: No cálculo do coeficiente, os termos que envolvem um valor **None** não devem ser considerados. Caso não seja possível calcular o coeficiente, deve-se devolver **None**. **Use a função do item anterior**, mesmo que você não a tenha feito.

QUESTÃO 3 (vale 4,0 pontos)

a) (vale 1,0 ponto) Escreva uma função

```
def simetrica(M):
```

que recebe uma matriz e devolve **True** caso a matriz seja simétrica em relação à sua coluna central, e devolve **False** em caso contrário. Suponha que a matriz é de inteiros, quadrada e de dimensão ímpar.

Exemplo: A matriz abaixo à esquerda é simétrica em relação à sua coluna central. Já a matriz à direita não é.

8	1	0	1	8
0	2	1	2	0
0	3	2	3	0
0	4	3	4	0
0	5	4	5	0

2	1	0	1	2
3	2	0	2	3
4	3	0	4	3
5	4	0	4	5
1	0	0	0	1

b) (vale 1,5 pontos) Escreva uma função

```
def recorta(M, k, i, j):
```

que recebe uma matriz M de inteiros, um inteiro positivo ímpar k , os índices i (linha) e j (coluna) de uma posição da matriz, e devolve uma matriz que é uma cópia da submatriz de M , quadrada e de dimensão k , centrada na posição $[i][j]$ da matriz. Você pode supor que a submatriz $k \times k$ está inteiramente contida em M .

Exemplo: Supondo que M é a matriz abaixo à esquerda, k é 3, e i e j são 2 e 1, respectivamente, a submatriz a ser devolvida é a que está abaixo à direita:

	2	1	0	1	2		
M =	3	2	0	2	3	3	2
	4	3	0	4	3	0	0
	5	4	0	4	5	5	4

c) (vale 1,5 pontos) Escreva um programa que lê uma matriz M de inteiros, um inteiro positivo ímpar k , e imprime o número de submatrizes $k \times k$ de M que são simétricas em relação à sua coluna central.

Para fazer este item, você pode supor que é dada uma função

```
def le_matriz():
```

que lê e devolve uma matriz, e pode usá-la sem escrevê-la.

Além disso, **use as funções dos itens anteriores**, mesmo que você não as tenha feito.

Exemplo: A matriz 5×4 abaixo (à esquerda) contém duas submatrizes 3×3 simétricas em relação a sua coluna central (as duas abaixo à direita).

	1	1	1	1		1	1	1
	1	0	1	1		1	0	1
M =	0	0	0	0		0	0	0	.		.	0	0	0
	1	0	0	0		0	0	0
	2	2	2	2		2	2	2

