



**Fundação CECIERJ - Vice-Presidência de Educação Superior a Distância**

**Curso de Tecnologia em Sistemas de Computação**

**Disciplina Fundamentos de Programação**

**AP1 2º semestre de 2018**

---

### **IMPORTANTE**

- Serão aceitos apenas soluções escritas na linguagem Python 3.
  - Prova sem consulta e sem uso de qualquer aparato eletrônico.
  - Use caneta para preencher o seu nome e assinar nas folhas de questões e de respostas.
  - Você pode usar lápis para responder as questões.
  - Ao final da prova, devolva as folhas de questões e as de respostas.
  - Todas as respostas devem ser transcritas nas folhas de respostas. As respostas nas folhas de questões não serão corrigidas.
- 

### **1ª Questão (2,0 pontos)**

Faça um programa que leia frases (linhas) da entrada padrão até que uma string vazia seja digitada. Para cada frase lida inverta todas as palavras contidas na frase cuja quantidade de vogais seja ímpar.

#### Exemplo

| Entrada  |
|--|
| Foi bonita a festa junina<br>Antes de viajar ligue pra universidade<br>Telefone da uff 2629-5000 |

| Saída  |
|--|
| Foi atinob a festa aninuj<br><br>Antes ed rajaiv eugil arp universidade<br><br>Telefone ad ffu 2629-5000 |

#### Distribuição de Pontos

Entrada – 0,2 pontos; Processamento – 1,6 pontos; Saída – 0,2 pontos.

## 2ª Questão (3,0 pontos)

Faça um programa que leia as dimensões de uma matriz, isto é: quantidade de linhas e a quantidade de colunas. Considere que as dimensões sempre serão maiores ou iguais a um. Em seguida, leia linha(s) de valor(es) inteiro(s) a ser(em) preenchido(s) na matriz. Mostre a matriz construída. Utilize a formatação de escrita de números inteiros com quatro dígitos e um espaço em branco entre eles. Em seguida, escreva as posições e seus respectivos valores, das células da matriz que possuam valores primos.

Definição: Um número inteiro é primo se e somente se ele for maior que um e for apenas divisível por ele e por um.

Sugestão: Utilize o operador % para verificar o resto inteiro da divisão. Exemplos de uso:  $31\%13 = 5$ ,  $10\%2 = 0$  e  $18\%5 = 3$ .

### Exemplo

| Entrada                                     | Saída  |
|---|--|
| 2 7<br>8 4 17 9 7 10 1<br>997 2 0 -5 1 6 29 | Matriz lida:<br>8 4 17 9 7 10 1<br>997 2 0 -5 1 6 29<br><br>Células da matriz que contêm números primos:<br>Linha = 1, Coluna = 3, Valor = 17<br>Linha = 1, Coluna = 5, Valor = 7<br>Linha = 2, Coluna = 1, Valor = 997<br>Linha = 2, Coluna = 2, Valor = 2<br>Linha = 2, Coluna = 7, Valor = 29 |

### Distribuição de Pontos

Leitura dos dados de entrada e conversão de tipos – 0,5 pontos; Identificação de números primos – 2,0 pontos; Formatação e impressão da saída – 0,5 pontos.

## 3ª Questão (3,0 pontos)

*Bubble Sort* é um algoritmo de ordenação cuja ideia central é percorrer várias vezes o vetor a ser ordenado, sendo que a cada passagem valores adjacentes são comparados e suas posições são trocadas sempre que necessário para manter-se a ordem dos elementos. A consequência natural dessas trocas é que valores maiores são carregados para o fim da sequência, resultando em um vetor ordenado.

Implemente um programa que recebe um vetor de números de ponto flutuante (`float`) desordenado e o ordene utilizando *Bubble Sort*. Especificamente:

- O programa principal deve ativar um subprograma que solicita ao usuário, via entrada padrão, o tamanho do vetor, gera o conteúdo desse vetor aleatoriamente e retorna o vetor gerado. Sugestão: utilize a função `random.random()`, disponível na API do Python. Essa função que retorna um número entre 0.0 e 1.0.
- Em seguida, o programa principal deve ativar um subprograma que escreve na saída padrão o vetor gerado (veja a formação no exemplo).
- Após a escrita, o programa principal deve ativar um subprograma que recebe um vetor de números de ponto flutuante e os ordene utilizando *Bubble Sort*.

- d) Por fim, o programa principal deve escrever na saída padrão o vetor ordenado utilizando o mesmo subprograma aplicada no item (b).

### Entrada

A entrada é composta por um único valor inteiro N que indica o tamanho do vetor a ser ordenado. É garantido que o usuário informará valores inteiros maiores ou iguais a zero. Logo, nenhum tratamento de consistência precisa ser aplicado pelo seu programa.

### Saída

A saída consiste na impressão de duas sequências de N valores, conforme apresentado no exemplo, assumindo duas casas decimais de precisão por valor. A primeira sequência corresponde ao vetor antes da ordenação e a segunda ao vetor após a ordenação.

### Exemplos

| Entrada | Saída  |
|---------|--|
| 5       | 0.12 1.00 0.25 0.33 0.98<br>0.12 0.25 0.33 0.98 1.00 |

### Distribuição de Pontos

Item (a) – 0,5 ponto; Itens (b) e (d) – 0,5 ponto; Item (c) – 2,0 pontos.

### **4ª Questão (2,0 pontos)**

Uma situação que todo programador enfrenta, por mais experiente que seja, é lidar com erros de sintaxe e problemas que levem a erros execução introduzidos por engano no código fonte. Nessa questão você deverá analisar o código fornecido e identificar os erros contidos nele. Para cada erro, indique claramente uma solução para sua correção.

```
def fatorial(k)
    if k > 1:
        return fatorial(k - 1) * k
    else:
        return "1"

n = int(input(Informe um valor inteiro não negativo: ))
while n < 0:
    n = int(input(Valor inválido. Informe um valor inteiro não negativo: ))

if n % 2 = 0:
    print("O valor %d é par." % n)
else:
    print("O valor %d é ímpar." % n)

r = fatorial(n)
print("O fatorial de %d é %d." % n, r)
```

### Distribuição de Pontos

A distribuição de pontos não é informada para evitar a dedução de quantos erros foram introduzidos no código fornecido.

**Boa Avaliação!**