



Fundação Centro de Ciências e Educação Superior a Distância do Estado do Rio de Janeiro
Centro de Educação Superior a Distância do Estado do Rio de Janeiro

AP1 – Fundamentos de Programação – 1/2025

Código da disciplina **EAD 05029**

Nome: _____

Matrícula: _____

Polo: _____

Atenção!

- Para cada folha de respostas que utilizar, **antes de começar a resolver as questões**, preencha conforme **modelo abaixo** (pintando os respectivos espaços na parte superior da folha) o **número do CPF** ou o **número da Matrícula**, o **código da disciplina** (indicado acima em negrito) e o **número da folha**.

PADRÃO DE PREENCHIMENTO NA FOLHA DE RESPOSTAS

1 UM 2 DOIS 3 TRÊS 4 QUATRO 5 CINCO 6 SEIS 7 SETE 8 OITO 9 NOVE 0 ZERO

- **Preencha o número total de folhas somente quando for entregar a prova!**
 - Identifique a Prova e as Folhas de respostas, colocando Nome, Matrícula e Polo.
 - É expressamente proibido o uso de qualquer instrumento que sirva para cálculo como também qualquer material que sirva de consulta.
 - Devolver esta prova e as Folhas de Respostas ao aplicador.
 - Somente utilize caneta esferográfica com tinta azul ou preta para registro das resoluções nas Folhas de Respostas.
 - As Folhas de Respostas serão o único material considerado para correção. Quaisquer anotações feitas fora deste espaço, mesmo que em folha de rascunho, serão ignoradas.
 - Não amasse, dobre ou rasure as Folhas de Respostas, pois isto pode inviabilizar a digitalização e a correção.
-

**Curso de Tecnologia em Sistemas de Computação
Disciplina Fundamentos de Programação - EAD 05029**

**Professores: Luís Felipe Ignácio Cunha – IC/UFF e
Uéverton dos Santos Souza – IC/UFF**

AP1 – Fundamentos de Programação – 1/2025

IMPORTANTE

- Serão aceitos apenas soluções escritas na linguagem Python 3.
 - Não serão aceitas respostas que usem bibliotecas.
 - Prova sem consulta e sem uso de qualquer aparato eletrônico.
 - 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.
-

Boa Avaliação!

1a Questão (2 pontos)

Descreva a finalidade do programa apresentado abaixo, e explique do que se trata a saída desse programa. Considere que o programa recebe como entrada padrão uma sequência de números inteiros.

```
salada = input()
def receita(ingredientes):
    ingredientes = ingredientes.split()
    alface = []
    for i in range(len(ingredientes)-1):
        for j in range(i+1, len(ingredientes)):
            if ingredientes[i] == ingredientes[j]:
                alface.append(ingredientes[j])
    return alface
saida = receita(salada)
print(f"A saída do programa é: {saida}")
```

Resposta: O objetivo do programa é determinar para cada elemento os elementos subsequentes iguais a ele, adicionando essas repetições em uma lista. Portanto, a saída é composta pelas sequências de repetições de cada elemento concatenadas em ordem.

2a Questão (3 pontos)

Considere o subprograma abaixo que recebe como entrada uma sequência com números inteiros.

- a) Defina a finalidade do programa, e explique do que se trata a saída do programa.
- b) Explique a finalidade da variável xyz.
- c) O código do subprograma possui erros? Em caso positivo indique as correções.

```
def abcd(efg):  
    efg = efg.split()  
    hij = []  
    for i in range(len(efg)-1, -1, -1):  
        hij.append(efg[i])  
    xyz = " "  
    return xyz.join(hij)
```

Resposta: O objetivo do programa é determinar a sequência obtida pela inversão da sequência de entrada. A variável xyz é o separador da string formada pelo join a partir da lista de inteiros recebidos. O código acima não possui erros.

3ª Questão (5,0 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:

- a) 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.
- b) Em seguida, o programa principal deve ativar um subprograma que escreve na saída padrão o vetor gerado (veja a formação no exemplo).
- c) 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 0.12 0.25 0.33 0.98 1.00

Resposta:

Subprogramas

```
def obterValores():
```

```
    from random import random
```

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

```
    lista = []
```

```
    for _ in range(n):
```

```
        lista.append(random())
```

```
    return lista
```

```
def escreverValores(lista):
```

```
    for valor in lista:
```

```
        print("%1.2f" % valor, end=" ")
```

```
    print()
```

```
def ordenarValores(lista):
```

```
    for n in range(len(lista) - 1, 0, -1):
```

```
        for i in range(n):
```

```
            if lista[i] > lista[i + 1]:
```

```
                lista[i], lista[i + 1] = lista[i + 1], lista[i]
```

Programa Principal

```
valores = obterValores()
```

```
escreverValores(valores)
```

```
ordenarValores(valores)
```

```
escreverValores(valores)
```