

Centro de Educação Superior a Distância do  
Estado do Rio de Janeiro – CEDERJ

Curso de Tecnologia em Sistemas de Computação – TSC

EAD-05.009 Fundamentos de Programação

# **Caderno de Exercícios**

## **Aula 10**

*(Estrutura de Dados – Conjunto)*

Professores

Dante Corbucci Filho  
Leandro A. F. Fernandes

## Instruções

- Utilize Python 3 e a IDE PyCharm na elaboração de soluções para os problemas propostos;
- Siga o formato apresentado na descrição da saída, caso contrário não é garantido que a saída emitida será considerada correta.

## Referências Autorais

Os exercícios apresentados nesta lista foram baseados em questões sugeridas anteriormente em Avaliações a Distância e Presenciais (ADs e APs) do curso de Fundamentos de Programação.

## Problema A

Sejam  $A$ ,  $B$  e  $C$  três conjuntos contendo valores inteiros gerados aleatoriamente no intervalo  $[1,20]$ . Escreva um programa que contenha quatro subprogramas, usando as seguintes definições:

```
def quantA(A, B, C):  
    # Retorna a quantidade elementos que pertencem aos três  
    # conjuntos.  
  
def quantB(A, B, C):  
    # Retorna a quantidade elementos que pertencem a exatamente  
    # dois conjuntos.  
  
def quantC(A, B, C):  
    # Retorna a quantidade elementos que pertencem a exatamente  
    # um conjunto.  
  
def quantD(A, B, C):  
    # Retorna a quantidade elementos do conjunto Universo que  
    # não pertencem a nenhum dos conjuntos A, B ou C.
```

### Entrada

A entrada consiste de três valores inteiros  $1 \leq N \leq 10$ , cada um em uma linha, que representam a quantidade de elementos dos conjuntos  $A$ ,  $B$  e  $C$ , respectivamente (**OBS: Não escreva nenhuma mensagem na tela, a entrada deve ser composta APENAS dos valores**).

### Saída

A saída consiste no retorno de cada um dos subprogramas, cada saída em uma linha.

### Exemplo

Entrada	Saída
4	2
5	2
8	7
	13

### Conjuntos (exemplo)

$A = \{1, 5, 10, 15\}$

$B = \{1, 2, 3, 4, 5\}$

$C = \{1, 3, 5, 7, 9, 11, 13, 15\}$

## Problema B

Sejam  $A$  e  $B$  dois conjuntos, cada um contendo no máximo 10 inteiros no intervalo  $[1,30]$ . Faça um programa que peça ao usuário o número de elementos do conjunto  $A$ , com a seguinte mensagem sendo exibida na saída padrão:

“Insira o tamanho do conjunto  $A$ : ”

Em seguida peça ao usuário os elementos do conjunto  $A$  (todos os valores devem ser lidos da mesma linha). A seguinte mensagem deve ser exibida:

“Insira os elementos do conjunto  $A$ : ”

Em seguida faça os mesmos procedimentos acima para o conjunto  $B$ , seguindo os mesmos critérios do conjunto  $A$ . As mensagens para o conjunto  $B$  devem ser:

“Insira o tamanho do conjunto  $B$ : ”

e

“Insira os elementos do conjunto  $B$ : ”

Por fim, seu programa deve:

- Mostrar na tela os elementos de  $A$  (na mesma linha). A seguinte mensagem deve ser mostrada: “Elementos de  $A$ : ”
- Mostrar na tela os elementos de  $B$  (na mesma linha). A seguinte mensagem deve ser mostrada: “Elementos de  $A$ : ”
- Mostrar na tela os elementos de  $A \cup B$  (na mesma linha). A seguinte mensagem deve ser mostrada: “ $A$  união  $B$ : ”
- Mostrar na tela os elementos de  $A - B$  (na mesma linha). A seguinte mensagem deve ser mostrada: “ $A$  diferença  $B$ : ”
- Mostrar na tela os elementos de  $B - A$  (na mesma linha). A seguinte mensagem deve ser mostrada: “ $B$  diferença  $A$ : ”
- Mostrar na tela os elementos de  $A \cap B$  (na mesma linha). A seguinte mensagem deve ser mostrada: “ $A$  interseção  $B$ : ”

## Problema C

Faça um programa contendo um subprograma que receba dois conjuntos  $X$  e  $Y$  como parâmetros de entrada e retorne o valor:

- 0 (zero), caso a interseção dos conjuntos seja vazia,
- 1, caso os conjuntos sejam iguais, ou
- 2, se nenhum dos dois casos anteriores acontecer.

O subprograma deverá atender a seguinte especificação:

```
def relacaoConjunto(A, B):
    ...
    return valor
```

No programa principal, peça ao usuário para digitar o tamanho dos conjuntos  $X$  e  $Y$ , até que o valor 0 (zero) seja digitado (não deve ser processado). Cada conjunto é formado por valores no intervalo  $[-10,10]$  gerados de forma aleatória. Em seguida, exiba na tela os conjuntos  $X$  e  $Y$  gerados (cada conjunto em uma linha) e a saída do subprograma `relacaoConjunto()`.

### Entrada

A entrada consiste de várias linhas, uma para cada caso. Os valores são do tipo inteiro, com  $1 \leq N \leq 50$  representando a quantidade de elementos dos conjuntos  $X$  e  $Y$ .

### Saída

Para cada caso, imprima os elementos dos conjuntos  $X$  e  $Y$ , respectivamente, separados por espaços em branco e cada conjunto em uma linha. Depois imprima (na linha seguinte) o retorno do subprograma pedido.

### Exemplo

Entrada	Saída
10	-10 -8 -6 -4 -2 0 2 4 6 8
3	-9 -7 -5 -3 -1 1 3 5 7 9
5	0
0	1 7 -3
	-3 5 -4
	2
	8 4 2 9 1
	2 9 4 1 8
	1

## Problema D

Faça um programa que peça ao usuário a quantidade de elementos de 3 conjuntos  $A$ ,  $B$  e  $C$  (os conjuntos podem ter quantidades diferentes de elementos). As seguintes mensagens devem ser exibidas na saída padrão:

“Insira o tamanho do conjunto A: ”

“Insira o tamanho do conjunto B: ”

“Insira o tamanho do conjunto C: ”

Uma vez informados os tamanhos, os conjuntos devem ser gerados de forma aleatória, com elementos no intervalo  $[0,20]$ . Seu programa deve mostrar na saída padrão:

- a) a quantidade e os elementos de  $(A \cup B) - C$
- b) a quantidade de primos e os elementos de  $(B - A) \cap C$
- c) a quantidade de números ímpares e os elementos de  $(C - B) \cup A$
- d) a quantidade de números pares e os elementos de  $(B - (A \cap B)) \cup C$

Para cada um dos itens acima, utilize a seguinte formatação de saída (cada resposta numa linha):

“Quantidade de elementos em (a): ”

“Elementos em (a): ”

Faça o mesmo para os demais itens.

**Sugestão:** Use subprogramas para dividir os problemas em partes menores.

## Problema E

Faça um programa que peça ao usuário para digitar dois subconjuntos  $A$  e  $B$ , de tamanhos  $N > 0$  e  $M > 0$ , respectivamente. Esses conjuntos devem conter somente números inteiros no intervalo  $[1,20]$ . Em seguida, peça ao usuário para digitar um valor inteiro  $1 \leq V \leq 20$ . Seu programa deve conter os seguintes subprogramas:

```
def estaContido(X, Y):  
    # função que recebe como parâmetros de entrada dois conjuntos  
    # chamados de X e Y e que retorna o valor 1 se  $X \subset Y$ , e 0  
    # (zero) caso contrário.  
  
def igual(X, Y):  
    # função que recebe como parâmetros de entrada dois conjuntos  
    # chamados de X e Y e que retorna o valor 1 se  $X = Y$ , e 0  
    # (zero) caso contrário.  
  
def pertence(valor,X):  
    # função que recebe como parâmetros de entrada um valor  
    # inteiro valor e um conjunto X, e retorna o valor 1 se  
    #  $\text{valor} \in X$ , e 0 (zero) caso contrário.
```

Seu programa deve mostrar na tela o valor de saída referente as seguintes operações relacionais:

- a)  $V \in A$
- b)  $V \in B$
- c)  $A = B$
- d)  $A \subset B$
- e)  $B \subset A$