Lista de Exercícios 1

Teoria da Computação — Mestrado em Computação Aplicada
Prof. Jefferson O. Andrade

Aluno: **Bruno Kobi Valadares de Amorim**

2022/2

1. Exercício

Implemente cada um dos programas a seguir como um programa SISO Python. Em cada caso, você pode assumir que a entrada é válida (ou seja, consiste em uma lista de números inteiros formatada corretamente).

1.a.

Escreva um programa que receba como entrada uma lista de inteiros separados por espaço em branco. A saída é uma string que representa a soma de cada segundo inteiro na lista. Por exemplo, se a entrada for "58 41 78 3 25 9", a saída será "53", porque $41 + 3 + 9 = 53$.

```
def pA(lista):
    x = lista.split(" ")
    soma: int = 0
    for i in range(len(x)):
        if i % 2 != 0:
            soma += int(x[i])
    return str(soma)
```

1.b.

Escreva um programa, semelhante ao programa em (a), mas somando cada terceiro elemento da entrada em vez de cada segundo elemento.

```
def pB(lista):
x = lista.split(" ")
```

```
    soma: int = 0
    for i in range(1, len(x)+1):
        if i % 3 == 0:
            soma += int(x[i-1])
            i += 3
    return str(soma)
```

1.c.

Escreva um programa de decisão que aceite uma lista de inteiros se a soma de cada terceiro elemento for maior que a soma de cada segundo elemento, e rejeite caso contrário. Seu programa deve importar e usar os programas de (a) e (b).

```
from pA import pA
from pB import pB

def pC(lista):
    i2 = int(pA(lista))
    i3 = int(pB(lista))

    if(i3 > i2):
        return 'true'
    else:
        return 'false'
```

2. Exercício

2.a. Qual o resultado de oooops("abc")?

Resposta = Accc

2.b. Qual o resultado de oooops("abcdefghij")?

Resposta = Acccccccccc

2.c. Qual o resultado de oooops("a")?

Resposta = undefined

2.d. Qual o resultado de oooops("008")?

Resposta = A88888888

2.e. Qual o resultado de oooops("8")?

Resposta = undefined

2.f. Qual o resultado de oooops(-11")?

Resposta = undefined

2.g. Descreva o conjunto de todas as strings I para as quais oooops(I) é indefinido.

Resposta = I é indefinido: (caso I.length $<$ 3) e (caso I seja inteiro $<$ 0)

$(I \subset Z^*) \ \cup \ (I = AB \mid A \ e \ B = [A - Z||0 - 9])$

3. Exercício

Use a prova por contradição para provar as seguintes afirmações:

3.a. Existem infinitos números inteiros positivos.

Inicialmente supomos que a quantidade números inteiros positivos é finita, então podemos supor que há apenas **n** números inteiros positivos.
Podemos colocar os números positivos em ordem **p1, p2 ... pn**, de tal forma que:
p1 $<$ **p2** $<$ **. . .** $<$ **pn**.
Com isto, teríamos que **pn** é o maior número positivo de todos.
Considere o número **pn** $+$ **1**, ele também é inteiro positivo, e é maior do que todos os demais números, incluindo **pn**. Isto contradiz a afirmação de que **pn** é o maior inteiro positivo de todos, então podemos garantir que existem infinitos números inteiros positivos.

3.b. Existem infinitos números inteiros negativos.

Inicialmente supomos que a quantidade números inteiros negativos é finita, então podemos supor que há apenas **n** números inteiros negativos.
Podemos colocar os números negativos em ordem **p1, p2 ... pn**, de tal forma que:
p1 $>$ **p2** $>$ **. . .** $>$ **pn**.
Com isto, teríamos que **pn** é o menor número negativo de todos.
Considere o número **pn - 1**, ele também é inteiro negativo, e é menor do que todos os demais números, incluindo **pn**. Isto contradiz a afirmação de que **pn** é o menor inteiro negativo de todos, então podemos garantir que existem infinitos números inteiros negativos.

3.c. Existem infinitos números pares.

Inicialmente supomos que a quantidade números pares é finita, então podemos supor que há apenas **n** números pares.
Podemos colocar os números pares em ordem **p1, p2 ... pn**, de tal forma que:
$\mathbf{p1} < \mathbf{p2} < . . . < \mathbf{pn}$.
Com isto, teríamos que **pn** é o maior número par de todos.
Considere o número $\mathbf{pn} + \mathbf{2}$, ele também é par, e é maior do que todos os demais números, incluindo **pn**. Isto contradiz a afirmação de que **pn** é o maior par de todos, então podemos garantir que existem infinitos números pares.

3.d. Não existe o menor número real positivo.

Inicialmente supomos que exista o menor número real positivo, então podemos supor que **M** é o menor numero real positivo.
Considere o número $\frac{\mathbf{M}}{2}$, ele também é real positivo, e é menor que **M**. Isto contradiz a afirmação de que **M** é o menor número real positivo, então podemos garantir que não existe o menor número real positivo.

4. Exercício

Como indicado na Figura 3.3 do livro "What Can Be Computed?", a saída do comando containsGAGA(rf('containsGAGA.py')) é "yes". Escreva uma nova versão deste programa, chamada containsGA_GA.py. Essa nova versão deve ser equivalente à antiga, i.e., produzir as mesmas respostas para as mesmas entradas. Além disso os comandoscontainsGA_GA(rf('containsGA_GA.py')) e containsGAGA(rf('containsGA_GA.py')) devem ambos retornar "no".

```python
def containsGA_GA(inString):
    chave = 'G'+'A'+'G'+'A'
    if chave in inString:
        return 'yes'
    else:
        return 'no'
```

5. Exercício

Após estudar os programas acima, determine a saída dos seguintes comando em Python:

5.a. yesOnStringApprox(rf('longerThan1K.py'), rf('longerThan1K.py'))

Resposta = **no**

5.b. yesOnStringApprox(rf('maybeLoop.py'), rf('maybeLoop.py'))

Resposta = **no**

5.c. yesOnSelfApprox(rf('longerThan1K.py'))

Resposta = **no**

5.d. notYesOnSelfApprox(rf('containsGAGA.py'))

Resposta = **no**