

## Prática 1: Aplicando o Teste Funcional

### Parte 1:

Com base na especificação abaixo, aplique os critérios funcionais particionamento em classes de equivalência e análise do valor limite e gere um conjunto de casos de teste que atende os dois critérios de teste.

Enviar os resultados em um arquivo, contendo: tabela de equivalência, passos da aplicação dos critérios e conjunto de casos de teste no site da **disciplina até 15/04**.

### Parte 2:

Usando a **Junit** como apoio e o código que implementa esse programa (estará disponível no site a partir do dia 15/04), analise o resultado de executar o programa com os casos de teste que você gerou. Faça um relatório com o resultado da execução, com informações sobre as saídas esperadas/geradas. Se defeitos foram identificados, apresentar quais e como deveria ser corrigido. **Entregar a atividade até 22/04**

### Specification for program “series”

#### Name

series – generate an additive series of numbers

#### Usage

**series** start end [stepsize]

#### Description

**series** prints the real numbers from **start** to **end**, one per line. **series** begins with **start** to which **stepsize** is repeatedly added or subtracted, as appropriate, to approach, possibly meet, but not pass **end**.

If all arguments are integers, only integers are produced in the output. The **stepsize** must be nonzero; if it is not specified, it is assumed to be of unit size (1). In all other cases, **series** prints an appropriate error message.

#### Example

To count from 1 to 100:

```
% series 1 100
```

To do the same, but backwards:

```
% series 100 1
```

## Limitations

The reported number of significant digits is limited. If the ratio of the series range to the **stepsize** is too large, several numbers in a row will be equal.

The maximum length of a series is limited to the size of the maximum long integer that can be represented on the machine in use. Exceeding this value has undefined results.