

Projeto de Laboratório de Computadores 2019/20 - my_seq

O projeto a desenvolver em Laboratório de Computadores consiste no desenvolvimento de vários utilitários que poderão ser utilizados para o processamento, na linha de comandos, de dados guardados em ficheiros.

Nesta ficha de trabalho deverá implementar o comando `my_seq`.

Nota: este enunciado poderá ainda sofrer alterações.

Comando my_seq

1. Comando `my_seq`

Sinopse:

```
my_seq [OPTION]... LAST
```

```
my_seq [OPTION]... FIRST LAST
```

```
my_seq [OPTION]... FIRST INCREMENT LAST
```

Descrição:

Implemente, em C, o comando `my_seq` que imprime na saída padrão, os números de *FIRST* até *LAST* com incrementos iguais a *INCREMENT*. Se *FIRST* ou *INCREMENT* forem omitidos, assume-se que o seu valor é 1. A sequência de números termina quando a soma do número atual e do *INCREMENT* se tornar maior do que *LAST*. *FIRST*, *INCREMENT* e *LAST* são interpretados como números de vírgula flutuante. *INCREMENT* é geralmente positivo se *FIRST* for menor que *LAST*, e *INCREMENT* é geralmente negativo se *FIRST* for maior que *LAST*.

Valor de retorno:

O programa deverá retornar (função `exit()`) o valor **0**, caso termine normalmente, ou um valor **>0**, caso ocorra algum erro.

Exemplos:

Exemplo 1:

```
$ ./my_seq 3
1
2
3
```

Exemplo 2:

```
$ ./my_seq 2 4
2
3
4
```

Exemplo 3:

```
$ ./my_seq 2 3 9
2
5
8
```

Exemplo 4:

```
$ ./my_seq 4 -1 2
4
3
2
```

Exemplo 5:

```
$ ./my_seq 2 0.1 2.4
2.0
2.1
2.2
2.3
2.4
```

2. Altere o programa `my_seq` para suportar as seguintes opções:

- `-f FORMAT` utilizar o estilo de formato da função `printf()` para imprimir cada número. Os caracteres de conversão válidos são *E*, *e*, *f*, *G*, *g*, e o *%*, bem como qualquer *flag* opcional ou campos para definição do tamanho e da precisão.
`-s string` utilizar *string* para separar os números. O separador por omissão é o `\n`.
- `-w` equalizar o tamanho de todos números utilizando zeros à esquerda, conforme necessário. Esta opção não é aplicável caso a opção `-f` seja utilizada.

Exemplo 1:

```
$ ./my_seq -s " " 3
1 2 3
```

Exemplo 2:

```
$ ./my_seq -s ", " 3
1, 2, 3,
```

Exemplo 3:

```
$ ./my_seq -f "num %g, " 3
num 1,
num 2,
num 3,
```

Exemplo 4:

```
$ ./my_seq -w 99 101
099
100
101
```

Sugestões

- Leia o manual do comando `seq` num terminal (`$ man seq`).
- Leia o manual das funções que considerar utilizar.
- Faça experiências com o comando `seq` num terminal. O seu programa deverá reproduzir o comportamento do comando `seq` .
- Para o processamento de opções, poderá recorrer à função `getopt()` das bibliotecas *unistd.h* e *getopt.h*. Poderá encontrar exemplos de utilização desta função nas ligações: [link1](#), [link2](#), e [link3](#).