

Descrição

 [Visualizar envios](#)

## EP04

**Disponível a partir de:** terça, 18 mai 2021, 00:01**Data de entrega:** sábado, 29 mai 2021, 23:59**Arquivos requeridos:** digitos.py ([Baixar](#))**Tipo de trabalho:** Trabalho individual**Redução por avaliação automática:** 0.5 **Avaliações livres:** 10

## EP04 - Dígitos verificadores

Fonte: <https://www.espacoeducar.net/2012/07/tirinhas-da-mafalda-reflexoes-sobre.html>O **limite de submissões livres** deste EP é 10.O **prazo de entrega** deste EP é 23h 59m do dia 24/05/2021. O sistema reabre a partir do dia 25/05 às 12h para envio de EPs com atraso, por mais 5 dias, recebendo desconto de 2 pontos por dia.

## Objetivos

Praticar o uso do comando de repetição `while` junto com expressões envolvendo operadores relacionais `==`, `!=`, `<`, `<=`, `>`, `>=` e valores inteiros, ou seja, valores do tipo `int`. Além disso, será praticado o uso dos operadores aritméticos `//` e `%`.

## Antes de começar

É recomendado que antes de fazer este EP você esteja relativamente familiarizado com [expressões aritméticas, relacionais e lógicas](#), [variáveis](#), [tipos e funções de entrada e saída](#), [seleção](#) e e repetições.

## Dígitos verificadores

Um [algarismo ou dígito](#) é um símbolo como `2` ou `7` usado em combinações para representar valores, como o `722`. No sistema de numeração decimal são utilizados os dígitos `0`, `1`, ..., `9`.

[Dígitos verificadores](#) (DVs, [check digit](#)) são dígitos concatenados a valores numéricos a fim de verificar a autenticidade ou detectar erros de digitação como um algarismo digitado incorretamente ou alguma permutação de algarismos adjacentes.

Números de documentos de identificação, como RG e CPF, utilizam dígitos de verificação. Cartões de crédito, boletos bancários, registros de encomendas, processos judiciais, títulos eleitorais e quaisquer outros códigos numéricos que necessitem de maior segurança também utilizam dígitos verificadores.

## DVs de MAC0110

Em MAC0110 os DVs de um número inteiro não-negativo serão formado por um par **dv1** e **dv2** de dígitos verificadores. Os dígitos **dv1** e **dv2** serão algarismos entre 0 e 9 calculados como descrito a seguir.

Para determinar o valor **dv1** de um número inteiro  $n = d_1d_2\dots d_{k-1}d_k$ , onde  $d_i$  é um dígito entre 0 e 9, devemos primeiro calcular a soma

$$1 \times d_1 + 2 \times d_2 + 3 \times d_3 + \dots + k \times d_k.$$

O dígito verificador **dv1** será o **resto da divisão** dessa soma por 11. Se esse resto for 10, então **dv1** será 0.

Por exemplo, para o número  $n=280012386$  temos que a soma acima é

$$1 \times 2 + 2 \times 8 + 3 \times 0 + 4 \times 0 + 5 \times 1 + 6 \times 2 + 7 \times 3 + 8 \times 8 + 9 \times 6 = 174,$$

e portanto **dv1** é 9.

O dígito verificador **dv2** será determinado de maneira semelhante. Primeiro calculamos uma soma levemente diferente da anterior

$$0 \times d_1 + 1 \times d_2 + 2 \times d_3 + 3 \times d_4 + \dots + (k-1) \times d_k + k \times dv1.$$

O dígito **dv2** será o **resto da divisão** dessa soma por 11. Se esse resto for 10, então **dv2** será 0.

Continuando o exemplo para o número  $n=280012386$ , essa nova soma é

$$0 \times 2 + 1 \times 8 + 2 \times 0 + 3 \times 0 + 4 \times 1 + 5 \times 2 + 6 \times 3 + 7 \times 8 + 8 \times 6 + 9 \times 9 = 225,$$

e portanto **dv2** é 5.

Dessa forma, os DVs de MAC0110 para o número **280012386** será o par **9 5**.

## Exercício programa

Faça um programa que leia um número inteiro **n** maior que zero e imprima os DVs de MAC0110 de **n**. As mensagens exibidas pelo seu programa devem ser **idênticas** àquelas exibidas no exemplos a seguir. O seu programa pode supor, sem verificar, que o dígito mais significativo de **n** é **diferente de zero**. Isso significa que seu programa pode supor que **n não é** um número como **007**, **012345678**, **0009**

Hmm. Testando o seu programa com os 9 primeiros dígitos do seu CPF, quais os DVs de MAC0110 obtidos?

Hmm. Qual é o menor inteiro positivo **n** para o qual os DVs de MAC0110 é **0 0**?

## Exemplos

A seguir estão exemplos de 6 execução do programa como seriam vistos no **Console do Spyder**.

O que aparece **vermelho** foi digitado pelo usuário. Observe que nesses exemplos o que aparece dentro do **runfile()** depois do **prompt** (como "In [1]:") depende de onde você salvou o seu arquivo e da forma que você usou para executá-lo. Ou seja, essa parte não precisa ser a mesma.

Observe também que parte dos espaçamentos é devido ao ambiente do Spyder. Por exemplo, a função **input("mensagem")** pula uma linha antes de exibir a mensagem, o que pode causar uma certa confusão inicial.

### Exemplo 1

```
In [1]: runfile('/Users/aluno/mac0110/ep03/digitos.py', wdir='/Users/aluno/mac0110/ep03/')
Digite n: 1
DV's = 1 1
In [2]:
```

### Exemplo 2

```
In [2]: runfile('/Users/aluno/mac0110/ep03/digitos.py', wdir='/Users/aluno/mac0110/ep03/')
Digite n: 77
DV's = 0 7
In [3]:
```

### Exemplo 3

```
In [3]: runfile('/Users/aluno/mac0110/ep03/digitos.py', wdir='/Users/aluno/mac0110/ep03/')

```

```
  Digite n: 15
  DVs = 0 5

```

```
In [4]:

```

## Exemplo 4

```
In [4]: runfile('/Users/aluno/mac0110/ep03/digitos.py', wdir='/Users/aluno/mac0110/ep03/')

```

```
  Digite n: 123456789
  DVs = 0 9

```

```
In [5]:

```

## Exemplo 5

```
In [5]: runfile('/Users/aluno/mac0110/ep03/digitos.py', wdir='/Users/aluno/mac0110/ep03/')

```

```
  Digite n: 908070605040302010
  DVs = 0 9

```

```
In [6]:

```

## Exemplo 6

```
In [6]: runfile('/Users/aluno/mac0110/ep03/digitos.py', wdir='/Users/aluno/mac0110/ep03/')

```

```
  Digite n: 1234567890987654321
  DVs = 9 2

```

```
In [7]:

```

## Roteiro

- **Baixe** o arquivo `digitos.py` para uma pasta no computador que estiver usando. Este é o único arquivo que deverá ser depositado nesta página.
- **Leia** o cabeçalho com atenção e **preencha** o seu nome e número USP.
- **Implemente e teste** o seu programa.
- Antes de depositar o seu EP, **verifique** se ele está produzindo exatamente a mesma saída mostrada nos três exemplos anteriores.
- Depois de testar o seu EP, **deposite-o** na página do EP.

O EP03 pode ser enviado 10 vezes sem prejuízo de nota. A partir da 11ª submissão, haverá um desconto de 0.5, meio ponto, por submissão. Verifique as mensagens do programa `avaliador` e, caso necessário, faça eventuais correções e resubmeta o seu EP. Evite submissões desnecessárias, para não ultrapassar o limite de 10 submissões. Procure depurar o seu programa no seu computador, usando o Spyder.

Submissões feitas após às 23h 59m do dia 24/05/2021 receberão desconto de 2 pontos por dia de atraso. O EP não poderá ser enviado após 29/05/2021.

## Honestidade Acadêmica

Esse é um exercício individual, não em grupo. Isso não significa que você não pode receber ajuda de outras pessoas, inclusive de seus colegas. De uma forma geral, gostaríamos de incentivar as discussões de ideias, conceitos e alternativas de solução. Nossa maior recomendação é evitar olhar o código fonte de uma solução antes de escrever o seu programa. Em caso de dúvida, consulte nossa [política de colaboração](#).

De forma sucinta, evite as seguintes ações que caracterizam desonestidade acadêmica na realização dos trabalhos individuais desta disciplina:

- buscar e obter uma solução, parcial ou completa, correta ou não, de um EP na internet ou qualquer outro meio físico ou virtual, durante o período de submissão do referido EP;
- solicitar ou obter uma cópia, parcial ou completa, correta ou não, da solução de um EP durante o seu período de submissão;
- permitir que um colega acesse uma cópia, parcial ou completa, correta ou não, do seu EP, durante o período de submissão;
- ainda mais grave é o plágio, que se configura pela utilização de qualquer material não visto em aula ou não descrito no enunciado, que não seja de sua autoria, em parte ou ao todo, e entregar, com ou sem edição, como se fosse seu trabalho, para ser avaliado.

## Arquivos requeridos

`digitos.py`

```

1  # -*- coding: utf-8 -*-
2  #-----
3  # LEIA E PREENCHA O CABEÇALHO
4  # NÃO ALTERE OS NOMES DAS FUNÇÕES, MÉTODOS E ATRIBUTOS
5  # NÃO APAGUE OS DOCSTRINGS
6  #-----
7
8  '''
9      Nome:
10     NUSP:
11
12     Ao preencher esse cabeçalho com o meu nome e o meu número USP,
13     declaro que todas as partes originais desse exercício programa (EP)
14     foram desenvolvidas e implementadas por mim e que portanto não
15     constituem desonestidade acadêmica ou plágio.
16     Declaro também que sou responsável por todas as cópias desse
17     programa e que não distribui ou facilitei a sua distribuição.
18     Estou ciente que os casos de plágio e desonestidade acadêmica
19     serão tratados segundo os critérios divulgados na página da
20     disciplina.
21     Entendo que EPs sem assinatura devem receber nota zero e, ainda
22     assim, poderão ser punidos por desonestidade acadêmica.
23
24     Abaixo descreva qualquer ajuda que você recebeu para fazer este
25     EP. Inclua qualquer ajuda recebida por pessoas (inclusive
26     monitores e colegas). Com exceção de material de MAC0110, caso
27     você tenha utilizado alguma informação, trecho de código,...
28     indique esse fato abaixo para que o seu programa não seja
29     considerado plágio ou irregular.
30
31     Exemplo:
32
33         A monitora me explicou que eu devia utilizar a função int() quando
34         fazemos leitura de números inteiros.
35
36         A minha função quicksort() foi baseada na descrição encontrada na
37         página https://www.ime.usp.br/~pf/algoritmos/aulas/quick.html.
38
39     Descrição de ajuda ou indicação de fonte:
40
41     '''
42
43     # escreva seu programa a seguir
44

```

[VPL](#)

◀ EP05 "test-drive"

Seguir para...

EP04 "test-drive" ▶

Você acessou como Joao Pedro Apolonio de Sousa Matos (Sair)  
MAC0110-2021

Disciplinas »

2021

2020

2019

2018

2017

2016

2015

2014

2013

2012

AACCs/FFLCH

Pró-Reitoria de Pós-Graduação

Outros

Suporte »

Documentação

HelpDesk e Contato

[Guia de uso](#)

[Sobre](#)

[Português - Brasil \(pt\\_br\)](#)

[Deutsch \(de\)](#)

[English \(en\)](#)

[Español - Internacional \(es\)](#)

[Français \(fr\)](#)

[Português - Brasil \(pt\\_br\)](#)