

Descrição

[Visualizar envios](#)

EP03

Disponível a partir de: terça, 11 mai 2021, 00:01

Data de entrega: sábado, 22 mai 2021, 23:59

Arquivos requeridos: carinha.py ([Baixar](#))

Tipo de trabalho: Trabalho individual

Redução por avaliação automática: 0.5 **Avaliações livres:** 10

EP03 - Carinha



[Saber em Rede](#)

O **limite de submissões livres** deste EP é 10.

O **prazo de entrega** deste EP é 23h 59m do dia 17/05/2021. O sistema reabre a partir do dia 18/05 às 12h para envio de EPs com atraso, por mais 5 dias, recebendo desconto de 2 pontos por dia.

Objetivos

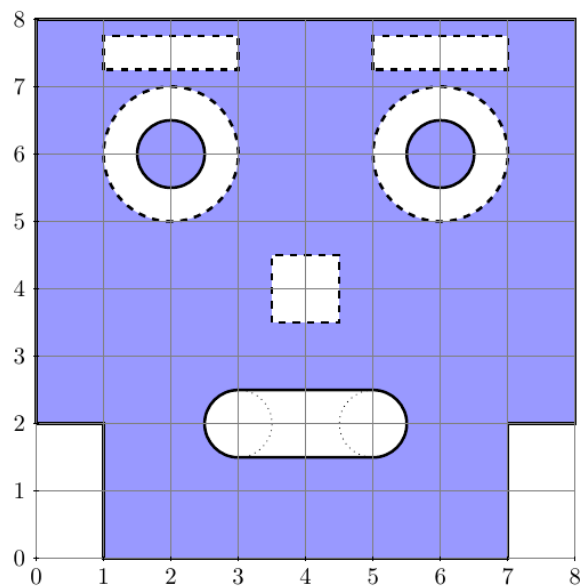
Praticar o [pensamento aplicado a resolução de problemas computacionais](#)... por meio do [pensamentos lógico e aritmético](#) e [pensamentos alternativos](#).

As [Gravações, rabiscos, exercícios e programas das reuniões anteriores](#) e as leituras recomendadas podem ser úteis.

Introdução

A motivação para esse exercício é, mais uma vez, treinar a sua atenção a detalhes, muitos detalhes. Vamos ver também uma forma de codificar informações de maneira simples e de forma compacta.

Considere a seguinte **região** ou **carinha** no plano.



As linhas cinzas de uma **grade** foram desenhadas apenas para facilitar a visualização das coordenadas. Chamaremos de **azul** qualquer ponto (x, y) que está em uma região azulada e de **branco** os demais pontos. Os pontos nos contornos em linhas **sólidas** entre regiões azuladas e brancas como $(0, 8)$, $(4, 8)$, $(8, 8)$, $(2, 0)$ e $(6, 0)$ **são azuis**. Já, os pontos nos contornos em linhas **tracejadas** como $(3.5, 3.5)$, $(4.5, 3.5)$ e $(2, 7.25)$ **são brancos**.

Grande parte dos pontos extremos nos contornos da carinha têm como coordenadas um par (x, y) de números inteiros, como os pontos $(0, 2)$, $(1, 2)$, $(1, 0)$ da **bochecha** esquerda da carinha. Há pontos extremos nos contornos que têm coordenadas da forma $.5$, como os pontos $(3.5, 3.5)$ e $(4.5, 4.5)$ do **nariz**. Há ainda pontos extremos no contornos que têm coordenadas da forma $.25$ ou $.75$, como os ponto $(5, 7.25)$ e $(7, 7.75)$ da **sobrancelha** direita. Finalmente, há pontos em contornos que distam 1 ou 0.5 de algum ponto de referência que é centro de um círculo. Como exemplo, considere os pontos do **olho** esquerdo que distam 1 do ponto $(2, 6)$ e os pontos da **íris** do olho esquerdo que distam 0.5 do ponto $(2, 6)$ e os pontos da **boca** que distam 0.5 do ponto $(3, 2)$.

O que você deve fazer

Escreva um programa que **leia** 2 números reais que correspondem às coordenadas x e y de um *ponto* e **imprima** **azul** se o ponto (x, y) for azul e **branco** se o ponto (x, y) for branco

O arquivo `carinha.py` contém um esqueleto do programa.

O comportamento esperado do programa é ilustrado nos exemplos a seguir.

Exemplos

A seguir estão exemplos de 12 execuções do programa. Números em **vermelho** foram digitados pelo usuário.

Exemplo 1

```
Digite x: -10
Digite y: 10
branco
```

Exemplo 2

```
Digite x: 2
Digite y: 7.25
branco
```

Exemplo 3

```
Digite x: 2
Digite y: 7
branco
```

Exemplo 4

```
Digite x: 2
Digite y: 6.5
azul
```

Exemplo 5

```
Digite x: 1
Digite y: 1
azul
```

Exemplo 6

```
Digite x: 7.5
Digite y: 0.5
branco
```

Exemplo 7

```
Digite x: 2
Digite y: 2.5
azul
```

Exemplo 8

```
Digite x: 2
Digite y: 4
azul
```

Exemplo 9

```
Digite x: 4.5
Digite y: 4
branco
```

Exemplo 10

```
Digite x: 8
Digite y: 8
azul
```

Exemplo 11

```
Digite x: 5
Digite y: 7.75
branco
```

Exemplo 12

```
Digite x: 6
Digite y: 7.25
branco
```

Roteiro

- **Baixe** o arquivo `carinha.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. Procure entender e prever o resultado de cada teste. Teste o programa para vários pontos.
- Antes de depositar o seu EP, **verifique** se ele está produzindo exatamente a mesma saída mostrada nos exemplos apresentados.
- Depois de testar o seu EP, **deposite-o** na página do EP e **veja** o resultado da avaliação.

Este EP 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 mensagem 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, ou qualquer outra ferramenta que você tiver familiaridade.

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

carinha.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     #####
44     ## ESCREVA SEU PROGRAMA ABAIXO
45
```

[VPL](#)

◀ EP04 "test-drive"

Seguir para...

EP03 "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)