

Universidade do Estado do Amazonas
Escola Superior de Tecnologia
Data: 14 de Janeiro de 2023
Professora: Elloá B. Guedes
Disciplina: Fundamentos Teóricos da Computação
Monitora: Débora Souza Barros

PROJETO PRÁTICO 1 TRANSAÇÕES PIX

1 Conhecendo a plataforma Run.Codes

A disciplina de *Fundamentos Teóricos da Computação* possui projetos práticos em sua avaliação, os quais possuem **caráter individual** e **obrigatório** e que serão executados por intermédio da plataforma *Run.Codes*.

O primeiro passo a ser realizado é o cadastro na plataforma. Acesse o site run.codes e, utilizando o seu **nome completo** (iniciais em maiúscula) e **e-mail institucional**. Após esta etapa, procure a disciplina **Fundamentos da Engenharia de Computação II**¹ (ESTECP006) e cadastre-se na turma **FTC 2022.2** utilizando o código **YVJ7**. Todos os alunos devem obrigatoriamente se cadastrarem até o dia **17/01/2023**. O não cadastro, o cadastro em turma incorreta ou a não realização dos exercícios nos prazos estabelecidos culminará em nota zero.

Para quem não conhece, o run.codes é uma plataforma automatizada para testes de entrada e saída. Cada aluno submete o seu código escrito na linguagem Python e este código é submetido a um conjunto de testes, previamente escritos e cadastrados pela professora e monitor. A nota do aluno é individual e obtida de maneira automática, correspondendo ao percentual de acertos nos testes. Por exemplo, se há 15 casos de testes cadastrados e o aluno acertou 12, a nota obtida é 8.

A partir do momento em que o exercício inicia até o momento do seu encerramento, o aluno pode submeter o código para a plataforma quantas vezes quiser, sem que isso afete a nota final. Por exemplo, se um aluno submeteu 10 vezes e acertou 12/15 casos de teste, a nota será a mesma de um outro aluno que acertou 12/15 mas que submeteu 300 vezes.

Uma outra regra a ser considerada na plataforma é que a última versão do código é sempre a que será considerada. Imagine que o aluno João Última Hora está enlouquecidamente programando o exercício e já tem 14/15 casos corretos mas, nos segundos finais do prazo limite submete alterações em seu código e cai para 8/15 acertos. Se o sistema encerrar, a versão 8/15 será a

¹Nomenclatura antiga, sorry!

considerada para avaliação. Se João Última Hora tivesse aproveitado melhor o tempo e começado a resolver o exercício desde o momento em que ficou disponível na plataforma, não teria passado esse sufoco e ficado com uma nota final tão ruim. Todos podemos aprender com o drama de João Última Hora e evitar tais problemas.

Algumas dicas finais para você ter um bom desempenho nos problemas práticos são:

1. Considere que seu programa recebe uma entrada de cada vez;
2. Efetue testes em seu programa antes de submetê-lo ao run.codes. É uma forma simples de conhecer como seu programa se comporta e uma oportunidade de acertar mais testes logo de primeira;
3. Aproveite o tempo;
4. Há boatos de que você não deve deixar seu computador saber quando você está com pressa!

2 Apresentação do Problema

O Pix é o meio de pagamento brasileiro criado pelo Banco Central (BC) em que os recursos financeiros são transferidos entre contas em poucos segundos, a qualquer hora do dia. Foi concebido para ser prático, rápido, seguro, alavancar a competitividade e a eficiência do mercado, promover a inclusão financeira e, dentre outros, preencher uma série de lacunas existentes na cesta de instrumentos de pagamentos disponíveis atualmente à população. O Pix pode ser realizado a partir de uma conta corrente, conta poupança ou conta de pagamento pré-paga e 734 instituições já estão aptas a prover o Pix para seus clientes.

Neste projeto prático, vamos simular a validação de lotes de transações Pix por uma dada instituição financeira. Um lote de transações Pix é composto de duas partes: (1) uma lista de clientes e suas respectivas chaves Pix e (2) uma lista de transações. Um lote é considerado válido (True) quando todas as suas partes obedecem ou não violam os critérios listados a seguir. Em caso contrário, é inválido (False).

1. **Lista de Clientes.** Uma lista de clientes e suas respectivas chaves Pix. Cada cliente é listado em uma linha da entrada.
 - a) **Identificador do Cliente.** CPF (Pessoa Física) ou CNPJ (Pessoa Jurídica).
 - **CPF.** É o número de Cadastro de Pessoa Física e segue o padrão difundido no Brasil. Exemplo: xxx.xxx.xxx-xx. em que x é um dígito. Além de checar se a entrada está no formato de um CPF, deve-se adicionalmente aplicar o algoritmo para checar a validade do CPF fornecido. Um CPF válido é aquele que passa no padrão e também no teste de validade;

- **CNPJ.** No Brasil, o Cadastro Nacional da Pessoa Jurídica (CNPJ) é um número único que identifica uma pessoa jurídica e outros tipos de arranjo jurídico sem personalidade jurídica junto à Receita Federal brasileira. O número do CNPJ possui 14 caracteres numéricos, para qualquer empresa, dispostos conforme xx.xxx.xxx/xxxx-xx, em que x é um dígito. Além de checar se a entrada está no formato de um CNPJ, deve-se adicionalmente aplicar o algoritmo para checar a validade do CNPJ fornecido. Um CNPJ válido é aquele que passa no padrão e também no teste de validade.
- b) **Lista de Chaves.** Além do identificador do cliente, pode haver até quatro chaves Pix adicionais e distintas, as quais são separadas por um espaço em branco. Pode haver repetição de um mesmo tipo de chave para um dado cliente, desde que seu conteúdo seja distinto. As chaves são únicas. Os tipos de chave são:
 - i. **E-mail:** Seguindo um formato de email simplificado, em que o início do endereço é feito com uma letra, há um arroba, e pelo menos um ponto após o arroba. São exemplos de e-mails válidos: ebgcosta@uea.edu.br, c4rl0s@teste.com.au;
 - ii. **Chave Rápida.** As chaves rápidas são compostas de 4 pares de dígitos hexadecimais que podem ser letras de *A* até *F* e números de 0 a 9 e que são separados por ponto. Para aumentar a segurança, duas letras não podem aparecer juntas e nenhum par de números pode ter dígitos iguais. São exemplos de chaves rápidas válidas: 03.A5.2B.F8, 14.35.28.92, etc;
 - iii. **Telefone.** Uma sequência de dígitos e símbolos que no formato +55(xx)xxxx-xxxx, em que x é um dígito.
- 2. **Separador.** Uma string contendo 10 símbolos =;
- 3. **Lista de Transações.** Contém múltiplas transações Pix realizadas entre os clientes previamente listados. Todas as transações válidas obedecem a estrutura a seguir.
 - a) **Chave de Origem.** Uma das chaves previamente listadas;
 - b) **Chave de Destino.** Uma das chaves previamente listadas, desde que não seja pertencente ao sujeito que origina a transação;
 - c) **Valor da Transação.** Um valor positivo sem sinal em reais, denotado com dois dígitos para os centavos. Um exemplo de valor numa transação é R\$ 2.100,03.
 - d) **Timestamp.** Uma marca temporal da realização da transação: DD/MM/AAAA HH:NN, em que D,M,A,H,N denotam dígitos de 0 a 9 que compõem uma data e hora válidos;

- e) **Código de Segurança.** Uma sequência de 12 símbolos composta de 3 letras maiúsculas, 4 dígitos, 2 caracteres especiais (\$,@,%,(*)) e 3 letras minúsculas, dispostos em qualquer ordem.

Para resolver o problema em questão, você deve utilizar a linguagem de programação Python 3+ e obrigatoriamente fazer uso de expressões regulares. Soluções que não fizerem uso de expressões regulares serão anuladas. Os exemplos a seguir auxiliam a ilustrar entradas e saídas para o problema considerado.

3 Exemplos de Entradas e Saídas

Entrada
04.280.196/0001-76 uea@uea.edu.br +55(92)3348-7601 09.628.825/0001-20 03.A4.2B.F8 ===== uea@uea.edu.br 09.628.825/0001-20 R\$ 1.000.000,00 09/01/2023 09:03 ABC@@1234abc 03.A4.2B.F8 +55(92)3348-7601 R\$ 2.000.000,00 08/01/2022 10:10 FG@H*12ab34c
Saída
True

4 Observações Importantes

- Lembre-se, a entrada de dados é feita via **input** e a saída via **print**;
- Atenha-se exatamente ao padrão de entrada e saída fornecidos nos exemplos. Qualquer mensagem adicional na entrada ou na saída de dados pode culminar em incorretude;
- Cuidado ao copiar caracteres do PDF! Eles podem estar com codificação incorreta. Atente-se ao enunciado;
- A cada execução do programa será fornecida apenas uma entrada, cujo resultado deve ser exibido ao final do processamento;
- Na construção do seu programa você deve usar apenas os conceitos aprendidos em sala de aula. Respostas que utilizem bibliotecas prontas não serão consideradas;
- Em caso de plágio, todos os envolvidos receberão nota zero! O run.codes utiliza o MOSS² e verifica os códigos submetidos dois a dois em busca de similaridades intencionais;

²Leia mais em <https://theory.stanford.edu/~aiken/moss/>.

- Na execução do seu programa no *run.codes*, existem casos de testes que vão além dos exemplos mostrados. Esses casos de teste não serão revelados. Pense em exemplos de entradas e saídas que podem acontecer e melhore o seu código para capturá-las.

5 Prazos Importantes

- **Início.** 09/01/2023 (horário do servidor)
- **Encerramento.** 18/01/2023 às 23h55min (horário do servidor)

6 Links Úteis

- <https://docs.python.org/3/library/re.html>
- <https://developers.google.com/edu/python/regular-expressions>
- <https://www.bcb.gov.br/estabilidadefinanceira/pix>
- https://pt.wikipedia.org/wiki/Cadastro_de_Pessoas_F%C3%ADsicas
- https://pt.wikipedia.org/wiki/Cadastro_Nacional_da_Pessoa_Jur%C3%ADdica