Universidade do Estado do Amazonas
Escola Superior de Tecnologia
Data: 14 de março de 2017
Disciplina: Fundamentos Teóricos da Computação
Professores: Elloá B. Guedes
Aluno:

Projeto Prático 1
Loja de Apps

1 Inscrição no Run.Codes

Você deve usar a plataforma Run.Codes para realizar as tarefas descritas neste projeto. Caso não tenha cadastro, primeiramente o faça em `https://run.codes/`. Utilize o seu e-mail institucional. Após a realização do cadastro, procure a turma ESTECP006 Fundamentos da Engenharia da Computação II - Fundamentos Teóricos da Computação 2017.1. O código para matrícula nesta turma é ZSH8.

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 de programação determinada pelo professor da disciplina e este é submetido a um conjunto de testes previamente cadastrado pelo professor. A nota do aluno corresponde ao percentual de acertos nestes testes.

Algumas observações importantes. O aluno pode submeter quantas vezes quiser e efetuar diversos testes. A nota final não sofre influência do número de submissões, apenas do número de acertos ao final. Considere que seu programa recebe uma entrada de cada vez. Efetue testes em seu próprio computador antes de submeter ao run.codes, é uma forma mais simples de detectar o que pode estar havendo de errado com o seu código. Aproveite o tempo! Você tem até a data limite para submeter a versão final do seu código.

Do ponto de vista do professor, o run.codes é uma ferramenta excelente para detecção de plágio, evitando a cópia de resposta entre os estudantes.

2 Apresentação

Nas linguagens de programação, as *expressões regulares* fornecem uma maneira automatizada de verificar se determinadas entradas estão escritas de acordo com um padrão ou regra de formação. Seu objetivo neste projeto é resolver o problema a seguir utilizando expressões regulares na linguagem Python. Soluções que não utilizem expressões regulares serão desconsideradas.

Neste projeto você vai validar com a validação de compras de uma loja virtual de aplicativos. Cada compra deve incluir o login, CPF, e-mail e senha do cliente. Além disso, contém também especificações do app que incluem o nome do app, a versão, a plataforma. Se a compra for feita para presentear outras pessoas, deve incluir ao final o login e o e-mail dos destinatários. A seguir, alguns detalhes sobre os elementos que compõem uma compra deste tipo.

- **Login do usuário**. Sempre iniciado com um caractere minúsculo, podendo ser seguido de outros caracteres. Caracteres especiais não são permitidos, tais como $, #, espaço, nem dígitos etc. Não há limite de caracteres para o login. São exemplos de logins válidos: elloa, elloaGuedes, umLoginMuitoMuitoGrande, etc.

- **CPF do usuário**. Como estamos considerando apenas espectadores brasileiros, o CPF é utilizado para identificar o comprador. Este dado segue o padrão de CPFs amplamente conhecido. Exemplo: 123.456.789-01. 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, disponível em `http://www.macoratti.net/alg_cpf.htm`. Um CPF válido é aquele que passa no padrão e também no teste de validade;

- **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, elloa.uea@gmail.com, c4rl0s@teste.com.au.;

- **Senha**. Para dificultar a ação de hackers, as senhas nesta loja de apps são compostas de 4 pares de dígitos 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. Senhas válidas: $03.A5.2B.F8$, 14.35.28.92, etc.

- **Nome do App**. Qualquer sequência de caracteres compostas por letras do alfabeto, independente de estarem em maiúscula ou minúscula. Por exemplo: twitter, gmail, bancoBrasil, etc.

- **Versão do App**. A versão do app é composta pela versão principal e pela sub-versão separadas por um ponto, com a restrição de que a sub-versão não pode ser maior que a versão. Exemplos: 13.01, 2.1, 4.4, 3.2, 10.7, etc.

- **Plataforma**. A plataforma pode ser um dos seguintes valores: windows, mac, linux, ios, android, windowsPhone.

- **Compra para Presente**. Se a compra foi feita com o intuito de presentear outras pessoas, após o item anterior podem vir pares de logins e e-mails, em qualquer quantidade. Não sendo uma compra pra presente, estas informações não são necessárias. Os logins e e-mails fornecidos devem ser validados como explicado anteriormente.

Os elementos que compõem uma compra são apresentados em uma única linha conforme a ordem especificada, separados por espaços em branco. Uma compra é válida quando todos os seus elementos são caracterizados de maneira adequada, na ordem especificada, como apresentado anteriormente. Não precisa fazer checagens se há saldo na conta do usuário ou afins, assuma que isso irá compor outro módulo deste sistema e que uma outra pessoa está responsável por implementar isto. De maneira similar, não preocupe-se em saber se o cliente existe ou se o e-mail está ativo, imagine que há uma outra equipe empenhada para estes fins.

De maneira resumida, a entrada do seu problema é uma string contendo uma especificação de compra. A saída é a palavra "True" quando a compra é válida e "False" em caso contrário.

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.

3 Links Úteis

- `https://developers.google.com/edu/python/regular-expressions`
- `http://goo.gl/SWwe4R`
- `https://www.debuggex.com/cheatsheet/regex/python`

4 Exemplos de Entrada e Saída

- **Entrada**: elloa 581.612.159-60 elloa.uea@gmail.com 47.A2.C9.D5 google-classroom 1.0 iphone almirOliveira adjunior@uea.edu.br marciaSampaio marcia@professora.com ⇒ **Saída**: True
- **Entrada**: entradaErrada 581.612.159-60 1especial@hotmail.com AA.BB.CC.DD aplicativoMaluco 3.7 outraPlataforma **Saída**: False

5 Prazos Importantes

- Apresentação da atividade: 21/03
- Cadastro da atividade no Run.Codes: 21/03
- Data limite de entrega: 29/03, 23h55min no horário do servidor