

Avaliação Técnica - Programa de Trainee 01/22

Olá, seja bem-vinda ao primeiro processo seletivo do Programa de Trainee da L2 Code no ano de 2022. Nesta avaliação iremos considerar a sua capacidade em executar os 3 desafios propostos a seguir. Você pode resolver as questões utilizando **qualquer linguagem de programação** e tecnologia que lhe seja mais confortável.

Esperamos que você consiga completar **ao menos duas questões**, portanto, fique tranquila caso deixe alguma inacabada. Dedique-se para entender os enunciados e entregar o melhor resultado possível. E mesmo que dois dos desafios estejam incompletos, iremos considerar a sua vontade em respondê-los e o seu raciocínio lógico utilizado.

E tem mais uma coisa: digamos que você consiga completar apenas 1 dos desafios, mas o fez brilhantemente. Isto também será considerado na sua avaliação.

Temos apenas uma única regra determinante para esta avaliação: por favor, desenvolva seu código para resolução das atividades propostas, publique em seu GitHub pessoal e envie o link para o e-mail **rh@l2code.com.br**. Iremos avaliar o seu código e rodar a sua solução para termos um melhor entendimento do seu conhecimento.

Caso você não tenha uma conta ainda, esta será uma ótima oportunidade para cadastrar-se e inaugurá-lo durante o processo. O GitHub é uma plataforma online e gratuita, utilizada para criar/hospedar repositórios e demais arquivos, podendo acessá-los remotamente de qualquer dispositivo ao redor do mundo.

Estamos felizes pelo sucesso e aceitação deste processo seletivo pelo público feminino de programadoras e queremos as melhores para agregar ao nosso time, mas não há motivos para medo ou insegurança. Desempenhe o seu melhor e boa sorte!



QUESTÃO 1

Banco de dados SQL

O Problema:

Um comércio eletrônico está realizando uma análise dos produtos em estoque no seu banco de dados (Arquivo SQL para [download aqui](#) do banco de dados).

Precisamos da sua ajuda para:

- Retornar uma lista que apresente o nome das categorias e a quantidade de produtos totais em cada categoria.
- Apresentar um relatório com as 4 categorias que mais tiveram produtos comprados.

Atividade: Esperamos que você escreva 2 SQLs, um para cada item acima, que retorne como resultado o esperado nos Outputs utilizando a base de dados fornecida. A base foi usada no banco de dados Postgres, mas por se tratar de SQL padrão deve funcionar em outros SGBDs.

O primeiro Output deverá retornar:

name	sum
luxury	350
modern	13000
wood	850
vintage	1000
futuristic	38
sports	100
office	3999
home office	0
gamer	400

O segundo Output deverá retornar:

id	categories	sum
1	wood	5000
8	gamer	182
7	sports	10
6	futuristic	03

QUESTÃO 2

Tradutor de Linguagem

Nomes dos arquivos

- **Arquivo fonte:** verb.pas ou verb.c ou verb.cpp
- **Arquivo de entrada:** verb.in
- **Arquivo de saída:** verb.out

O Problema

Recentemente, um time de exploradores descobriu uma tribo perdida na floresta Amazônica. Os nativos falam em uma linguagem desconhecida pelos exploradores, e para facilitar a comunicação você deve escrever um classificador de verbos. O contato inicial com os nativos permitiram a dedução das seguintes regras:

1. Todos os verbos são regulares
2. Todos os verbos terminam com uma consoante seguida do sufixo 'en'
3. Existem apenas 3 tempos verbais: passado, presente e futuro
4. Para conjugar o verbo, os nativos substituem o sufixo 'en' por outro, de acordo com a seguinte tabela

Person	Present	Past	Future
1st	o	ei	ai
2nd	os	es	ais
3rd	a	e	i
4th	om	em	aem
5th	ons	est	aist
6th	am	im	aim

O verbo 'campten' tem a seguinte conjugação:

Person	Present	Past	Future
1st	campto	camptei	camptai
2nd	camptos	camptes	camptais

Person	Present	Past	Future
3rd	campta	campte	campti
4th	camptom	camptem	camptaem
5th	camptons	camptest	camptaist
6th	camptam	camptim	camptaim

O programa deve analisar uma palavra, verificar se é um verbo ou não, se for um verbo, deve indicar a pessoa e o tempo verbal em que ela está conjugada.

Formato de entrada

Uma lista de palavras, uma por linha.

Formato de saída

Uma palavra analisada por linha, seguida pelo resultado da análise, como mostrado nos exemplos.

Exemplos

Suposta a seguinte entrada:

```
casos
porre
corraem
picel
oficina
param
```

Então temos a seguinte saída:

```
casos - verb casen, present tense, 2nd person
porre - verb porren, past tense, 3rd person
corraem - verb corren, future tense, 4th person
picel - not a verb case
oficina - not a verb case
param - verb paren, present tense, 6th person
```

QUESTÃO 3

ALGOC - A Language to Generate Only Constants

Nomes dos arquivos

- **Arquivo fonte:** `algoc.pas` ou `algoc.c` ou `algoc.cpp`
- **Arquivo de entrada:** `algoc.in`
- **Arquivo de saída:** `algoc.out`

O Problema

Um ambicioso professor de Ciência da Computação quer se tornar famoso desenvolvendo uma nova linguagem de programação, ALGOC - A Language to Generate Only Constants. Esta linguagem é bem simples, e tem apenas quatro construtores:

- `PLUSONE` - cria a constante 1 (positivo)
- `MINUSONE` - cria a constante -1 (negativo)
- `INC` - adiciona um à constante criada
- `DUP` - multiplica a constante criada por dois

Um programa nesta linguagem é uma sequência destes construtores, um para cada linha, executados sequencialmente. O professor quer que os programas escritos nesta linguagem sejam simples, curtos e rápidos. Para atingir esta meta, ele adiciona as seguintes regras:

- Cada programa deve começar obrigatoriamente com `PLUSONE` ou `MINUSONE`
- As constantes devem ser geradas com o menor número de instruções possíveis
- Se uma constante `C` pode ser gerada com mais de um programa (com o mesmo número de instruções), então o programa mais rápido deve ser utilizado. Para este propósito, suponha que a instrução `DUP` é executada em `T` nanosegundos e a instrução `INC` em `2T` nanosegundos

Você foi contratada pelo professor para escrever vários programas de amostra, para que ele possa usar em conferências e demonstrar o poder da sua nova linguagem. O professor irá disponibilizar algumas constantes e a sua tarefa é escrever programas para gerar estas constantes, seguindo as regras acima.

Formato de entrada

O arquivo de entrada contém `N` constantes a serem avaliadas, uma por linha. Todos os números são inteiros diferentes de zero entre **-32768** e **32767**. A linha que contém o valor zero representa o fim do arquivo de entrada.

Formato de saída

Para cada constante, o seu programa deve printar uma linha dizendo **Constant XXX**, onde **XXX** é a constante sendo avaliada, seguido pelo programa mais eficiente para gerar esta constante, sendo uma instrução por linha.

Cada programa é finalizado com uma linha em branco.

Exemplos

Suposta a seguinte entrada:

```
3
-5
1
7
0
```

Então temos a seguinte saída:

```
Constant 3
PLUSONE
DUP
INC
```

```
Constant -5
MINUSONE
DUP
DUP
INC
DUP
INC
```

```
Constant 1
PLUSONE
```

```
Constant 7
PLUSONE
DUP
INC
DUP
INC
```