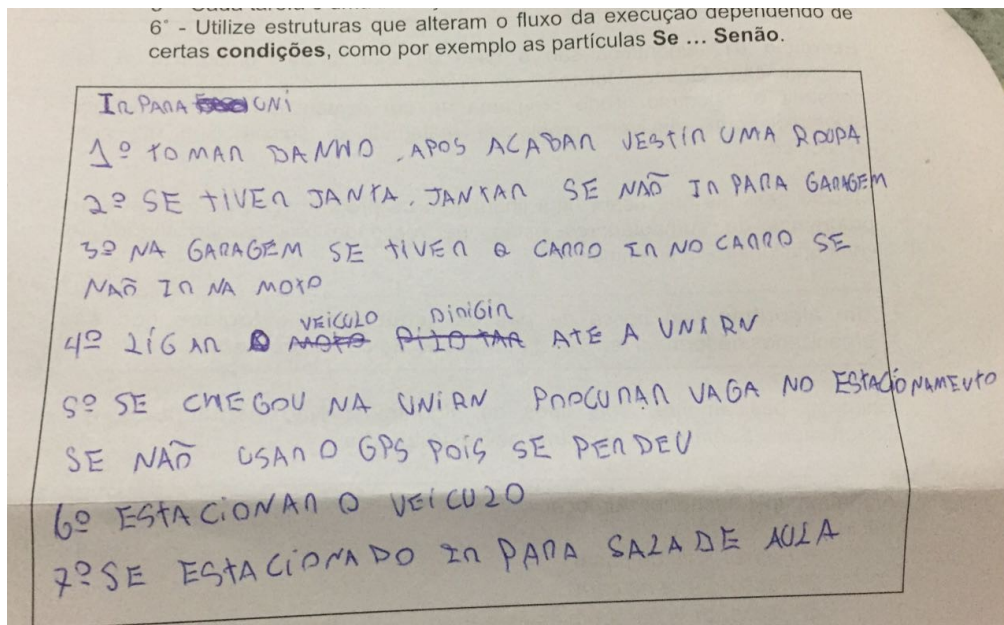


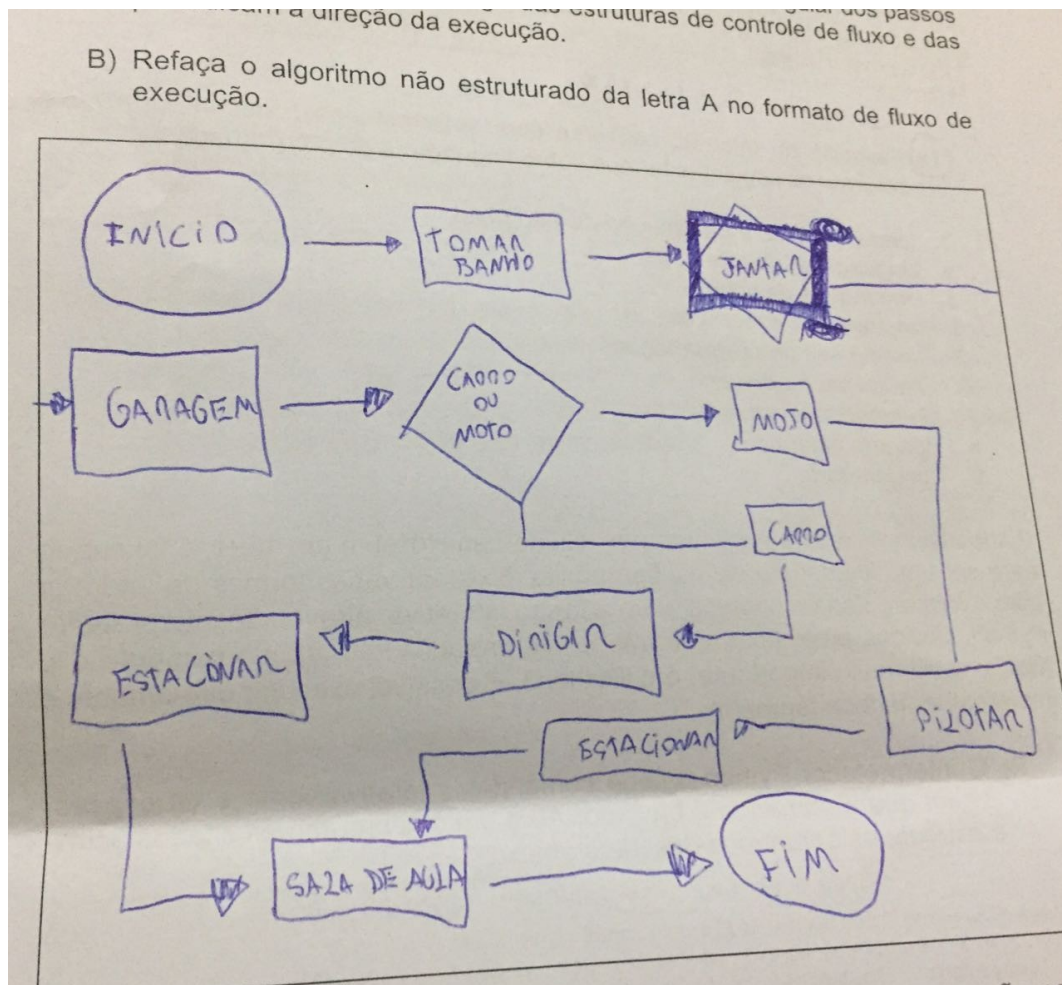
LISTA DE EXERCICIO 01 – RESPOSTAS

ALUNO: GUSTAVO HENRIQUE MARINHO DE OLIVEIRA – 2018B010153

A)



B)



D)

Linguagens de programação de alto nível → são linguagens que se parece o mais próximo possível de um idioma falado por nós, como por exemplo o Ruby e Python que praticamente você consegue ler frases em inglês dentro dos algoritmos feitos com estas linguagens

Linguagens interpretadas → Python, Ruby, JavaScript, são linguagens que têm a presença de um interpretador para ir lendo a instrução por instrução

Linguagens de script → é toda linguagem que é usada para fazer um script e script é uma sequência de comandos para executar uma tarefa

Linguagens imperativas → Na Ciência da Computação, programação imperativa é um paradigma de programação que descreve a computação como ações, enunciados ou comandos que mudam o estado (variáveis) de um programa. Muito parecido com o comportamento imperativo das linguagens naturais que expressam ordens, programas imperativos são uma sequência de comandos para o computador executar. O nome do paradigma, Imperativo, está ligado ao tempo verbal imperativo, onde o programador diz ao computador: faça isso, depois isso, depois aquilo... Este paradigma de programação se destaca pela simplicidade, uma vez que todo ser humano, ao se programar, o faz imperativamente, baseado na ideia de ações e estados, quase como um programa de computador.

Paradigmas de programação → Um paradigma de programação fornece e determina a visão que o programador possui sobre a estruturação e execução do programa. Por exemplo, em programação orientada a objetos, os programadores podem abstrair um programa como uma coleção de objetos que interagem entre si, enquanto em programação funcional os programadores abstraem o programa como uma sequência de funções executadas de modo empilhado.

Orientação a Objetos → é um tipo de paradigma de programação onde os programadores vão dando instruções como se tivesse se criando um objeto e suas características

Linguagens funcionais → Em ciência da computação, programação funcional é um paradigma de programação que trata a computação como uma avaliação de funções matemáticas e que evita estados ou dados mutáveis. Ela enfatiza a aplicação de funções, exemplos: Elixir, ML, Haskell

Tipagem dinâmica → Tipagem dinâmica é uma característica de determinadas linguagens de programação, que não exigem declarações de tipos de dados, pois são capazes de escolher que tipo utilizar dinamicamente para cada variável, podendo alterá-lo durante a compilação ou a execução do programa. Exemplo: Python, Ruby

Tipagem forte → Tipagem forte significa que o interpretador do Python avalia as expressões (evaluate) e não faz coerções automáticas entre tipos não compatíveis

E)

```
gustavo@gustavo-Lenovo:~$ python3
Python 3.5.2 (default, Nov 23 2017, 16:37:01)
[GCC 5.4.0 20160609] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> print("Ola mundo")
Ola mundo
>>> █
```

F) deu erro apenas quando remove os parênteses das ações solicitadas na questão

G)

```
gustavo@gustavo-Lenovo: ~  
ola mundo .  
>>> import this  
The Zen of Python, by Tim Peters  
  
Beautiful is better than ugly.  
Explicit is better than implicit.  
Simple is better than complex.  
Complex is better than complicated.  
Flat is better than nested.  
Sparse is better than dense.  
Readability counts.  
Special cases aren't special enough to break the rules.  
Although practicality beats purity.  
Errors should never pass silently.  
Unless explicitly silenced.  
In the face of ambiguity, refuse the temptation to guess.  
There should be one-- and preferably only one --obvious way to do it.  
Although that way may not be obvious at first unless you're Dutch.  
Now is better than never.  
Although never is often better than *right* now.  
If the implementation is hard to explain, it's a bad idea.  
If the implementation is easy to explain, it may be a good idea.  
Namespaces are one honking great idea -- let's do more of those!  
>>> █
```

H) as operações seguem o mesmo sentido da matemática normal e quando a parêntese é executado primeiro o que está dentro os números do tipo float são representados com ponto no lugar da vírgula que usamos na matemática comum, o número negativo é representado com o – na frente do número, as dízimas são imprimidas normais, sendo que com alguns comandos é possível limitar o número de casas decimais

Desafio 01

```
1 ▶ a= int(input("Digite A:"))  
2 b= int(input("Digite B:"))  
3 c= int(input("Digite C:"))  
4  
5 x=(b**2)-(4*a*c)  
6 x1=(-b+x)/(2*a)  
7 x2=(-b-x)/(2*a)  
8 print("x1",x1)  
9 print("x2",x2)  
10
```