Nome: Marcelo Vieira da Silva RA: 2041382211027

1. def invert\_word(p):

      pilha = Stack()

      List = p.split()

      for i in List:

       if i in "0123456789":

            pilha.push(int(i))

            pilha.imprimir()

       elif i in "ABCDEFGHIJKLMNOPRSRUVWXYZ" or i in "abcdefghijklmnoprstuvwxyz":

            pilha.push(i)

            pilha.imprimir()

       else:

           pilha.pop()

           invertido = pilha.pop()

      return invertido[: :-1]

invert\_word("Junior")

2. def generate\_palindrome(p):

      pilha = Stack()

      List = p.split()

      for i in List:

       if i in "0123456789":

            pilha.push(int(i))

            pilha.imprimir()

       elif i in "ABCDEFGHIJKLMNOPRSRUVWXYZ" or i in "abcdefghijklmnoprstuvwxyz":

            pilha.push(i)

            pilha.imprimir()

       else:

           pilha.pop()

           invertido = pilha.pop()

def verified\_palindrome(palindromo):

         if  invertido[i] != List:

                  print(invertido.reverse())

                  print('Esse número não é um palindromo')

         else:

                  print(p)

                  print("Esse número é um palindromo")

palindromo = input('Digite um número para descobrir se é um palíndromo: ')

generate\_palindrome(palindromo)

verified\_palindrome(palindromo)

3. class Stack():

    def \_\_init\_\_(self):

        self.items = []

    def is\_empty(self):

        return self.items == []

    def push(self, item):

        self.items.append(item)

    def pop(self):

        return self.items.pop()

    def peek(self):

        return self.items[len(self.items) - 1]

    def size(self):

        return len(self.items)

    def imprimir(self):

        return print(self.items)

p.push(“a”)

p.imprimir()

p.push(“b”)

p.imprimir()

p.push(“c”)

p.imprimir()

p.push(p.peek())

p.imprimir()

p.push(“d”)

p.imprimir()

p.pop()

p.imprimir()

p.pop()