João Nuno Roque Costa 99 088

1) a) <frase> ::= r <char> c

<char> ::= <ad> | <ad> <char>

<ad> ::= a | d

b)
```
def reconhece(frase):
a=0
    if not type(frase) == string:
        raise ValueError("argumento invalido")
    if frase[0] == "r" and frase[len(frase)-1] == "c"
        frase = list(frase)
        for i in range(1, len(frase)-1):
            if frase[i] not in "ad":
                return False

        return True
    return False
```

João André Rosa Costa 99088

2) a)
```
class Condutor:
    def __init__(self, carta, idade):
        self.id = carta
        self.anos = idade
        self.pontos = 0

    def id(self):
        return self.id
    def pontos(self):
        return self.pontos
    def anos(self):
        return self.anos

    def aumenta(self, val):
        self.pontos = self.pontos + val

    def sem_carta(self):
    if self.pontos >= 20:
        return "Condutor" + self.id + "perdeu a carta"
```

joão André Rego Cota 99088 J4

2.) b)

```
def simula(c, n, limite)
    condutores, pontos = [], 0
    while c != 0:
        c = c - 1
        condutores += [gen_id(),]
    while n != 0:
        for i in condutores:
            vel = randint(limite - 20, 2 * limite)
            if 0 < vel < limite * 1,1:
                print("condutor" + i + "não foi multado")
            elif limite * 1,1 ≤ vel < limite * 1,2:
                pontos = 2
            elif limite * 1,2 ≤ vel < limite * 1,3:
                pontos = 4
            elif limite * 1,3 ≤ vel < limite * 1,4
                pontos = 6
            else:
                pontos = 10
            if pontos != 0:
                "Condutor" + i + "circulava a" + vel + "Km/h"
                ", multado com" + pontos + "pontos"
```

João André Roque Costa 99088 js

3) def algum_lista(lst, pred):
    return True if len(filtra(pred, lst)) >= 1 else False

João André Roque Costa 99088 js

4) cria_lista_raizes(val):
    res = []
    while val > 2:
        res += [val,]
        val = sqrt(val)
    return res

João Andre Rego Costa 19 088 Jr

5) a)
```
def codifica(frase):
    pares, impares = [], []
    frase = list(frase)
    for i in range(0, len(frase)):
        if i % 2 == 0
            pares = pares + frase[i]
        else:
            impares = impares + frase[i]
    return str(impares) + str(pares)
```

b)
```
def decodifica(frase):
    if len(frase) % 2 != 0:
        meio = len(frase) // 2 + 1
    else:
        meio = len(frase) // 2
    inicio = 0
    res = ''
    while len(res) != len(frase):
        res += frase[inicio]
        inicio += 1
        res += frase[meio]
        meio += 1
    return res
```

João André Rego Castro 99 088 jr

6) c)
```
def Soma_divisores_nao_primos(val):
    res=0
    for i in range(1, val):
        if Val % i == 0 and eh_primo(i) == False:
            res += i
    return res
```

b)
```
def Soma_divisores_nao_primos(val):

    def aux(val, res):
        if Val == 0
            return res
        if eh_primo(Val) == False:
            res += Val
        return aux(val-1, res)

    return aux(v
```

7) a) F
b) V
c) F
d) V
e) V

8)
```
def primo(n):
    val = 0
    for i in range(2, n):
        if n % i == 0
            val += i

    if val == n:
        return True
    else:
        return False
```

João André Rosa Costa 99088

9) a) Lista

b)
```
def nova_fila_tupla():
    return []

def primeiro(fila):
    if len(fila) >= 1:
        return fila[0]

def ultimo(fila):
    if len(fila) >= 1:
        return fila[len(fila)-1]

def comprimento(fila):
    return len(fila)

def entra_inicio_fila(fila, el):
    fila = [el,] + fila
    return fila

def entra_fim_fila(fila, el):
    fila = fila + [el,]
    return fila
```

João Pedro Rego Leite 99088

Continuação 9.)

```
def sai_inicio_fila(fila):
    fila = fila[1:]
    return fila

def sai_fim_fila(fila):
    fila = fila[:len(fila)-1]

def eh_fila_dupla(fila):
    return type(fila) == list

def eh_fila_vazia(fila):
    return eh_fila_dupla(fila) and len(fila) == 0
```

9.) c)

```
def junta_filas_duplas(fila1, fila2)
    for i in range(len(fila2)-1, 0, -1):
        fila1 = fila1 + [fila2[i],]
    return fila1
```

João André Rocha Coito 99 088 JR

10)
```
def conta_palavras(Frase):
    frase = list(frase)
    pal = ""
    dic = {}
    for i in range(0, len(frase)):
        if i != " ":
            pal += i
        else:
            if pal not in dic:
                dic[pal] = 1
            else
                dic[pal] += 1
            pal = ""
    return dic
```

João André Rocha Costa 99088 1ºA

11)
```
def junta_ordenadas(L1, L2):
    def aux(L1, L2)
        return L1[0] if L1[0] ≤ L2[0] else L2[0]

    def valida(L1, L2)
        if L1[0] ≤ L1[1] and L2[0] ≤ L2[1]
            return valida(L1[1:], L2[1:])
        else:
            raise ValueError("argumentos invalidos")

    if aux(L1, L2) in L1:
        return junta_ordenadas(L1[1:], L2) + [L1[0],]

    else:
        return junta_ordenadas(L1, L2[1:]) + [L2[0],]
```