

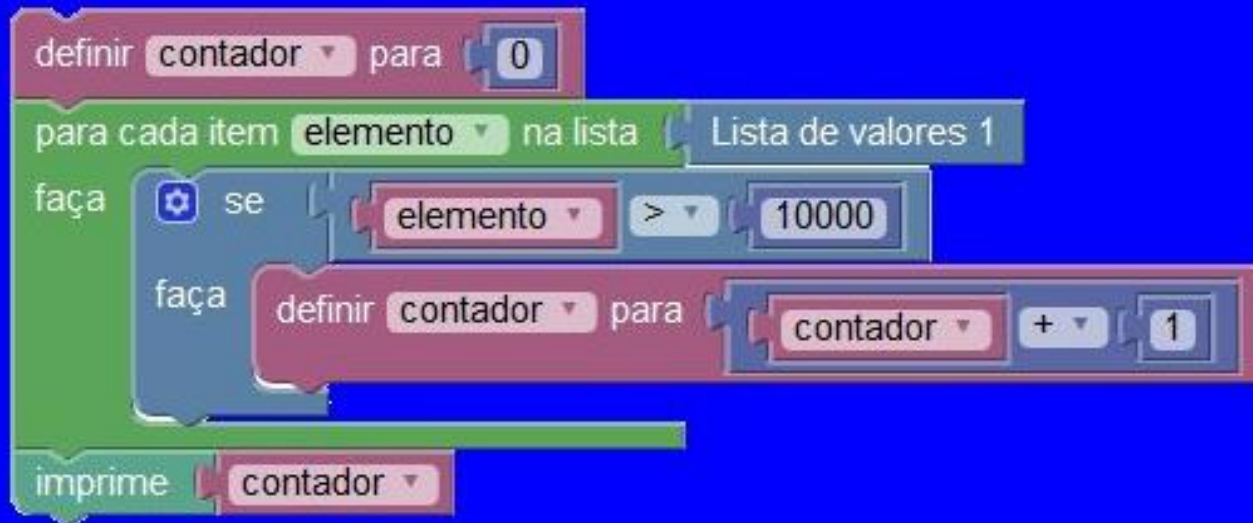
# Soluções dos Problemas



FCUL Rally Pro 2017

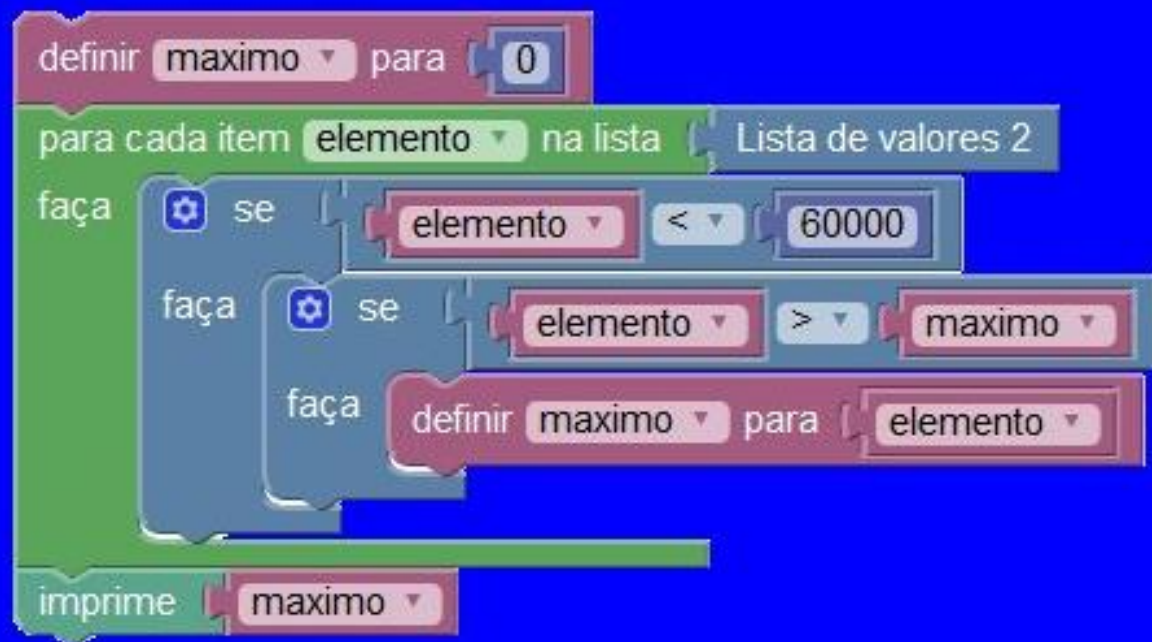
# 1) Números > 10000

É necessário percorrer a lista e contabilizar os números que satisfazem a condição pedida:



## 2) O maior elemento menor que 60000

É uma pesquisa pela lista à procura do máximo, mas filtrando números que não satisfazem a condição



### 3) Comparar elementos de duas listas

Pesquisar elemento a elemento nas duas listas (a variável *indice* como índice comum) e contabilizar apenas os pares de elementos que nos interessam.



## 4) Soma de múltiplos

Uma possibilidade passa por usar o resto da divisão inteira:



## 4) Soma de múltiplos

Queremos múltiplos de 3 que não são múltiplos de 9.

E queremos os primeiros 10000! Estes são:

3 6 9 12 15 18 21 24 27 30 33 36 39 42 45 48 51...

É importante detectar padrões para escrever algoritmos:

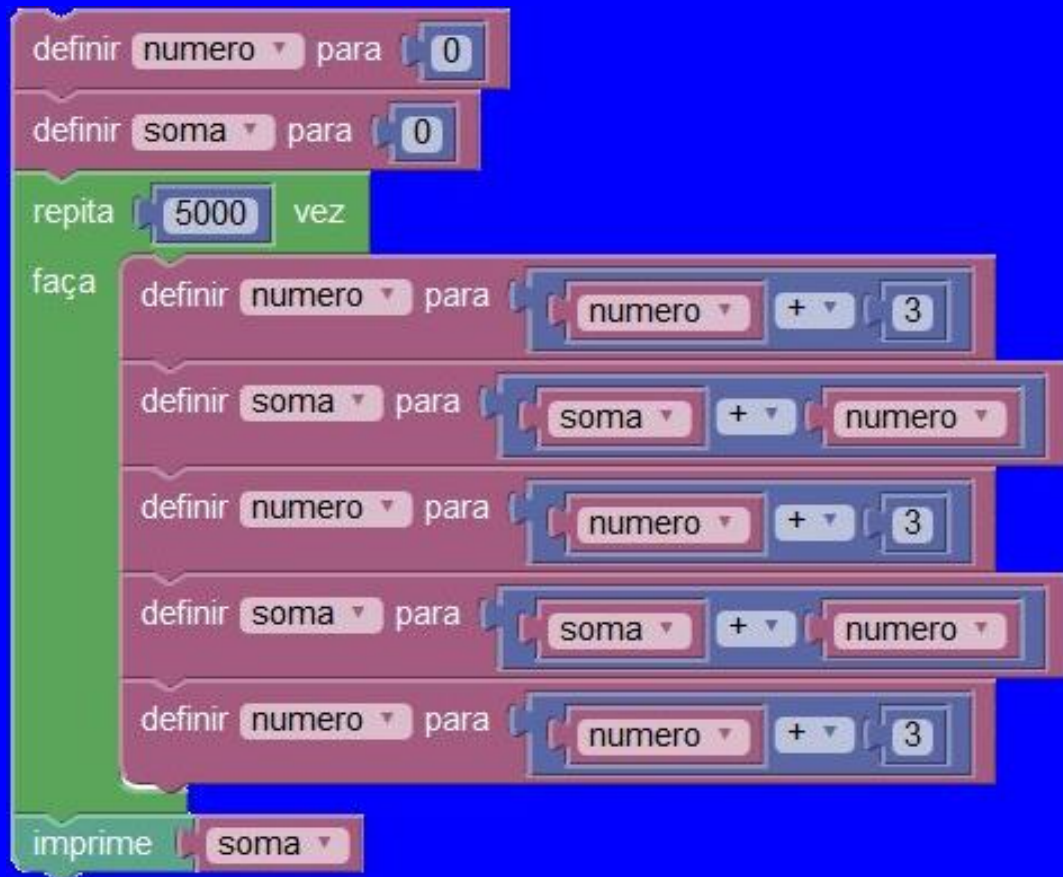
os múltiplos de 9 surgem de três em três na lista dos múltiplos de três

Um método: gerar 5000 triplos (15000 números dos quais escolhemos 10000)

## 4) Soma de múltiplos

O que resulta em um algoritmo como este:

Criamos triplos e um deles não é somado.



## 4) Soma de múltiplos

Outra possibilidade é fazer umas 'contas' na sequência anterior:

$$3 \ 6 \ 9 \ 12 \ 15 \ 18 \ 21 \ 24 \ 27 \ 30 \ 33 \ 36 \ 39 \ 42 \ 45 \ 48 \ 51 \dots =$$

$$3 * (1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 \ 8 \ 9 \ 10 \ 11 \ 12 \ 13 \ 14 \ 15 \ 16 \ 17 \dots)$$

ou seja, podemos somar os números de 1 a 15000 e subtrair os múltiplos de 3  
depois é só multiplicar por 3

$$3 \left( \sum_{i=1}^{15000} i - 3 \sum_{i=1}^{5000} i \right) = 3 \left( \frac{15000 \times 15001}{2} - 3 \frac{5000 \times 5001}{2} \right)$$

Usando progressões aritméticas substituímos o algoritmo por uma expressão!



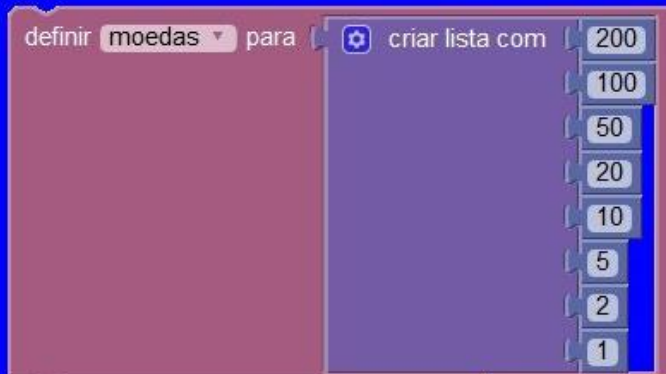
## 5) Primos entre 1 e 100000

Este problema poderia ser realizado implementando o algoritmo para verificar se um dado número é primo.

No entanto, um programador tem de conhecer as ferramentas disponíveis!

Puzzle: como duplicar a velocidade deste ciclo mudando alguns valores?





## 6) Trocos

O truque aqui é começar por subtrair as moedas maiores, e quando o montante é menor que a moeda atual, passar para a moeda seguinte.

Este é um exemplo de algoritmo ganancioso



## 7) Jogo de Bachet

Este problema requer guardar as soluções intermédias para acelerar o cálculo. Usamos uma lista para esse efeito.

Esta técnica é conhecida por programação dinâmica.

