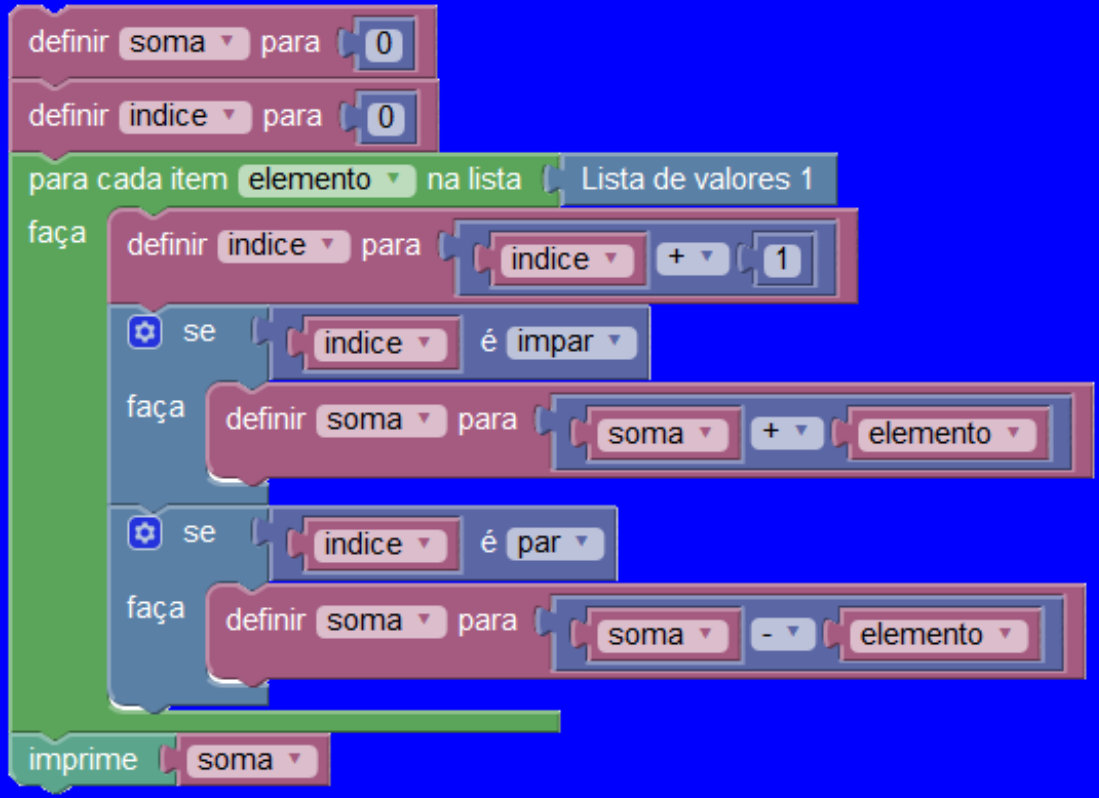
Soluções dos Problemas

FCUL Rally Pro 2018

1) somar e subtrair alternadamente

Percorrer a lista e saber se estamos num índice ímpar ou par



2) quantos números maiores que o anterior

Em cada ciclo mantemos dois números para determinar se a condição é verificada

3) Pares de somas pares

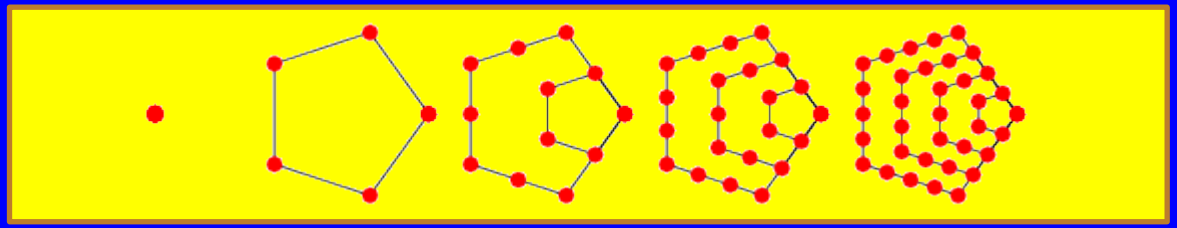
Pesquisar elemento a elemento nas duas listas (a variável *indice* como índice comum) e contabilizar apenas os pares de elementos que nos interessam.

4) Triplos Pitagóricos

Usamos três ciclos tendo a atenção que as guardas dos ciclos ajudem a satisfazer a condição a<b<c, tornando o algoritmo mais eficiente

5) Números pentagonais

A dificuldade deste problema era perceber qual a expressão numérica que produz o n-ésimo número pentagonal P_n

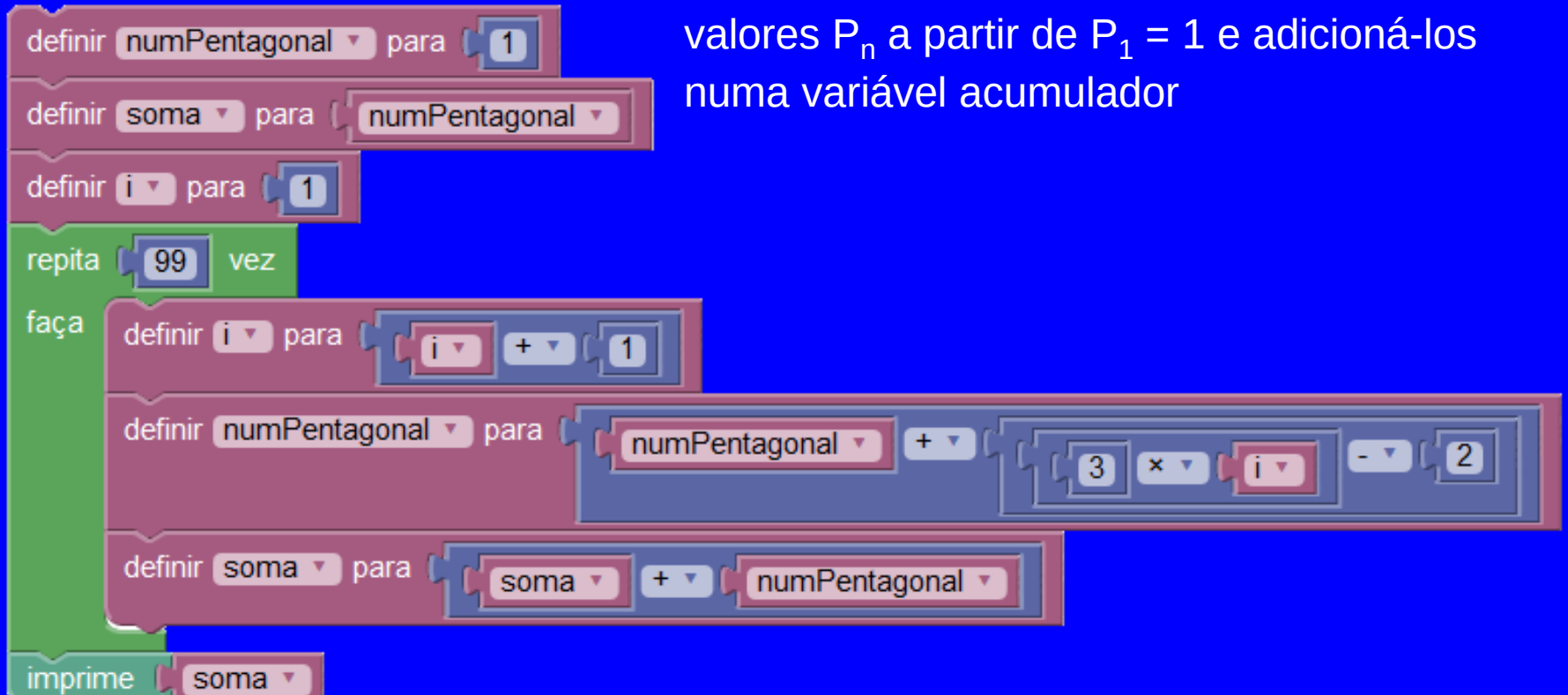


A figura correspondente a um número pentagonal adiciona 3 novas arestas à figura anterior.

Se for o i-ésimo pentágono, essas três arestas têm i pontos cada, mas dois desses pontos são vértices partilhados por duas arestas, logo têm-se de descontar, ie, $P_i = P_{i-1} + 3i - 2$

5) Números pentagonais

Tendo a expressão, é preciso gerar os 100 valores P_n a partir de $P_1 = 1$ e adicioná-los numa variável acumulador



6) Ladrilhos

Este problema tem a ver com a representação ternária de números.

Se os blocos fossem potências de 10, seria fácil ver que para ladrilhar, por exemplo, 123, seriam precisos 6 ladrilhos, um de 100, dois de 10, três de 1.

Como não temos um conversor de décimal para ternário, podemos usar uma abordagem egoísta (greedy algorithm) e ir ladrilhando com as maiores potências, subtraindo a distância que sobra até que esta seja zero.

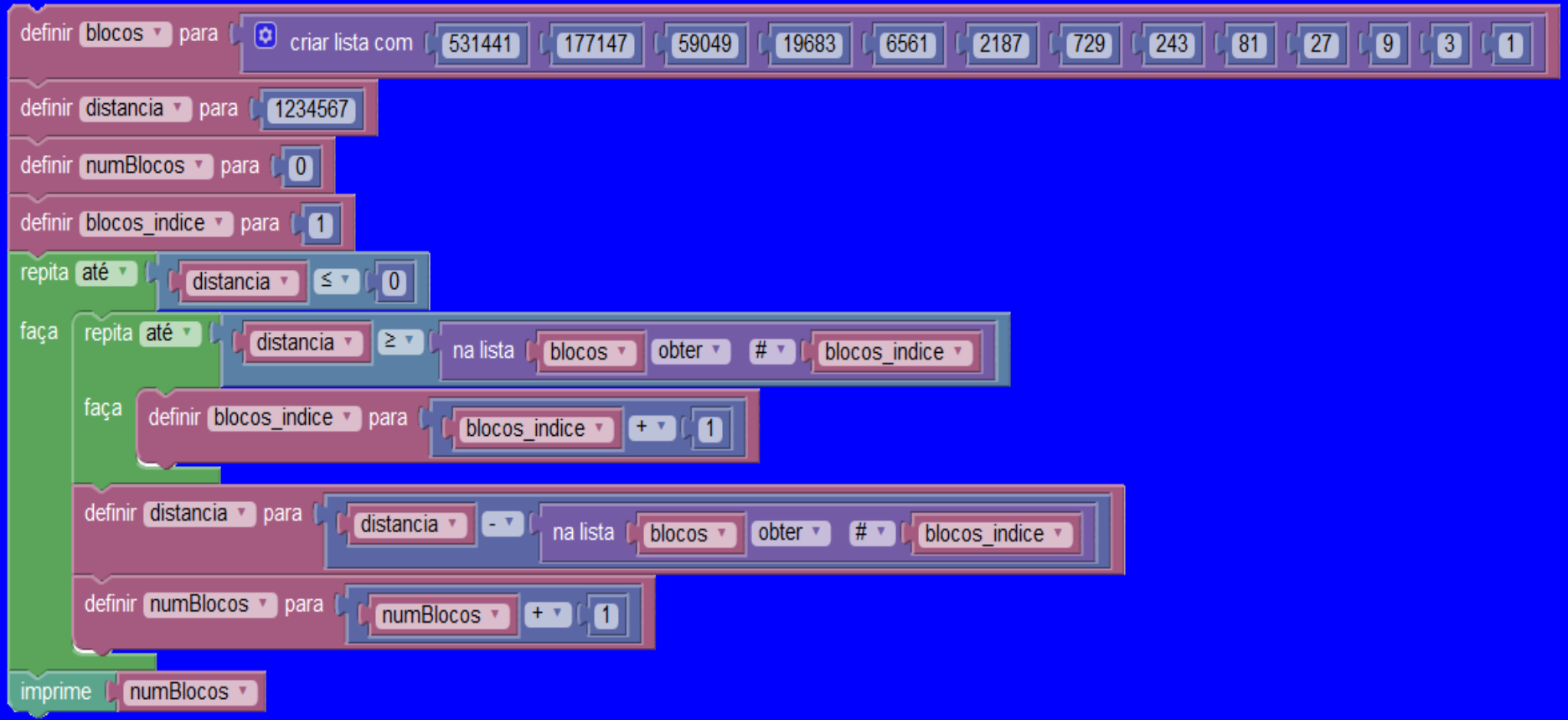
Por exemplo, para 123 seria:

- $123 - 1x3^4 = 123 - 81 = 42$
- $42 - 1x3^3 = 42 - 27 = 15$
- $15 - 1x3^2 = 15 - 9 = 6$
- $6 - 2x3^1 = 0$

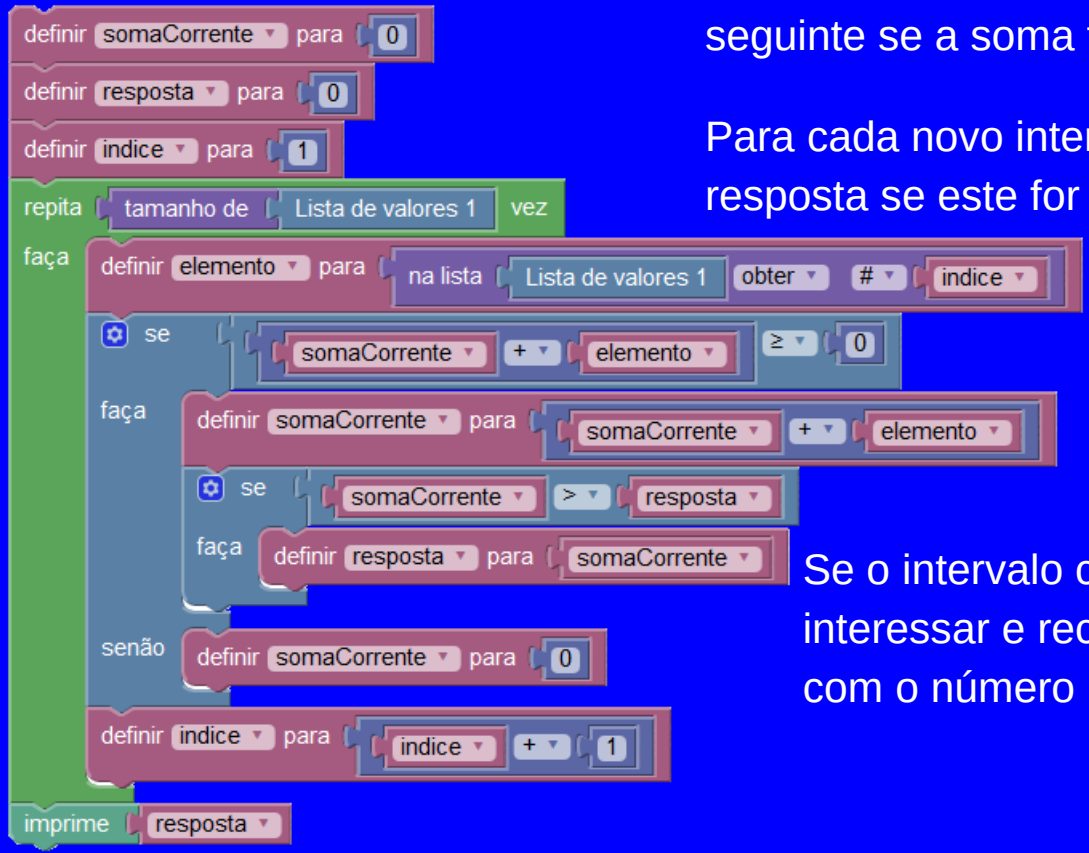
Ou seja, $123_{10} = 11120_3$

precisaríamos de 5 ladrilhos

6) Ladrilhos


definir blocos para criar lista com 531441 177147 59049 19683 6561 2187 729 243 81 27 9 3 1
definir distancia para 1234567
definir numBlocos para 0
definir blocos_indice para 1
repita até distancia ≤ 0
faça
repita até distancia ≥ na lista blocos obter # blocos_indice
faça
definir blocos_indice para blocos_indice + 1
definir distancia para distancia - na lista blocos obter # blocos_indice
definir numBlocos para numBlocos + 1
imprime numBlocos

7) Créditos e débitos



A técnica é ir guardando, no intervalo corrente, o número seguinte se a soma total não for zero.

Para cada novo intervalo corrente, guardamo-lo como resposta se este for o maior até agora encontrado.

Se o intervalo corrente ficar negativo, deixa de nos interessar e recomeçamos um novo intervalo corrente com o número positivo seguinte.