Maria, as maçãs e o LOGO

Ao desafio lançado no n.º 2 de Educação e Matemática — escrever um programa em LOGO para resolver o problema «Quantas maçãs tinha a Maria?» — responderam três leitores. Apresentamos duas das respostas e publicamos na íntegra a resposta mais completa, de João Palma, da Escola Secundária de Sampaio, em Sesimbra.

1. Fernando Nunes, da Escola Preparatória da Brandoa, começa por construir um procedimento para o cálculo de potências, de que irá necessitar:

```
to potencia :base :exp
make "pot 1
repeat :exp [make "pot :pot * :base]
output :pot
end
```

Depois constrói um procedimento para calcular as maçãs que Maria tinha no ínicio, considerando como inputs o número de amigos (:amigos) e maçãs com que Maria fica no fim (:sobra):

```
to maçãs_maria :amigos :sobra
make "maças (potência 2 :amigos) * (:sobra + 1)
-1
pr :maçãs
end
```

Como podem verificar, estes procedimentos resolvem o problema no caso considerado, em que apenas podem variar o número de amigos e as maçãs com que Maria fica no fim. O interesse desta solução de Fernando Nunes consiste em mostrar como, mesmo não usando recursões, é possível obter resultados em LOGO. O emprego da recursão pode, em muitos casos, simplificar a escrita de programas. Por exemplo, usando a definição recursiva de potência ($a^0 = 1$, $a^n = a * a^{n-1}$), o procedimento de potência escreve-se assim:

```
to potencia :base :exp
if :exp = 0 [op 1]
op :base * potencia :base :exp -1
end
```

Posteriormente, F. Nunes enviou-nos uma outra solução para o problema, agora resolvendo o caso geral com um único procedimento:

```
to maças_maria :amigos :fraccao :pconst :sobra
repeat :amigos [make "sobra ( :sobra + :pconst)
/ (1_ :fraccao)]
pr :sobra
end
```

2. Helena Paradinha, da Escola Secundária da Falagueira, enviou-nos também uma solução para o caso geral, utilizando um procedimento recursivo. Sendo :NA o número de amigos, :PM a parte das maçãs que dá a cada amigo, :AM o acréscimo constante e :RM as maçãs que sobram a Maria no fim, a solução proposta é:

```
TO CALC.M :NA :PM :AM :RM
IF :NA = 0 [OP :RM]
OP ((CALC.M :NA-1 :PM :AM :RM) + :AM) /
(1 — :PM)
END
```

3. Segue-se agora a resposta que J. Palma nos enviou:

Uma generalização:

«Maria tinha um cesto de maçãs. Encontrou um amigo e deu-lhe metade das maçãs que levava e mais meia maçã. Encontrou outro amigo e deu-lhe metade das maçãs que restavam e mais meia maçã. Depois encontrou outro amigo...

Depois ficou sem maçãs.

Quantas maçãs tinha a Maria?»

Designando x o número inicial de maçãs,

ao encontrar o 1.º amigo, Maria deu-lhe $\frac{x+1}{2}$ e ficou com $\frac{x-1}{2}$

ao encontrar o 2.º amigo deu $\frac{x+1}{4}$ e ficou com $\frac{x-3}{4}$

ao encontrar o 3.º amigo deu $\frac{x+1}{8}$ e ficou com $\frac{x-7}{8}$

...

ao encontrar o n.º amigo deu $\frac{x+1}{2^n}$ e ficou com $\frac{x-S_n}{2^n}$

(S_n é uma sucessão que pode ser de definida recursivamente:

$S_1 = 1$

$S_n = S_{n-1} + 2 * (n-1)$)

Ficando a Maria sem maçãs ao fim de n amigos:

$$\frac{x-S_n}{2^n} = \Rightarrow x = S_n$$

(continua na pág. 28)

(continuação da pág. 26)

A solução LOGO:

```
TO P1 :N                :N designa o número de amigos
IF :N = 1 [OP 1]
OP 2 * (:N–1) + P1 :N–1
END
```

O problema geral

«A Maria encontra sucessivamente n amigos. Dá a cada um $\frac{1}{B}$ das maçãs que tem, mais A maçãs. No fim ficou com y maçãs.

Quantas maçãs tinha a Maria?»

Supondo que, quando Maria encontra o amigo de ordem k, tem x maçãs, entrega-lhe:

$$\frac{x}{B} + A \quad \text{maçãs}$$

designando por y1 o número de maçãs com que ela fica após o encontro:

$$y1 = x - (\frac{x}{B} + A)$$

ou seja:

$$x = \frac{B}{B-1} * (y1 + A).$$

A solução LOGO

```
TO P :A :B :N :Y
IF :N = 0 [PR :Y STOP]
P :A :B :N–1 (:Y + :A)* :B/(:B–1)
(PR [NR. MAÇAS APOS O AMIGO] :N [–] :Y)
END
```

Notas:

1. Obrigado aos nossos leitores por terem correspondido ao desafio. Esperamos a vossa colaboração em futuras edições do LOGO.MAT.

2. (Para os fanáticos do LOGO e outros interessados...). Todas as soluções apresentadas têm a particularidade de exigir, antes da construção do programa, a dedução de uma expressão que depois é programada. Será possível proceder de outra forma mais natural em programação, que é partir directamente do enunciado sem deduções laterais? A solução seguinte parece responder a esta questão:

```
to maria :inicio :a :b :amigos :resto
if :resto = maria 1 :inicio :a :b :amigos [pr (se
[no principio Maria tinha] :inicio_maças) stop]
maria :inicio + 1 :a :b :amigos :resto
end

to maria 1 :inicio :a :b :amigos
if :amigos = 0 [op :inicio]
op maria 1 (:inicio — (:inicio / :b) — :a) :a :b
:amigos –1
end
```

Como podem verificar, no entanto, estes procedimentos resolvem apenas seguramente os casos em que a solução é inteira, pois sendo a pesquisa feita somando de cada vez 1 a :inicio (inicialmente 0), os valores não inteiros não são apanhados, e os procedimentos não param...! Basta ensaiar os casos:

maria 0.5 2 3 0 resultado 7

maria 0.7 3 4 2 a execução não pára, o resultado correcto é 18.6562, como se pode ver, por exemplo, com o procedimento de João Palma.

P. 7 3 4 2

Ficha técnica

Educação e Matemática n.º 3
Data: Julho de 1987
Composição, montagem e fotografia executadas e oferecidas pela Texto Editora
Impressão: Costa e Valério
Tiragem: 1500 exemplares

Correspondência: Henrique M. Guimarães ou Paulo Abrantes
Faculdade de Ciências - Departamento de Educação
Av. 24 de Julho, 134, 4.º
1300 LISBOA