



Unidade 9: Módulo Turtle em TI-Python

Lição 3: Resolução de um problema

Nesta terceira lição irá escrever um programa que lhe permita usar o módulo Turtle do TI-Python para ilustrar a resolução de um problema simples de matemática.

Objetivos:

- Resolver um qualquer triângulo
- Desenhar uma representação geométrica

0. Introdução

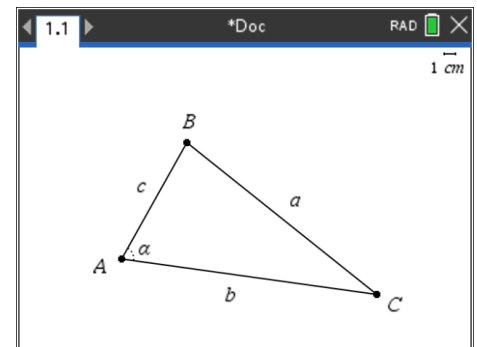
O que propomos é resolver um triângulo escaleno utilizando as relações do matemático Al Kashi.

Este resultado é assim designado em homenagem a **al-Kâshî**, matemático persa do início do século XV.

Estas igualdades são, também, conhecidas por "**Lei dos Cossenos**", e generalizam o **Teorema da Pitágoras** para triângulos não retângulos.

O **teorema de Al-Kâshî** afirma:

sendo [ABC] um qualquer triângulo e α a medida em radianos do ângulo BAC, então $\overline{AB}^2 + \overline{AC}^2 = \overline{BC}^2 + 2 \times \overline{AB} \times \overline{AC} \times \cos(\alpha)$.



Para simplificar, tendo em atenção a notação constante na figura ao lado, podemos afirmar que, para cada um dos lados do triângulo, temos as seguintes igualdades:

$$a^2 = b^2 + c^2 - 2b \times c \times \cos(\widehat{BAC})$$

$$b^2 = a^2 + c^2 - 2a \times c \times \cos(\widehat{ABC})$$

$$c^2 = a^2 + b^2 - 2a \times b \times \cos(\widehat{ACB})$$

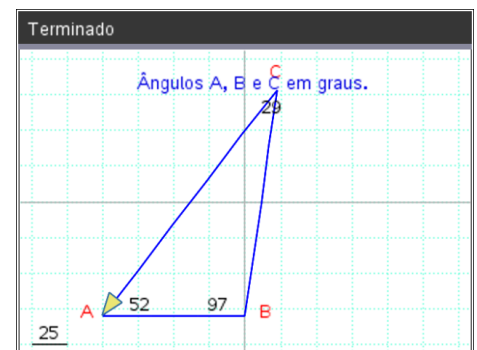
1. O problema

Suponha que são conhecidas as medidas dos três lados de um dado triângulo.

Escreva um programa, com recurso ao módulo Turtle do TI-Python, que:

- apresente a representação do triângulo;
- obtenha e apresente as medidas dos ângulos internos, em graus.

A medida do comprimento dos lados será dada em pixéis.



SUGESTÃO:

Tome em atenção que as medidas propostas deverão satisfazer a desigualdade triangular, isto é, o comprimento de um qualquer lado deve ser inferior à soma das medidas dos outros dois lados.

A gestão desta premissa poderá ser contemplada no próprio programa, surgindo uma mensagem de erro quando ela não for cumprida.

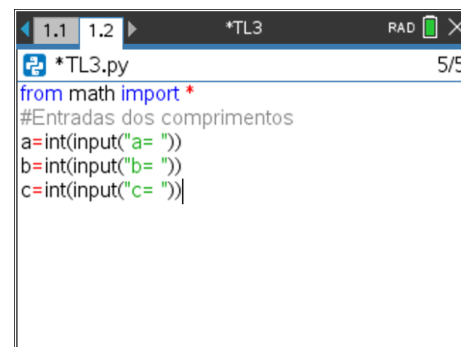


2. Criar um novo programa em TI-Python

Na aplicação TI-Python:

- crie um novo programa e designe-o por TL3;
- importe, para já, apenas o módulo Matemática;
- crie as variáveis a, b e c correspondentes aos comprimentos, em pixéis, dos lados do triângulo.

Os comprimentos dos lados serão entradas inteiras do nosso programa, conforme podemos observar no ecrã ao lado.



```
*TL3.py 5/5
from math import *
#Entradas dos comprimentos
a=int(input("a= "))
b=int(input("b= "))
c=int(input("c= "))
```

OBSERVAÇÃO:

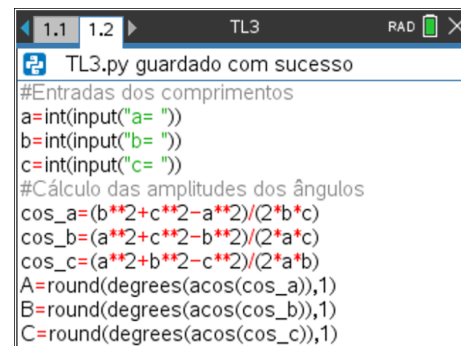
Os comprimentos dos lados do triângulo podem ser inseridos como argumentos de uma função em Python, o que pode ser interessante, mas não nesta lição porque se pretende a representação geométrica do triângulo mantendo visível a grelha do módulo Turtle.

Note que, ao usar uma função dentro de um programa em que se utilizam instruções do módulo Turtle, será removida do ecrã a grelha e rapidamente surgirá a saída >>> no Shell do Python.

3. Inserir as instruções para calcular as amplitudes dos ângulos internos

De seguida, usando a “Lei dos cossenos”, escreva as linhas de código para:

- calcular o valor dos cossenos dos ângulos \widehat{BAC} , \widehat{ABC} e \widehat{ACB} , respetivamente **cos_a**, **cos_b**, **cos_c**.
- calcular, usando a função trigonométrica inversa **acos()**, os valores das amplitudes dos ângulos internos do triângulo, em graus e arredondados às décimas.

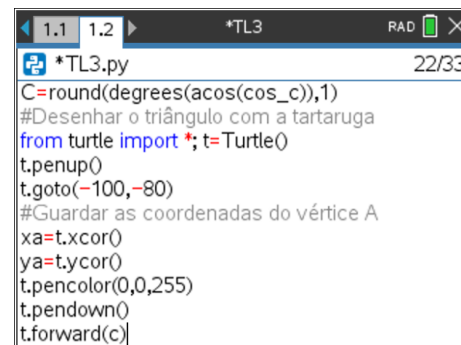


```
TL3.py guardado com sucesso
#Entradas dos comprimentos
a=int(input("a= "))
b=int(input("b= "))
c=int(input("c= "))
#Cálculo das amplitudes dos ângulos
cos_a=(b**2+c**2-a**2)/(2*b*c)
cos_b=(a**2+c**2-b**2)/(2*a*c)
cos_c=(a**2+b**2-c**2)/(2*a*b)
A=round(degrees(acos(cos_a)),1)
B=round(degrees(acos(cos_b)),1)
C=round(degrees(acos(cos_c)),1)
```

4. Inserir as instruções para desenhar o triângulo

Nesta etapa irão ser inseridas as linhas de código, do módulo Turtle, que permitam representar geometricamente o triângulo. Para isso, considere que:

- deve, agora, importar o módulo Turtle;
- o desenho do triângulo começará no ponto de coordenadas (-100,-80), em pixéis, isto é, a posição de um dos seus vértices;
- serão usadas as instruções **var=t.xcor()** e **var=t.ycor()** para mais facilmente serem inseridas as etiquetas nos vértices do triângulo. Estas instruções são usadas para se guardar as coordenadas da tartaruga,



```
*TL3.py 22/33
C=round(degrees(acos(cos_c)),1)
#Desenhar o triângulo com a tartaruga
from turtle import *; t=Turtle()
t.penup()
t.goto(-100,-80)
#Guardar as coordenadas do vértice A
xa=t.xcor()
ya=t.ycor()
t.pencolor(0,0,255)
t.pendown()
t.forward(c)
```



num dado momento, e podem ser obtidas pressionando tecla **[Menu]**, depois **A: Mais módulos**, em seguida, **Turtle Graphics** e, finalmente, **6: Tell Turtle's state**; e **3: var=t.xcor**.

- a instrução **t.pencolor(0,0,255)** é usada para colocar a caneta em azul, usando-se o código RGB. Também pode usar-se a instrução **t.pencolor("azul")**.

```

1.1 1.2 *TL3 RAD 33/33
*TL3.py
#Guardar as coordenadas do vértice B
xb=t.xcor()
yb=t.ycor()
t.left(180-B)
t.forward(a)
#Guardar as coordenadas do vértice C
xc=t.xcor()
yc=t.ycor()
t.left(180-C)
t.forward(b)
t.left(180-A)

```

5. Inserir as instruções para colocar as etiquetas com os nomes dos vértices

A última parte desta lição consiste em inserir as instruções que permitem exibir os nomes dos vértices do triângulo, bem como os valores das amplitudes dos seus ângulos internos. Para isso:

- levantar a caneta inserindo a instrução **t.penup()**;
- ativar a cor vermelho da caneta para a escrita, inserindo-se a instrução **t.pencolor("red")**;
- colocar o cursor, a tartaruga, junto de cada vértice usando a instrução **t.goto(x,y)** e as variáveis onde se guardaram as coordenadas;
- utilizar a instrução **t.write("text")** para escrever a designação do vértice;
- utilizar a instrução **t.write(Var)** para escrever a amplitude do ângulo interno correspondente a cada vértice.

```

1.1 1.2 1.3 *TL3 RAD 43/53
*TL3.py
#Inserir as etiquetas e amplitudes dos vértices
t.penup()
t.pencolor("red")
t.goto(xa-20,ya-10)
t.write("A") #escrever a etiqueta
t.goto(xa+15,ya-5)
t.write(A) #escrever a amplitude
t.pencolor("red")
t.goto(xb+5,yb-10)
t.write("B")
t.goto(xb-30,yb-5)

```

```

1.1 1.2 1.3 *TL3 RAD 52/52
*TL3.py
t.write("B")
t.goto(xb-30,yb-5)
t.write(B)
t.pencolor("red")
t.goto(xc-10,yc)
t.write("C")
t.goto(xc-15,yc-25)
t.write(C)
t.pencolor(0,0,225)
t.goto(-80,70)
t.write("Ângulos A, B e C em graus.")

```

Nota:

Note que no Python da TI-Nspire CX II, a instrução **t.write()** requer que a cor da caneta seja definida antes de cada utilização, pois por defeito ficará preto.

6. Executar o programa para resolver vários triângulos

Por fim, execute o programa agora criado para resolver triângulos para quaisquer valores de a, b e c, que verifiquem a desigualdade triangular.

Por exemplo para:

- a=100, b=150 e c=200
- a=160, b=250 e c=200

