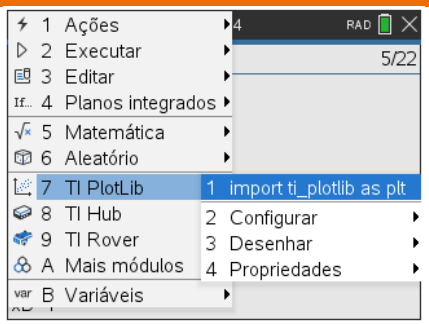
```
import ti_plotlib as plt
from math import *
xA=3
yA=7
xB=1
yB=-8
n=4
xv=xB-xA
yv=yB-yA
xx=[]
yy=[]
for i in range(n+1):
  di=i/n
  xx=xx+[xA+di*xv]
  yy=yy+[yA+di*yv]
plt.plot(xx,yy,"o")
```

1. Num segmento de reta definido pelas coordenadas de dois pontos distintos, como lá colocar pontos igualmente espaçados?

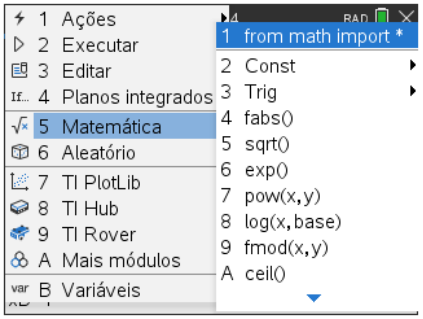


I. Pretende-se um programa que, depois de executado, apresente, a partir das coordenadas de dois pontos, uma representação gráfica de um segmento de reta com pontos igualmente espaçados.
Para que sejam reconhecidas as instruções a utilizar nas representações gráficas, é necessário importar, no início, a biblioteca *ti_plotlib*, o que pode ser feito escrevendo no teclado, se conhecer a sintaxe, ou no menu.

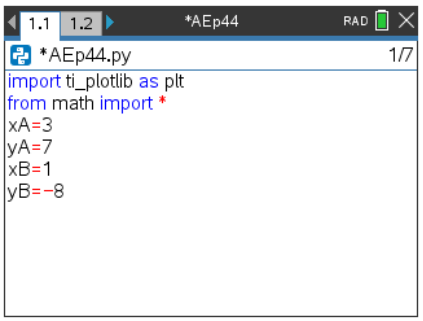
menu → 7 **TI PlotLib** → 1 **import ti_plotlib as plt**

Para além disto, note que no programa aparecem divisões e multiplicações. Mesmo que estas sejam operações matemáticas básicas, é uma boa prática importar também a biblioteca *math*.

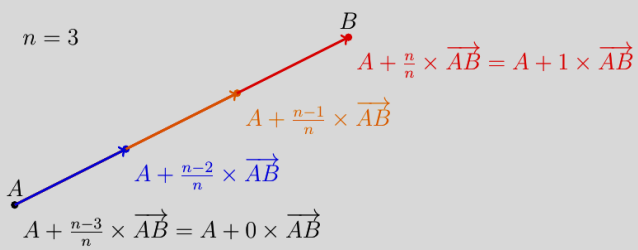
menu → 5 **Matemática** → 1 **from math import***



II. Dado que o segmento de reta final está definido por dois pontos então, após as bibliotecas importadas, no programa, tem de se indicar as coordenadas dos dois pontos. Assim, serão definidas as variáveis xA, yA, xB e yB que representam, respetivamente, a abcissa e a ordenada de cada um dos dois pontos extremos do segmento de reta. Para além disto, será definida a variável n que indicará o número de segmentos com que o inicial ficará dividido, a partir de $n-1$ pontos que irão ficar no interior do segmento de reta $[AB]$ à mesma distância uns dos outros, a mesma distância dos que estão mais próximos dos extremos A e B, para esses extremos.

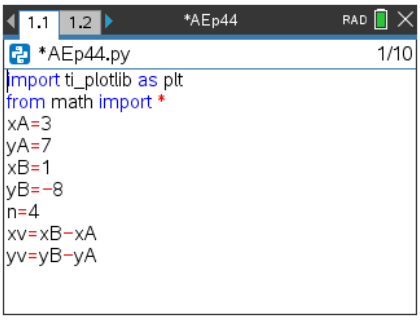


Note-.se que os pontos interiores do segmento de reta $[AB]$, vão ser gerados pela translação do ponto A de vetor $\overrightarrow{AB}$.
Como ilustrado na figura ao lado, com $n=3$, vão ser gerados dois novos pontos interiores de $[AB]$, onde o primeiro ponto é obtido pela translação do ponto A aplicado o vetor $\frac{n-2}{n}\times\overrightarrow{AB}=\frac{3-2}{3}\times\overrightarrow{AB}=\frac{1}{3}\times\overrightarrow{AB}$, o segundo ponto é obtido pela translação do ponto A de vetor $\frac{n-1}{n}\times\overrightarrow{AB}=\frac{3-1}{3}\times\overrightarrow{AB}=\frac{2}{3}\times\overrightarrow{AB}$.



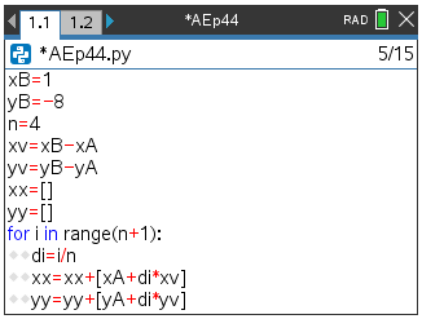
III. Para gerar os pontos interiores de $[AB]$ é necessário obter as coordenadas do vetor diretor $\overrightarrow{AB}$. Designando-se por xv e yv, respetivamente, a abcissa e a ordenada do vetor diretor, vamos acrescentar ao código as seguintes linhas:

```
xv=xB−xA
yv=yB−yA
```



IV. Devem agora criar-se duas listas vazias para armazenamento das coordenadas dos pontos A e B e dos pontos interiores, **xx** (abcissas) e **yy** (ordenadas), as quais vão sendo sucessivamente obtidas com um algoritmo num ciclo de repetição.

```
xx=[]
yy=[]
for i in range(n+1): # i varia de 0 ao número inteiro que antecede o do argumento.
    di=i/n
    xx=xx+[xA+di*xv]
    yy=yy+[yA+di*yv]
```

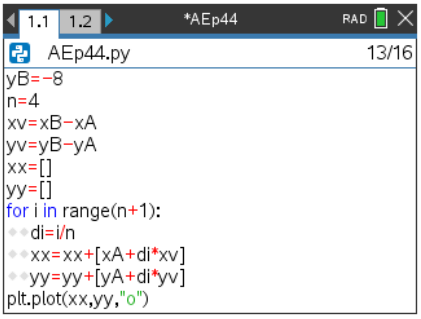


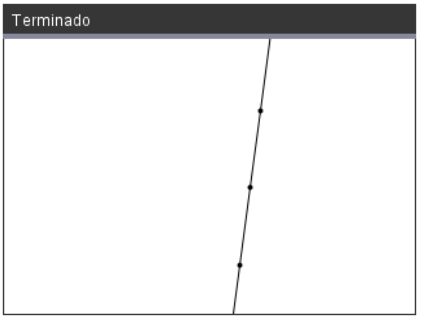
V. Para se concluir o programa é necessário colocar a instrução que permita, quando executado, mostrar a representação gráfica. Para tal, é necessário colocar a seguinte linha de código:

```
plt.plot(xx,yy,"o")
```

onde **xx** e **yy** são as listas que contêm as coordenadas de todos os pontos (extremos e interiores) e "o" refere-se à forma dos pontos a serem marcados. Os extremos do segmento de reta, nos limites do ecrã, não se observarão com a forma.
Para obter o código, pode escrever no teclado, se conhecer a sintaxe, ou selecionar no menu:

menu → 7 **TI PlotLib** → 3 **Desenhar** → 6 **plot(x,y,"mark")**





Nota: Para executar o programa pode utilizar-se uma instrução do menu (menu 2 1), mas é claramente mais simples utilizar um atalho, uma combinação de teclas (ctrl + R).

Algumas ideias sobre programação, relacionadas com o contexto