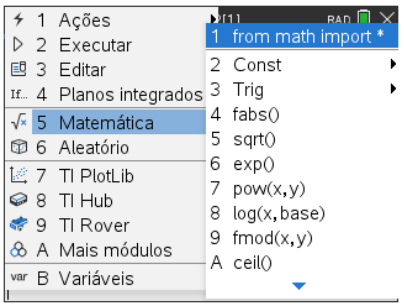
```
from math import *

def median(x):
  x.sort()
  n=len(x)
  if n%2==0:
    return (x[int(n/2)-1]+x[int(n/2)])/2
  else:
    return x[int((n)/2)]

n=int(input("Quantas disciplinas tens? "))
print("Escreve as tuas notas das disciplinas")
notas=[]
for i in range(n):
  x = int(input("?"))
  notas=notas+[x]
notas.sort()
soma=0
for i in range(n):
  soma=soma+notas[i]
  media=soma/n
desvios=0
for i in range(n):
  desvios = desvios + (notas[i]-media)**2
  desvio_padrao=round(sqrt(desvios/n),3)

print("Notas:",notas)
print('Desvio-padrão:', desvio_padrao)
print('Média:', media)
print("Mediana:",median(notas))
print("Amplitude:",max(notas)-min(notas))
```

1. Pedidas as classificações de n disciplinas de um aluno, como obter a média, a mediana, o desvio-padrão e a amplitude dessas classificações?

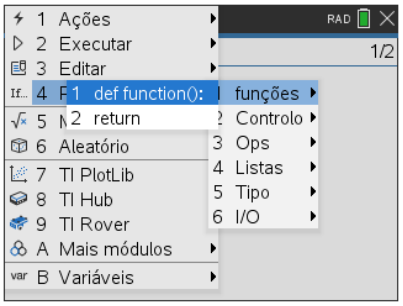
Pretende-se um programa que, depois de executado, apresente em relação às classificações das disciplinas de um aluno, indicadas pelo utilizador, a média, a mediana, o desvio-padrão e também a amplitude.



I. Antes de tudo, é preciso notar que vai ser preciso fazer operações matemáticas como, por exemplo, a determinação do valor da raiz quadrado de um número. Assim, é necessário importar o **módulo math**, logo no início do programa. Isso pode ser feito escrevendo no teclado, se conhecer a sintaxe, ou no menu.

B 5 Matemática → 1 from math import*

II. Na calculadora TI-Nspire CX II-T, o Python não tem a biblioteca *statistics* pelo que, não sendo possível, utilizar o comando *median.statistics,* enquanto este não for integrado na tecnologia TI-Nspire CX II-T, tem-se de construir uma função que, perante uma lista de valores, devolva a **mediana**. Assim, pode ser boa ideia começar o programa com a construção dessa função, a nomear como ***median.***
Esta "função" é semelhante ao que noutras linguagens de programação se chama de sub-rotina, ou seja, uma secção de código que é ativada a partir das linhas de código do programa subsequentes. Esta estrutura é bastante útil, por exemplo, quando no programa se precisa de fazer algumas vezes as ações que nela estão previstas.



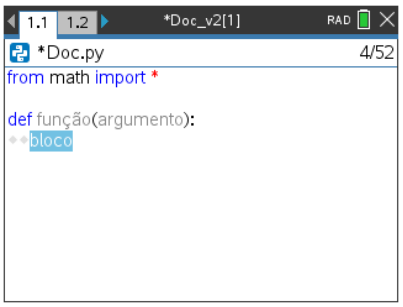
Para obter esta estrutura, caso conheça a sintaxe, poderá escrever com o teclado, mas pode também obter através do menu:

menu → 4 **Planos integrados** → 1 **funções** → 1 **def function():**

Obtida a estrutura, é agora necessário preencher os campos em falta.

Para o cálculo da mediana, é necessário começar por ordenar os dados, o que pode ser feito, sobre a lista, com o código **nome_da_lista.sort()**.
Tal como está o código, a ordenação é feita por ordem crescente. Embora, caso conheça a sintaxe, possa escrever com o teclado, poderá aceder ao comando através do menu:

menu → 4 **Planos integrados** → 4 **Listas** → E **.sort()**

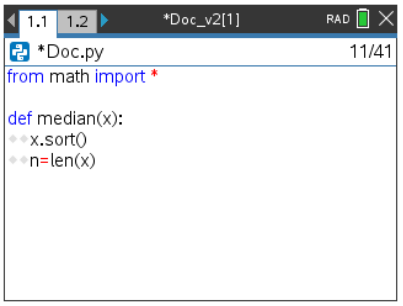


Para além disto, vai ser necessário saber quantos elementos tem a lista, havendo uma função em Python para tal, que é **len()**, a qual se pode obter no menu:

menu → 4 **Planos integrados** → 4 **Listas** → 3 **len()**

Assim, atribuindo à variável **n** o número de elementos da lista, temos as seguintes linhas de código:

```
def median(x):          # o argumento da função é uma lista x
    x.sort              # os elementos da lista são ordenados (crescente)
    n=len(x)            # n é o  número de elementos da lista
```



Nota: dada uma lista de n dados, ordenada, então a mediana é:

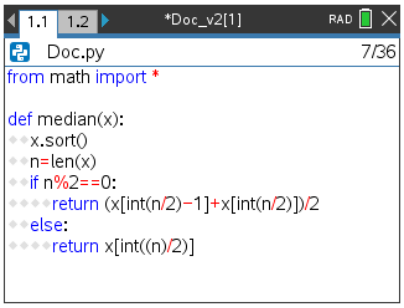
- $x_{\left(\frac{n}{2}+1\right)}$, ou seja, o elemento de ordem $\frac{n}{2}+1$ da lista, caso o número de elementos da lista seja par;
- $\frac{x_{\left(\frac{n}{2}\right)}+x_{\left(\frac{n}{2}+1\right)}}{2}$, ou seja, a média dos elementos "centrais" da lista, caso o número de elementos da lista seja ímpar.

Assim, passando agora para linguagem *Python*, é necessário utilizar uma estrutura condicional, que pode escrever com o teclado ou obter no menu.

B → 4 **Planos integrados** → 2 **Controlo** → 3 **if..elif..else..**

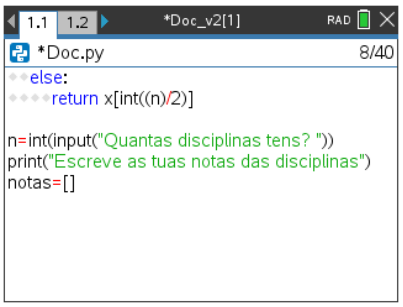
Tem-se, então a função completa:

```
def median(x):          # o argumento da função é uma lista x
    x.sort              # os elementos da lista são ordenados
    n=len(x)            # o número de elementos da lista é n
    if n%2==0:          # se n for par, ou seja, o resto da divisão por 2 for 0
        return (x[int(n/2)-1]+x[int(n/2)])/2    # retorna o valor
    else:               # caso contrário, ou seja, se n for ímpar
        return x[int(n)/2)]     # retorna o valor
```



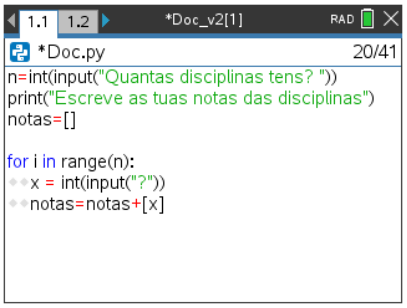
III. O programa "pede" ao utilizador para escrever o número de disciplinas. Este pedido será exibido no ecrã e, para além disso, o valor inteiro que o utilizador colocar será atribuído à variável **n**. Assim, a primeira linha de código a escrever-se deverá ser a escrita desse pedido, ou seja:

```
n=int(input("Quantas disciplinas tens?"))
```



IV. Para além do programa "pedir" o número de disciplinas, vai pedir ao utilizador que indique as respetivas classificações. Então, ele vai repetir **n** vezes o pedido de um número inteiro. Estes **n** dados serão armazenados numa lista que, inicialmente, está vazia e vai sendo preenchida à medida que o utilizador introduz os valores. Esta lista é aqui nomeada por **notas**.
Assim sendo, colocam-se as seguintes linhas de código:

```
print("Escreve as tuas notas das disciplinas?"))
notas = []              # a lista idades inicialmente vazia
for i in range(n):      # por cinco vezes:
    x=int(input("?"))   # pede um valor inteiro
    notas=notas+[x]     # armazena o valor inteiro na lista
```

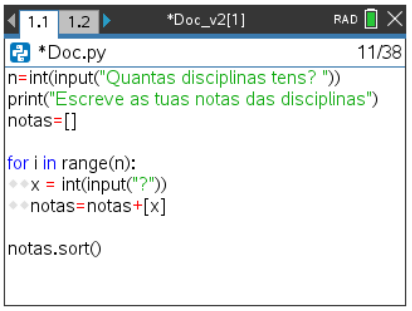


Para obter estas linhas de código pode utilizar o teclado, não esquecendo dos ":" e também da indentação "◆◆", para o que o ciclo de repetição **for** se aplique apenas ao que estiver identado. Pode também recorrer ao menu para obter as linhas de código, a completar com os elementos específicos.

menu → 4 **Planos integrados** → 2 **Controlo** → 4 **for index in range(size):**

V. Para obter a mediana é necessário ordenar os dados, com **nome_da_lista.sort()**.
Pode-se encontrar este comando em:

menu → 4 **Planos integrados** → 4 **Listas** → E **.sort()**



VI. Para obter a média, um procedimento possível passa por determinar a soma das notas, dividida pelo número de elementos da lista "notas". A soma pode ser obtida quando alista estiver completa ou, desde logo, à medida que os valores são introduzidos, serem automaticamente adicionados à soma dos anteriores e colocado esse valor numa variável.
A opção que aqui se mostra, não sendo melhor ou pior, mas com o intuito de revelar outra abordagem, percorre os elementos da lista já criada e adiciona-os sucessivamente, registando a soma numa variável, designada aqui por **soma**.

```
soma=0                  # no início vale 0
soma=soma+notas[0]      # soma passa a valer o que valia a soma anterior,
                          adicionada de notas[0].
soma=soma+notas[1]      #soma passa a valer o que valia a soma anterior,
                          adicionada de notas[1].
```

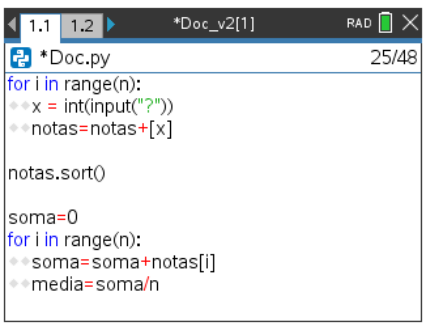
Este procedimento repete-se até:

```
soma=soma+notas[n-1]
```

Na verdade, o algoritmo pode ser traduzido pela utilização de **soma=soma+notas[i]**, desde que **i=0** até que **i=n-1**, que em código *Python* pode ser traduzido por:

```
for i in range(n):
    soma=soma+notas[i]
    media=soma/n
```

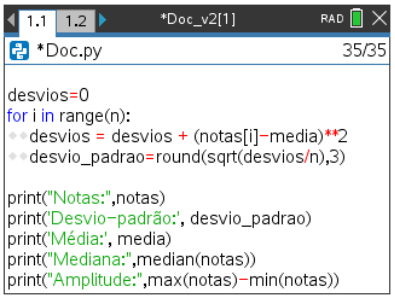
No código, como se pode visualizar, é acrescentado a última linha que permite obter a média, ao se dividir a soma das notas, **soma**, pelo número total de disciplina, **n**.



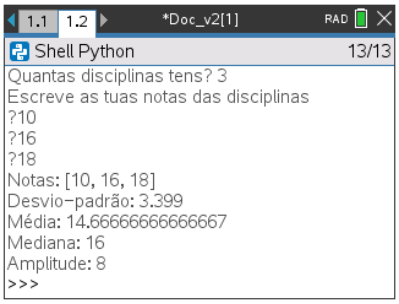
VII. Para o cálculo do desvio-padrão, o procedimento é análogo. Como se exige que saia o resultado do desvio-padrão com três casas decimais, utiliza-se o comando **round(x,n)**, onde x é o número decimal que é arredondado para o número especificado de dígitos.

VIII. Por fim, de modo a imprimir-se os valores pretendidos após a execução, considera-se as seguintes linhas de código:

```
print("Notas:",notas)
print("Desvio-padrão:",desvio_padrao)
print("Média:",media)
print("Mediana:",median(notas))
print("Amplitude:",max(notas)-min(notas))
```



Como se pode observar, para imprimir a mediana, recorre-se à função **median**, criada no início. Para calcular a amplitude, é preciso obter a nota máxima e a nota mínima, isto recorrendo aos comandos **max** e **min**, respetivamente.



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).

Para voltar ao editor de *Python*, deve utilizar o touchpad, seja com o toque do dedo e o botão central ( ) depois de sobrepor o cursor ao número da página, ou com as teclas laterais, premindo previamente ctrl., para abrir a página anterior ou posterior.

Algumas ideias sobre programação, relacionadas com o contexto