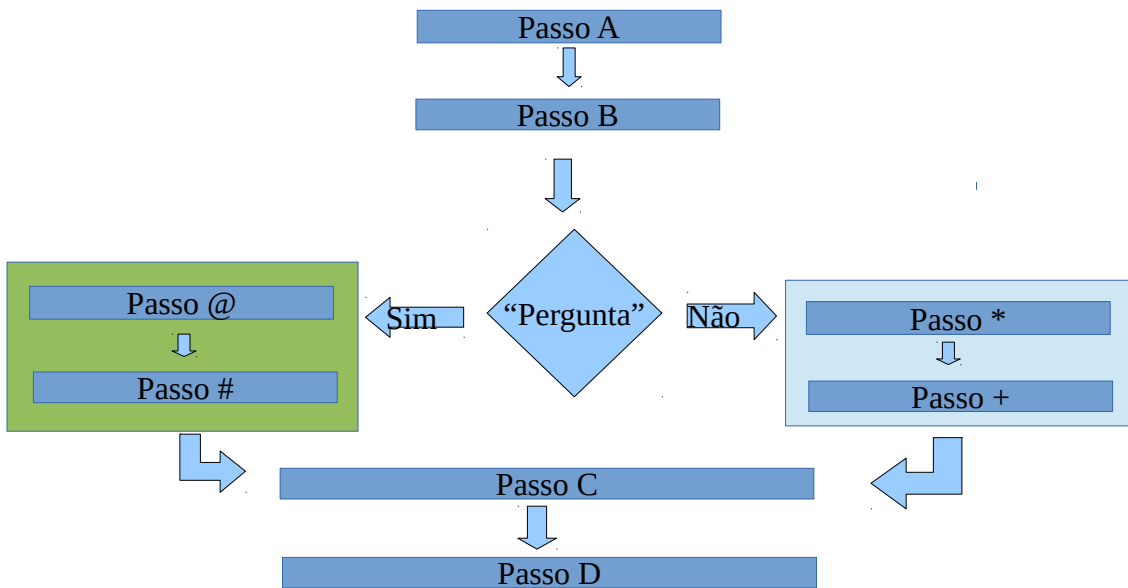


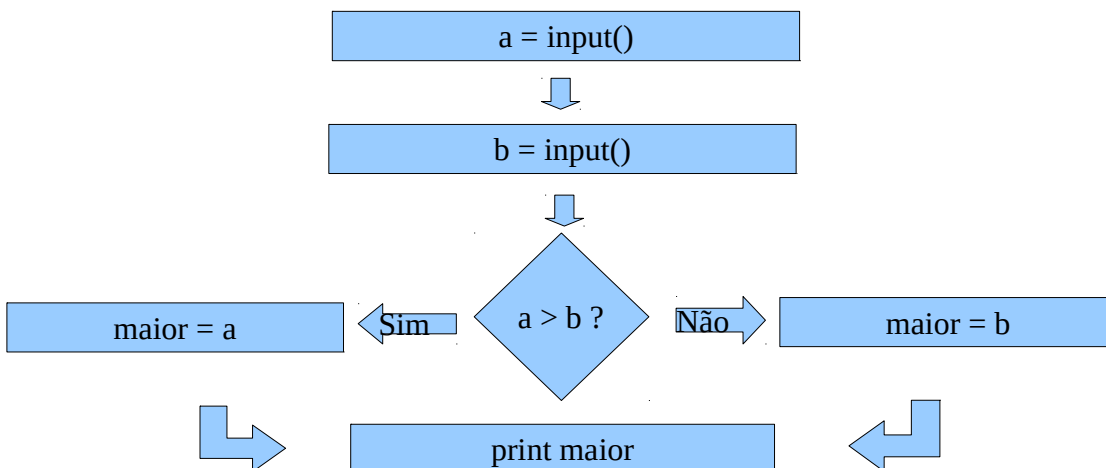
- Uma *condição* (um losango no fluxograma de um algoritmo) é descrita, em *Python*, com as palavras-chave **if** (*se*, em inglês) e **else** (*senão*, em inglês).
- O seguinte algoritmo é descrito, em *Python*, pelo programa a seguir.



```

Passo A
Passo B
if “pergunta”:
    Passo @
    Passo #
else:
    Passo *
    Passo +
Passo C
Passo D
  
```

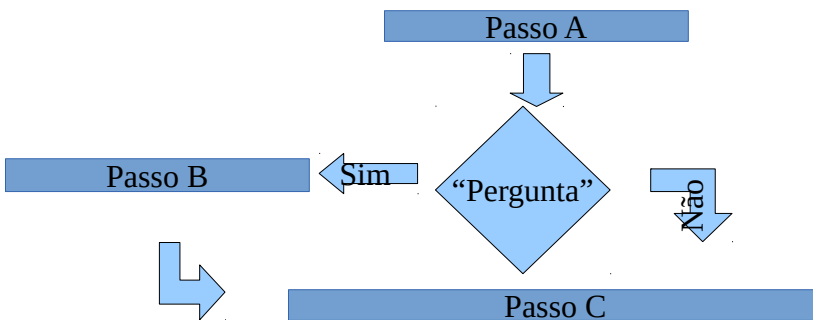
- Exemplo: Considere o seguinte problema: Dados dois números, determine o **maior deles**.
- O seguinte algoritmo resolve este problema:



- Este algoritmo é escrito da seguinte maneira em *Python*:

```
a = input()
b = input()
if a > b:
    maior = a
else:
    maior = b
print maior
```

- O uso de : (dois-pontos) após a “pergunta” (*expressão booleana*) e **else** é obrigatório;
- Os passos condicionados são escritos **com espaços em branco** antes deles.
  - Esses espaços formam a **identação** do programa. São necessários para definir quais passos são condicionados e quais não são.
    - Como exemplo, note que, se todos os passos fossem escritos sem espaços em branco, não seria possível dizer se *print maior* está condicionado ao **else** ou não.
  - O número de espaços em branco **deve ser igual** em todas as linhas.
    - É comum usar **dois** ou **quatro** espaços, ou **um TAB**.
- O **else** nem sempre é necessário. O algoritmo



é descrito, *em Python*, por:

```
Passo A
if “pergunta”:
    Passo B
Passo C
```

- O passo B é executado **apenas** se a resposta da expressão booleana (a “pergunta”) for *sim*.
  - O passo C **sempre** é executado.
- A expressão booleana (“pergunta”) pode conter, dentre outros, os operadores:

>	Maior	if (a > b):	“Se o valor de a é maior que o valor de b”
>=	Maior ou igual	if (a >= b):	“Se o valor de a é maior ou igual que o valor de b”
<	Menor	if (a < b):	“Se o valor de a é menor que o valor de b”
<=	Menor ou igual	if (a <= b):	“Se o valor de a é menor ou igual que o valor de b”
==	Igual	if (a == b):	“Se o valor de a é igual ao valor de b”
!=	Diferente	if (a != b):	“Se o valor a é diferente do valor de b”

- Note que **dois iguais** são usados em expressões booleanas, para o diferenciador do operador de atribuição.
  - **a = 4\*b + 2**
    - “armazene na variável 'a' o valor da variável 'b' multiplicado por 4, mais 2”
  - **if a == b:**
    - “Se o valor de a é igual ao valor de b”
- Exemplo: Considere o seguinte problema: Dados um número natural, diga *sim* se ele é par, e *nao* caso contrário.
  - Um número é par se e somente se o *resto da divisão* do número por 2 é igual a 0:

```
n = input()
if n % 2 == 0:
    print "sim"
else:
    print "nao"
```

- É possível combinar expressões booleanas com **and** (E) e **or** (OU), além de usar **not** (NÃO):

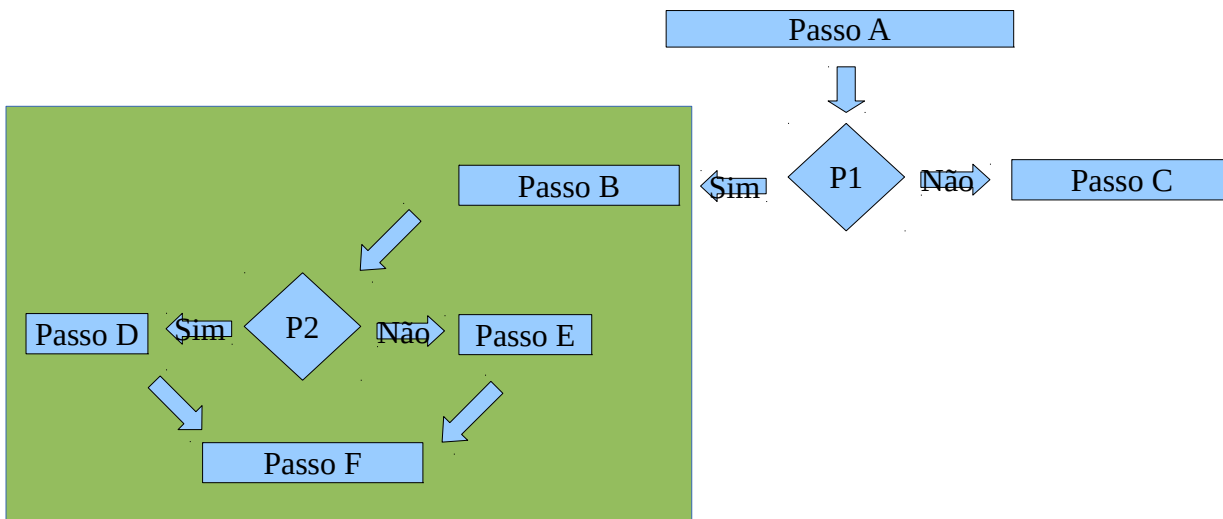
<b>and</b>	<b>if (exp1) and (exp2):</b>	<b>if (a &gt; 3) and (b != 2):</b>	“Se a é maior que 3 e b é diferente de 2”
<b>or</b>	<b>if (exp1) or (exp2):</b>	<b>if (a == 0) or (b == 1):</b>	“Se a é igual a 0 ou b é igual a 1”
<b>not</b>	<b>if not (exp):</b>	<b>if not (a != 2):</b>	“Se a <b>não</b> é diferente de 2”

(Obs: **if not (a != 2):** é logicamente equivalente a **if a==2:.**)

- Exemplo: Considere o seguinte problema: Dados um número, diga *sim* se ele está no intervalo [0,100], e *nao* caso contrário.

```
n = input()
if (0 <= n) and (n <= 100):
    print "sim"
else:
    print "nao"
```

- Condições podem ser **aninhadas** (colocadas uma dentro de outra):



Passo A

if P1:

Passo B

if P2:

Passo D

else:

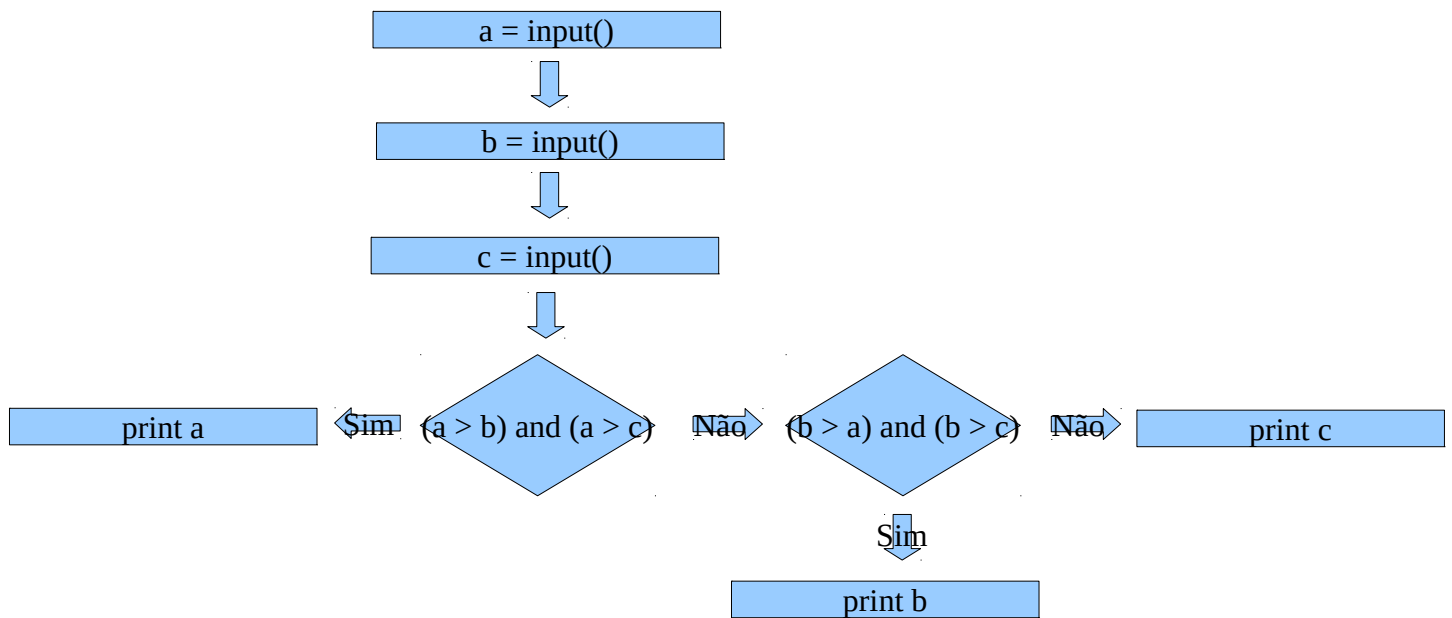
Passo E

Passo F

else:

Passo C

- Note que os passos D e E têm *o dobro* de espaços em braco (pois são condicionados duas vezes).
- Exemplo: Considere o seguinte problema: Dados três números *distintos*, determine qual deles é o maior.



```
a = input()
b = input()
c = input()
if (a > b) and (a > c):
    print a
else:
    if (b > a) and (b > c):
        print b
    else:
        print c
```