

MC102

Algoritmos e Programação de Computadores

Prova 1

Turmas A B C E F G H I

Primeiro Semestre de 2019

| Questão | Nota |
|---------|------|
| 1 | |
| 2 | |
| 3 | |
| 4 | |
| Total | |

Nome:

RA:

Importante: Esta avaliação é individual. Não é permitida a consulta a qualquer material impresso, nem o uso de celulares ou outros dispositivos eletrônicos de comunicação/computação. As respostas deverão ser escritas nos espaços demarcados, opcionalmente a lápis. Não se esqueça de indentar corretamente os códigos solicitados. **Boa prova!**

1. Após resolver as Atividades Conceituais na área MC102 no Moodle, Gabriela executou vários trechos de código para compreender melhor o funcionamento de alguns operadores e comandos. Em cada série, para cada trecho de código elaborado por ela, indique o que será escrito quando os programas forem executados. Caso o programa não escreva nada, indique “—”. Caso algum erro seja encontrado, indique o motivo e marque no código o ponto em que ele ocorre.

a) (1.0 ponto) **Tipos básicos**

```
# int ou float?
v = 2 + 3.0
print(v)
```

5.0

```
s1 = "A"
s2 = "B"
print(s1 + s2)
```

AB

```
v = "palavra"
v = 5
print(v)
```

5

```
a = "palavra"
b = 5
print(c)
```

'c' não definido

```
q = 10 // 3
r = 10 % 3
print(q, r)
```

3 1

```
q = 11 // 3
r = 11 - q * 3
print("r =", r)
```

r = 2

```
a = 4 * 3 - 2
b = 4 - 3 / 2 + 2
print(a, b, a - b)
```

10 4.5 5.5

Dicas: // executa divisão inteira e % obtém o resto da divisão

```
# True ou
# False?
print(6 < 1)
```

False

```
a = 5
b = 5
print("a == b")
```

a == b

```
a = True
b = False
print(a, not b, a and b or b)
```

True True False

b) (1.0 ponto) **Comandos condicionais**

| | | | | |
|---------------------------|---------------------------|----------------------------|---------------------------|-----------------------------|
| <code>a = 5</code> | <code>a = 10</code> | <code>a = 2</code> | <code>a = 5</code> | <code>a = 7</code> |
| <code>b = 3</code> | <code>b = 10</code> | <code>b = 2</code> | <code>b = 5</code> | <code>b = 7</code> |
| <code>if a > b:</code> | <code>if a > b:</code> | <code>if a > b:</code> | <code>if a > b:</code> | <code>if a > b:</code> |
| <code>print(a)</code> | <code>print(a)</code> | <code>print(a)</code> | <code>print(a)</code> | <code>print(a)</code> |
| <code>if b > a:</code> | <code>else:</code> | <code>if b >= a:</code> | <code>else:</code> | <code>elif b > a:</code> |
| <code>print(b)</code> | <code>print(b)</code> | <code>print(b)</code> | <code>print(b)</code> | <code>print(b)</code> |
| <code>else:</code> | <code>if a == b:</code> | <code>if a == b:</code> | <code>elif a == b:</code> | <code>else:</code> |
| <code>print(0)</code> | <code>print(0)</code> | <code>print(0)</code> | <code>print(0)</code> | <code>print(0)</code> |

| | | | | |
|----------------|-----------------|----------------|----------------------------|----------------|
| <code>5</code> | <code>10</code> | <code>2</code> | <code>elif: erro de</code> | <code>0</code> |
| <code>0</code> | <code>0</code> | <code>0</code> | <code>sintaxe</code> | |

c) (1.0 ponto) **Listas e comandos repetitivos**

```
lista = [0, 1, 4, 5, 12]
for i in range(len(lista)) :
    if lista[i] % 2 == 1 :
        lista[i] = lista[i] + 1
print(lista)
```

`[0, 2, 4, 6, 12]`

```
lista = [1, 3, 5, 7, 0]
i = 0
while lista[i] != 0 :
    lista[i] = lista[i] - 1
    i = i + 1
print(lista)
```

`[0, 2, 4, 6, 0]`

```
lista_a = [1, 3, 7, 4]
lista_b = [5, 2, 1, 6]
lista_c = []
for i in range(len(lista_a)) :
    if lista_a[i] > lista_b[i] :
        lista_c.append(lista_a[i])
    else:
        lista_c.append(lista_b[i])
print(lista_c)
```

`[5, 3, 7, 6]`

Dica: método `append()` adiciona um elemento ao final de uma lista

2. (1.5 ponto) Lembre-se da recomendação feita na tarefa de laboratório 03:

| | |
|------------------------------------|--|
| <code># NÃO!!!</code> | |
| <code>if CP > 0.5 :</code> | <code># SIM!!!</code> |
| <code>cp_suficiente = True</code> | <code>cp_suficiente = CP > 0.5</code> |
| <code>else:</code> | |
| <code>cp_suficiente = False</code> | |

Considerando que `a` e `b` são objetos que receberam algum valor inteiro, siga a recomendação e substitua os códigos abaixo pela atribuição de uma expressão à variável `r`. Você pode utilizar os operadores `not`, `and` e `or`.

```
if a > 5 :
    r = False
else:
    r = True
```

`r = not a > 5`

```
if a > 5 :
    r = True
elif b > 10 :
    r = True
else:
    r = False
```

`r = a > 5 or b > 10`

```
if a > 10 :
    if b < 25 :
        r = True
    else:
        r = False
else:
    r = False
```

`r = a > 10 and b < 25`

3. (2.5 pontos) João recebeu de seu pai algumas notas de R\$ 50,00 para fazer uma compra. Muito generoso, seu pai disse que ele receberia como gorjeta as notas e moedas do troco de valor inferior a R\$ 50,00. O programa abaixo lê o número de notas recebidas por João e o valor da compra. A saída indica o número de notas devolvidas para o pai e o valor da gorjeta.

```
notas_rec = int(input())
valor_compra = float(input())
troco_total = notas_rec * 50 - valor_compra
notas_dev = int(troco_total // 50)
gorjeta = troco_total % 50
print(notas_dev)
print("R$", format(gorjeta, ".2f"))
```

Note que quando o valor entregue pelo pai não é suficiente para a compra, João fica sem gorjeta. **Complete o programa abaixo** para que, nestes casos, João comente a situação e pergunte: “Pai, o dinheiro não foi suficiente... Posso ficar com R\$ 50,00?”. Seu programa deverá ler a resposta. Se o pai responder “SIM”, João irá ficar com uma das notas como gorjeta. Caso contrário, João continua sem nada, mas feliz por ter tentado. A saída deve continuar indicando, para todos os casos, o número de notas devolvidas para o pai e o valor da gorjeta de João. Não é necessário testar outras condições de erro ou fazer outras negociações com o pai.

```
notas_rec = int(input())
valor_compra = float(input())

if valor_compra > notas_rec * 50 :
    print("Pai, o dinheiro não foi suficiente... Posso ficar com R$ 50.00?")
    resp = input()
    if resp == "SIM":
        notas_dev = notas_rec - 1
        gorjeta = 50.00
    else:
        notas_dev = notas_rec
        gorjeta = 0.00
else:
    troco_total = notas_rec * 50 - valor_compra
    notas_dev = int(troco_total // 50)
    gorjeta = troco_total % 50

print(notas_dev)
print("R$", format(gorjeta, ".2f"))
```

4. (3.0 pontos) Considere um cenário de leilão semelhante ao proposto na tarefa de laboratório 06. A entrada é composta por um lance mínimo, por zero ou mais pares de linhas com um nome e um valor e pela palavra FIM, como no exemplo ao lado.

O leiloeiro não deverá ter paciência com clientes que ficam ofertando valores menores do que o lance mínimo. Ao ouvir um valor com estas características, deverá advertir o cliente pelo nome com a mensagem: ‘<nome_cliente>, por favor, emita apenas lances válidos!’ Ao final, na ausência de lances válidos, o leiloeiro deverá dizer Item não foi vendido. Caso contrário, ele deverá anunciar o valor do maior lance com a mensagem: Vendido por <oferta_maxima>.

Implemente um **programa completo** que leia os dados e tenha o comportamento descrito acima sem utilizar listas. Utilize o verso desta folha, se necessário.

| |
|--------|
| 50.00 |
| Paulo |
| 40.00 |
| João |
| 120.00 |
| Maria |
| 150.00 |
| FIM |

```
lance_minimo = float(input())
oferta_maxima = 0
nome = input()
while nome != "FIM":
    lance = float(input())
    if lance < lance_minimo :
        print(nome + ", por favor, emita apenas lances válidos!")
    else:
        if lance > oferta_maxima :
            oferta_maxima = lance
        nome = input()
if oferta_maxima == 0 :
    print("Item não foi vendido.")
else:
    print("Vendido por", oferta_maxima)
```

MC102

Algoritmos e Programação de Computadores

Prova 1

Turmas A B C E F G H I

Primeiro Semestre de 2019

| Questão | Nota |
|---------|------|
| 1 | |
| 2 | |
| 3 | |
| 4 | |
| Total | |

Nome:

RA:

Importante: Esta avaliação é individual. Não é permitida a consulta a qualquer material impresso, nem o uso de celulares ou outros dispositivos eletrônicos de comunicação/computação. As respostas deverão ser escritas nos espaços demarcados, opcionalmente a lápis. Não se esqueça de indentar corretamente os códigos solicitados. **Boa prova!**

1. Após resolver as Atividades Conceituais na área MC102 no Moodle, Gabriela executou vários trechos de código para compreender melhor o funcionamento de alguns operadores e comandos. Em cada série, para cada trecho de código elaborado por ela, indique o que será escrito quando os programas forem executados. Caso o programa não escreva nada, indique “—”. Caso algum erro seja encontrado, indique o motivo e marque no código o ponto em que ele ocorre.

a) (1.0 ponto) **Tipos básicos**

```
# int ou float?
v = 2 + 8.0
print(v)
```

10.0

```
s1 = "A"
s2 = "C"
print(s1 + s2)
```

AC

```
a = "palavra"
b = 5
print(c)
```

'c' não definido

```
v = "palavra"
v = 5
print(v)
```

5

```
q = 11 // 3
r = 11 % 3
print(q, r)
```

3 2

```
q = 10 // 3
r = 10 - q * 3
print("r =", r)
```

r = 1

```
a = 5 * 3 - 2
b = 4 - 5 / 2 + 2
print(a, b, a - b)
```

13 3.5 9.5

Dicas: // executa divisão inteira e % obtém o resto da divisão

```
# True ou
# False?
print(6 > 1)
```

True

```
a = 5
b = 5
print("a != b")
```

a != b

```
a = True
b = False
print(not a, b, a and b or a)
```

False False True

b) (1.0 ponto) **Comandos condicionais**

| | | | | |
|--|---|--|---|---|
| a = 12 b = 12 if a > b: print(a) else: print(b) if a == b: print(0) | a = 3 b = 3 if a > b: print(a) if b >= a: print(b) if a == b: print(0) | a = 5 b = 5 if a > b: print(a) else: print(b) elif a == b: print(0) | a = 7 b = 7 if a > b: print(a) elif b > a: print(b) else: print(0) | a = 5 b = 3 if a > b: print(a) if b > a: print(b) else: print(0) |
|--|---|--|---|---|

12
0

3
0

elif: erro de
sintaxe

0

5
0

c) (1.0 ponto) **Listas e comandos repetitivos**

```
lista = [0, 1, 4, 5, 12]
for i in range(len(lista)) :
    if lista[i] % 2 == 0 :
        lista[i] = lista[i] + 1
print(lista)
```

[1, 1, 5, 5, 13]

```
lista = [1, 3, 5, 7, 0]
i = 0
while lista[i] != 0 :
    lista[i] = lista[i] * 2
    i = i + 1
print(lista)
```

[2, 6, 10, 14, 0]

```
lista_a = [1, 3, 7, 4]
lista_b = [0, 2, 8, 6]
lista_c = []
for i in range(len(lista_a)) :
    if lista_a[i] > lista_b[i] :
        lista_c.append(lista_a[i])
    else:
        lista_c.append(lista_b[i])
print(lista_c)
```

[1, 3, 8, 6]

Dica: método `append()` adiciona um elemento ao final de uma lista

2. (1.5 ponto) Lembre-se da recomendação feita na tarefa de laboratório 03:

```
# NÃO!!!
if CP > 0.5 :
    cp_suficiente = True
else:
    cp_suficiente = False

# SIM!!!
cp_suficiente = CP > 0.5
```

Considerando que `a` e `b` são objetos que receberam algum valor inteiro, siga a recomendação e substitua os códigos abaixo pela atribuição de uma expressão à variável `r`. Você pode utilizar os operadores `not`, `and` e `or`.

```
if a < 5 :
    r = False
else:
    r = True
```

r = not a < 5

```
if a > 3 :
    r = True
elif b > 4 :
    r = True
else:
    r = False
```

r = a > 3 or b > 4

```
if a > 5 :
    if b > 10 :
        r = True
    else:
        r = False
else:
    r = False
```

r = a > 5 and b > 10

3. (2.5 pontos) João recebeu de seu pai algumas notas de R\$ 20,00 para fazer uma compra. Muito generoso, seu pai disse que ele receberia como gorjeta as notas e moedas do troco de valor inferior a R\$ 20,00. O programa abaixo lê o número de notas recebidas por João e o valor da compra. A saída indica o número de notas devolvidas para o pai e o valor da gorjeta.

```
notas_rec = int(input())
valor_compra = float(input())
troco_total = notas_rec * 20 - valor_compra
notas_dev = int(troco_total // 20)
gorjeta = troco_total % 20
print(notas_dev)
print("R$", format(gorjeta, ".2f"))
```

Note que quando o valor entregue pelo pai não é suficiente para a compra, João fica sem gorjeta. **Complete o programa abaixo** para que, nestes casos, João comente a situação e pergunte: “Pai, o dinheiro não foi suficiente... Posso ficar com R\$ 20,00?”. Seu programa deverá ler a resposta. Se o pai responder “SIM”, João irá ficar com uma das notas como gorjeta. Caso contrário, João continua sem nada, mas feliz por ter tentado. A saída deve continuar indicando, para todos os casos, o número de notas devolvidas para o pai e o valor da gorjeta de João. Não é necessário testar outras condições de erro ou fazer outras negociações com o pai.

```
notas_rec = int(input())
valor_compra = float(input())

if valor_compra > notas_rec * 20 :
    print("Pai, o dinheiro não foi suficiente... Posso ficar com R$ 20.00?")
    resp = input()
    if resp == "SIM":
        notas_dev = notas_rec - 1
        gorjeta = 20.00
    else:
        notas_dev = notas_rec
        gorjeta = 0.00
else:
    troco_total = notas_rec * 20 - valor_compra
    notas_dev = int(troco_total // 20)
    gorjeta = troco_total % 20

print(notas_dev)
print("R$", format(gorjeta, ".2f"))
```

4. (3.0 pontos) Considere um cenário de leilão semelhante ao proposto na tarefa de laboratório 06. A entrada é composta por um lance mínimo, por zero ou mais pares de linhas com um nome e um valor e pela palavra FIM, como no exemplo ao lado.

O leiloeiro não deverá ter paciência com clientes que ficam ofertando valores menores do que o lance mínimo. Ao ouvir um valor com estas características, deverá advertir o cliente pelo nome com a mensagem: '<nome_cliente>, por favor, emita apenas lances válidos!' Ao final, na ausência de lances válidos, o leiloeiro deverá dizer Item não foi vendido. Caso contrário, ele deverá anunciar o valor do maior lance com a mensagem: Vendido por <oferta_maxima>.

Implemente um **programa completo** que leia os dados e tenha o comportamento descrito acima sem utilizar listas. Utilize o verso desta folha, se necessário.

| |
|--------|
| 50.00 |
| Paulo |
| 40.00 |
| João |
| 120.00 |
| Maria |
| 150.00 |
| FIM |

```
lance_minimo = float(input())
oferta_maxima = 0
nome = input()
while nome != "FIM":
    lance = float(input())
    if lance < lance_minimo :
        print(nome + ", por favor, emita apenas lances válidos!")
    else:
        if lance > oferta_maxima :
            oferta_maxima = lance
        nome = input()
if oferta_maxima == 0 :
    print("Item não foi vendido.")
else:
    print("Vendido por", oferta_maxima)
```


MC102

Algoritmos e Programação de Computadores

Prova 1

Turmas A B C E F G H I

Primeiro Semestre de 2019

| Questão | Nota |
|---------|------|
| 1 | |
| 2 | |
| 3 | |
| 4 | |
| Total | |

Nome:

RA:

Importante: Esta avaliação é individual. Não é permitida a consulta a qualquer material impresso, nem o uso de celulares ou outros dispositivos eletrônicos de comunicação/computação. As respostas deverão ser escritas nos espaços demarcados, opcionalmente a lápis. Não se esqueça de indentar corretamente os códigos solicitados. **Boa prova!**

1. Após resolver as Atividades Conceituais na área MC102 no Moodle, Gabriela executou vários trechos de código para compreender melhor o funcionamento de alguns operadores e comandos. Em cada série, para cada trecho de código elaborado por ela, indique o que será escrito quando os programas forem executados. Caso o programa não escreva nada, indique “—”. Caso algum erro seja encontrado, indique o motivo e marque no código o ponto em que ele ocorre.

a) (1.0 ponto) **Tipos básicos**

```
# int ou float?
v = 2 + 5.0
print(v)
```

7.0

```
s1 = "X"
s2 = "Y"
print(s1 + s2)
```

XY

```
v = 5
v = "palavra"
print(v)
```

palavra

```
a = 5
b = "palavra"
print(c)
```

'c' não definido

```
q = 8 // 3
r = 8 % 3
print(q, r)
```

2 2

```
q = 13 // 5
r = 13 - q * 5
print("r =", r)
```

r = 3

```
a = 3 * 3 - 2
b = 4 - 7 / 2 + 2
print(a, b, a - b)
```

7 2.5 4.5

Dicas: // executa divisão inteira e % obtém o resto da divisão

```
# True ou
# False?
print(5 < 2)
```

False

```
a = 7
b = 4
print("a != b")
```

a != b

```
a = True
b = False
print(a, not b, a and b or b)
```

True True False

b) (1.0 ponto) **Comandos condicionais**

| | | | | |
|---|--|---|---|--|
| a = 11 b = 11 if a > b: print(a) if b >= a: print(b) if a == b: print(0) | a = 3 b = 3 if a > b: print(a) else: print(b) elif a == b: print(0) | a = 7 b = 7 if a > b: print(a) elif b > a: print(b) else: print(0) | a = 5 b = 3 if a > b: print(a) if b > a: print(b) else: print(0) | a = 10 b = 10 if a > b: print(a) else: print(b) if a == b: print(0) |
|---|--|---|---|--|

11
0

elif: erro de
sintaxe

0

5
0

10
0

c) (1.0 ponto) **Listas e comandos repetitivos**

```
lista = [0, 2, 7, 9, 12]
for i in range(len(lista)) :
    if lista[i] % 2 == 1 :
        lista[i] = lista[i] + 1
print(lista)
```

[0, 2, 8, 10, 12]

```
lista = [1, 3, 5, 7, 0]
i = 0
while lista[i] != 0 :
    lista[i] = lista[i] + 2
    i = i + 1
print(lista)
```

[3, 5, 7, 9, 0]

```
lista_a = [1, 3, 7, 4]
lista_b = [5, 2, 1, 6]
lista_c = []
for i in range(len(lista_a)) :
    if lista_a[i] < lista_b[i] :
        lista_c.append(lista_a[i])
    else:
        lista_c.append(lista_b[i])
print(lista_c)
```

[1, 2, 1, 4]

Dica: método `append()` adiciona um elemento ao final de uma lista

2. (1.5 ponto) Lembre-se da recomendação feita na tarefa de laboratório 03:

```
# NÃO!!!
if CP > 0.5 :
    cp_suficiente = True
else:
    cp_suficiente = False

# SIM!!!
cp_suficiente = CP > 0.5
```

Considerando que `a` e `b` são objetos que receberam algum valor inteiro, siga a recomendação e substitua os códigos abaixo pela atribuição de uma expressão à variável `r`. Você pode utilizar os operadores `not`, `and` e `or`.

```
if a > 10 :
    r = False
else:
    r = True
```

r = not a > 10

```
if a < 6 :
    r = True
elif b < 12 :
    r = True
else:
    r = False
```

r = a < 6 or b < 12

```
if a > 10 :
    if b > 25 :
        r = True
    else:
        r = False
else:
    r = False
```

r = a > 10 and b < 25 ou
a > 10 and b > 25

3. (2.5 pontos) João recebeu de seu pai algumas notas de R\$ 50,00 para fazer uma compra. Muito generoso, seu pai disse que ele receberia como gorjeta as notas e moedas do troco de valor inferior a R\$ 50,00. O programa abaixo lê o número de notas recebidas por João e o valor da compra. A saída indica o número de notas devolvidas para o pai e o valor da gorjeta.

```
notas_rec = int(input())
valor_compra = float(input())
troco_total = notas_rec * 50 - valor_compra
notas_dev = int(troco_total // 50)
gorjeta = troco_total % 50
print(notas_dev)
print("R$", format(gorjeta, ".2f"))
```

Note que quando o valor entregue pelo pai não é suficiente para a compra, João fica sem gorjeta. **Complete o programa abaixo** para que, nestes casos, João comente a situação e pergunte: “Pai, o dinheiro não foi suficiente... Posso ficar com R\$ 50,00?”. Seu programa deverá ler a resposta. Se o pai responder “SIM”, João irá ficar com uma das notas como gorjeta. Caso contrário, João continua sem nada, mas feliz por ter tentado. A saída deve continuar indicando, para todos os casos, o número de notas devolvidas para o pai e o valor da gorjeta de João. Não é necessário testar outras condições de erro ou fazer outras negociações com o pai.

```
notas_rec = int(input())
valor_compra = float(input())

if valor_compra > notas_rec * 50 :
    print("Pai, o dinheiro não foi suficiente... Posso ficar com R$ 50.00?")
    resp = input()
    if resp == "SIM":
        notas_dev = notas_rec - 1
        gorjeta = 50.00
    else:
        notas_dev = notas_rec
        gorjeta = 0.00
else:
    troco_total = notas_rec * 50 - valor_compra
    notas_dev = int(troco_total // 50)
    gorjeta = troco_total % 50

print(notas_dev)
print("R$", format(gorjeta, ".2f"))
```

4. (3.0 pontos) Considere um cenário de leilão semelhante ao proposto na tarefa de laboratório 06. A entrada é composta por um lance mínimo, por zero ou mais pares de linhas com um nome e um valor e pela palavra FIM, como no exemplo ao lado.

O leiloeiro não deverá ter paciência com clientes que ficam ofertando valores menores do que o lance mínimo. Ao ouvir um valor com estas características, deverá advertir o cliente pelo nome com a mensagem: '<nome_cliente>, por favor, emita apenas lances válidos!' Ao final, na ausência de lances válidos, o leiloeiro deverá dizer Item não foi vendido. Caso contrário, ele deverá anunciar o valor do maior lance com a mensagem: Vendido por <oferta_maxima>.

Implemente um **programa completo** que leia os dados e tenha o comportamento descrito acima sem utilizar listas. Utilize o verso desta folha, se necessário.

| |
|--------|
| 50.00 |
| Paulo |
| 40.00 |
| João |
| 120.00 |
| Maria |
| 150.00 |
| FIM |

```
lance_minimo = float(input())
oferta_maxima = 0
nome = input()
while nome != "FIM":
    lance = float(input())
    if lance < lance_minimo :
        print(nome + ", por favor, emita apenas lances válidos!")
    else:
        if lance > oferta_maxima :
            oferta_maxima = lance
        nome = input()
if oferta_maxima == 0 :
    print("Item não foi vendido.")
else:
    print("Vendido por", oferta_maxima)
```

MC102

Algoritmos e Programação de Computadores

Prova 1

Turmas A B C E F G H I

Primeiro Semestre de 2019

| Questão | Nota |
|---------|------|
| 1 | |
| 2 | |
| 3 | |
| 4 | |
| Total | |

Nome:

RA:

Importante: Esta avaliação é individual. Não é permitida a consulta a qualquer material impresso, nem o uso de celulares ou outros dispositivos eletrônicos de comunicação/computação. As respostas deverão ser escritas nos espaços demarcados, opcionalmente a lápis. Não se esqueça de indentar corretamente os códigos solicitados. **Boa prova!**

1. Após resolver as Atividades Conceituais na área MC102 no Moodle, Gabriela executou vários trechos de código para compreender melhor o funcionamento de alguns operadores e comandos. Em cada série, para cada trecho de código elaborado por ela, indique o que será escrito quando os programas forem executados. Caso o programa não escreva nada, indique “—”. Caso algum erro seja encontrado, indique o motivo e marque no código o ponto em que ele ocorre.

a) (1.0 ponto) **Tipos básicos**

```
# int ou float?
v = 2 + 13.0
print(v)
```

15.0

```
s1 = "X"
s2 = "Z"
print(s1 + s2)
```

XZ

```
a = 5
b = "palavra"
print(c)
```

'c' não definido

```
v = 5
v = "palavra"
print(v)
```

palavra

```
q = 11 // 5
r = 11 % 5
print(q, r)
```

2 1

```
q = 14 // 5
r = 14 - q * 5
print("r =", r)
```

r = 4

```
a = 5 * 2 - 2
b = 4 - 1 / 2 + 2
print(a, b, a - b)
```

8 5.5 2.5

Dicas: // executa divisão inteira e % obtém o resto da divisão

```
# True ou
# False?
print(5 > 2)
```

True

```
a = 4
b = 9
print("a == b")
```

a == b

```
a = True
b = False
print(not a, b, a and b or a)
```

False False True

b) (1.0 ponto) **Comandos condicionais**

| | | | | |
|---|---|---|---|---|
| <pre>a = 6 b = 6 if a > b: print(a) if b >= a: print(b) if a == b: print(0)</pre> | <pre>a = 7 b = 7 if a > b: print(a) elif b > a: print(b) else: print(0)</pre> | <pre>a = 10 b = 10 if a > b: print(a) else: print(b) if a == b: print(0)</pre> | <pre>a = 9 b = 3 if a > b: print(a) if b > a: print(b) else: print(0)</pre> | <pre>a = 5 b = 5 if a > b: print(a) else: print(b) elif a == b: print(0)</pre> |
| 6 0 | 0 | 10 0 | 9 0 | elif: erro de sintaxe |

c) (1.0 ponto) **Listas e comandos repetitivos**

```
lista = [0, 3, 4, 6, 11]
for i in range(len(lista)) :
    if lista[i] % 2 == 0 :
        lista[i] = lista[i] + 1
print(lista)
```

[1, 3, 5, 7, 11]

```
lista = [1, 3, 5, 7, 0]
i = 0
while lista[i] != 0 :
    lista[i] = -lista[i]
    i = i + 1
print(lista)
```

[-1, -3, -5, -7, 0]

```
lista_a = [1, 3, 7, 5]
lista_b = [5, 2, 1, 7]
lista_c = []
for i in range(len(lista_a)) :
    if lista_a[i] > lista_b[i] :
        lista_c.append(lista_a[i])
    else:
        lista_c.append(lista_b[i])
print(lista_c)
```

[5, 3, 7, 7]

Dica: método `append()` adiciona um elemento ao final de uma lista

2. (1.5 ponto) Lembre-se da recomendação feita na tarefa de laboratório 03:

```
# NÃO!!!
if CP > 0.5 :
    cp_suficiente = True
else:
    cp_suficiente = False

# SIM!!!
cp_suficiente = CP > 0.5
```

Considerando que `a` e `b` são objetos que receberam algum valor inteiro, siga a recomendação e substitua os códigos abaixo pela atribuição de uma expressão à variável `r`. Você pode utilizar os operadores `not`, `and` e `or`.

```
if a < 10 :
    r = False
else:
    r = True
```

r = not a < 10

```
if a > 15 :
    r = True
elif b < 30 :
    r = True
else:
    r = False
```

r = a > 15 or b < 30

```
if a > 5 :
    if b < 30 :
        r = True
    else:
        r = False
else:
    r = False
```

r = a > 5 and b < 30

3. (2.5 pontos) João recebeu de seu pai algumas notas de R\$ 20,00 para fazer uma compra. Muito generoso, seu pai disse que ele receberia como gorjeta as notas e moedas do troco de valor inferior a R\$ 20,00. O programa abaixo lê o número de notas recebidas por João e o valor da compra. A saída indica o número de notas devolvidas para o pai e o valor da gorjeta.

```
notas_rec = int(input())
valor_compra = float(input())
troco_total = notas_rec * 20 - valor_compra
notas_dev = int(troco_total // 20)
gorjeta = troco_total % 20
print(notas_dev)
print("R$", format(gorjeta, ".2f"))
```

Note que quando o valor entregue pelo pai não é suficiente para a compra, João fica sem gorjeta. **Complete o programa abaixo** para que, nestes casos, João comente a situação e pergunte: “Pai, o dinheiro não foi suficiente... Posso ficar com R\$ 20,00?”. Seu programa deverá ler a resposta. Se o pai responder “SIM”, João irá ficar com uma das notas como gorjeta. Caso contrário, João continua sem nada, mas feliz por ter tentado. A saída deve continuar indicando, para todos os casos, o número de notas devolvidas para o pai e o valor da gorjeta de João. Não é necessário testar outras condições de erro ou fazer outras negociações com o pai.

```
notas_rec = int(input())
valor_compra = float(input())

if valor_compra > notas_rec * 20 :
    print("Pai, o dinheiro não foi suficiente... Posso ficar com R$ 20.00?")
    resp = input()
    if resp == "SIM":
        notas_dev = notas_rec - 1
        gorjeta = 20.00
    else:
        notas_dev = notas_rec
        gorjeta = 0.00
else:
    troco_total = notas_rec * 20 - valor_compra
    notas_dev = int(troco_total // 20)
    gorjeta = troco_total % 20

print(notas_dev)
print("R$", format(gorjeta, ".2f"))
```

4. (3.0 pontos) Considere um cenário de leilão semelhante ao proposto na tarefa de laboratório 06. A entrada é composta por um lance mínimo, por zero ou mais pares de linhas com um nome e um valor e pela palavra FIM, como no exemplo ao lado.

O leiloeiro não deverá ter paciência com clientes que ficam ofertando valores menores do que o lance mínimo. Ao ouvir um valor com estas características, deverá advertir o cliente pelo nome com a mensagem: '<nome_cliente>, por favor, emita apenas lances válidos!' Ao final, na ausência de lances válidos, o leiloeiro deverá dizer Item não foi vendido. Caso contrário, ele deverá anunciar o valor do maior lance com a mensagem: Vendido por <oferta_maxima>.

Implemente um **programa completo** que leia os dados e tenha o comportamento descrito acima sem utilizar listas. Utilize o verso desta folha, se necessário.

| |
|--------|
| 50.00 |
| Paulo |
| 40.00 |
| João |
| 120.00 |
| Maria |
| 150.00 |
| FIM |

```
lance_minimo = float(input())
oferta_maxima = 0
nome = input()
while nome != "FIM":
    lance = float(input())
    if lance < lance_minimo :
        print(nome + ", por favor, emita apenas lances válidos!")
    else:
        if lance > oferta_maxima :
            oferta_maxima = lance
        nome = input()
if oferta_maxima == 0 :
    print("Item não foi vendido.")
else:
    print("Vendido por", oferta_maxima)
```