

Repetição em Python

Como qualquer linguagem estruturada moderna, Python tem as ferramentas básicas para interar. São elas: while, for, bem como as saídas break e continue. A mais amigável é while. Já a a mais potente é for, pois funciona como um interador, que é uma extensão e tanto ao conceito de indexador. Já os comandos break permitem o abandono de um ciclo, enquanto continue serve para abandonar a interação atual. Veja alguns exemplos:

```
i=3
while i<10:
    print(i,' ')
    if i%7==0:
        break
    i=i+1
print('acabou')
```

```
i=3
while i<10:
    print(i,' ')
    if i%7==0:
        continue
    i=i+1
print('acabou')
```

```
s="Curitiba - Paraná - Brasil"
qtd=0
for c in s:
    if c=="a":
        qtd=qtd+1
print("achei ",qtd," a")
```

Exercícios Nos exercícios a seguir, você deve simular o interpretador Python e descobrir qual valor vai ser impresso ao final. Todos os códigos estão sintaticamente corretos. A sugestão é que depois de ter calculado o valor na ponta do lápis, você confira o resultado com um computador real.

Exemplos

Seja o programa

```
def AA():
    A=5;B=5;C=3
    if C>=18:
        while B<=18:
            C=C-3
            B=B+4
        else:
            C=C+4
            B=B+2
        print(C+C+B)
AA()
```

Cujo resultado é 21 e seja o seguinte código

```
def AA():
    A=4;B=3;C=6
    D=1
    C=C+6+3
    while C>6:
        B=D+7
        while A<=14:
            D=B+4
            A=A+2
            D=B+19
            C=C-1
        print(C+B+B)
AA()
```

cujo resultado é 438.

🔗 Para você fazer

Exercício 1

```
def AA():
    A=6;B=8;C=3
    while C<14:
        A=A+1
        A=A+6
        C=C+1
        print(A+A+C)
AA()
```

Informe no quadro próprio o valor que foi impresso.

Exercício 2

```
def AA():
    A=2;B=1;C=1
    if C>13:
        while B<13:
            C=C-6
            B=B+4
        else:
            C=C+4
            B=B+1
        print(C+C+B)
AA()
```

Exercício 3

```
def AA():
    A=1;B=6;C=4
    if B==20:
        while C<20:
            B=A+2
            C=C+3
        else:
            while C<12:
                A=B-5
                C=C+2
            print(B+A+C)
AA()
```

Exercício 4

```
def AA():
    A=2;B=4;C=7
    D=3
    D=D+7+4
    while D>=5:
        A=D+6
        while C<16:
            A=B+4
            C=C+1
            B=B+13
            D=D-4
        print(D+B+A)
AA()
```

Exercício 5

```
def AA():
    A=3;B=8;C=5
    D=7
    C=C+5+8
    while C>5:
        B=C+6
        if A<=6:
            while A<17:
                D=B-1
                A=A+2
                D=B+15
            else:
                D=B+6
                C=C-4
        print(D+D)
AA()
```

Exercício 6

```
def AA():
    A=1;B=8;C=6
    D=5
    C=C+6+8
    while C>7:
        A=D+4
        if B>=10:
            while B<17:
                B=B+3
                A=D+6
            else:
                A=A-18
        if D>7:
            D=D+4
        else:
            A=A-6
            C=C-2
        print(C+B)
AA()
```

Exercício 7

```
def AA():
    A=1;B=6;C=5
    D=3
    A=A+5+6
    while A>=1:
        B=A+3
        if A>=10:
```

```
        while D<18:
            D=D+5
            B=B-1
        else:
            C=B-16
        if D<7:
            C=C+9
        else:
            B=C-9
            A=A-4
        print(B+A)
AA()
```

Exercício 8

```
def AA():
    A=5;B=8;C=2
    D=4
    C=C+2+8
    while C>=4:
        B=D+7
        if D>8:
            while A<15:
                B=B+2
                A=A+6
                B=B+14
            else:
                D=D+5
                C=C-2
        print(A+B)
AA()
```

Exercício 9

```
def AA():
    A=8;B=4;C=2
    D=3
    B=B+2+4
    while B>=7:
        A=D+3
        while D<16:
            C=A-7
            D=D+5
            A=A+15
            B=B-4
        print(C+C+B)
AA()
```

Exercício 10

```
def AA():
    A=4;B=5;C=4
    if B>=16:
        while C<16:
            B=B-7
            C=C+3
        else:
            while C<12:
                B=A+7
                C=C+1
            print(B+B+C)
AA()
```

Responda aqui:

1	2
3	4
5	6
7	8
9	10



Repetição em Python

Como qualquer linguagem estruturada moderna, Python tem as ferramentas básicas para interar. São elas: while, for, bem como as saídas break e continue. A mais amigável é while. Já a a mais potente é for, pois funciona como um interador, que é uma extensão e tanto ao conceito de indexador. Já os comandos break permitem o abandono de um ciclo, enquanto continue serve para abandonar a interação atual. Veja alguns exemplos:

```
i=3
while i<10:
    print(i,' ')
    if i%7==0:
        break
    i=i+1
print('acabou')
```

```
i=3
while i<10:
    print(i,' ')
    if i%7==0:
        continue
    i=i+1
print('acabou')
```

```
s="Curitiba - Paraná - Brasil"
qtd=0
for c in s:
    if c=="a":
        qtd=qtd+1
print("achei ",qtd," a")
```

Exercícios Nos exercícios a seguir, você deve simular o interpretador Python e descobrir qual valor vai ser impresso ao final. Todos os códigos estão sintaticamente corretos. A sugestão é que depois de ter calculado o valor na ponta do lápis, você confira o resultado com um computador real.

Exemplos

Seja o programa

```
def AA():
    A=5;B=5;C=3
    if C>=18:
        while B<=18:
            C=C-3
            B=B+4
        else:
            C=C+4
            B=B+2
    print(C+C+B)
AA()
```

Cujo resultado é 21 e seja o seguinte código

```
def AA():
    A=4;B=3;C=6
    D=1
    C=C+6+3
    while C>6:
        B=D+7
        while A<=14:
            D=B+4
            A=A+2
            D=B+19
            C=C-1
        print(C+B+B)
AA()
```

cujo resultado é 438.

🔗 Para você fazer

Exercício 1

```
def AA():
    A=6;B=8;C=4
    while B<14:
        C=C+4
        C=A-1
        B=B+1
    print(C+C+B)
AA()
```

Informe no quadro próprio o valor que foi impresso.

Exercício 2

```
def AA():
    A=4;B=1;C=2
    if A<12:
        while C<=12:
            A=B+5
            C=C+4
        else:
            B=B+8
            C=C+2
    print(A+B+C)
AA()
```

Exercício 3

```
def AA():
    A=6;B=4;C=8
    if B<=17:
        while C<=17:
            B=B+4
            C=C+3
        else:
            while C<=12:
                A=B-4
                C=C+2
            print(B+B+C)
AA()
```

Exercício 4

```
def AA():
    A=5;B=8;C=6
    D=3
    A=A+6+8
    while A>6:
        C=D+5
        while B<=16:
            C=C-7
            B=B+3
            C=D-19
            A=A-4
        print(B+A+D)
AA()
```

Exercício 5

```
def AA():
    A=5;B=3;C=4
    D=8
    A=A+4+3
    while A>4:
        B=C-5
        if B>=7:
            while D<13:
                C=B-8
                D=D+1
            C=B-19
        else:
            C=B-5
            A=A-4
    print(A+A)
AA()
```

Exercício 6

```
def AA():
    A=5;B=3;C=2
    D=4
    C=C+2+3
    while C>=4:
        A=B+8
        if C<=9:
            while B<=13:
                B=B+6
            A=A-5
        else:
            D=D+14
        if B>4:
            D=D+6
        else:
            A=D+10
            C=C-4
    print(A+C)
AA()
```

Exercício 7

```
def AA():
    A=7;B=2;C=8
    D=1
    C=C+8+2
    while C>=1:
        A=A-8
        if A<=10:
```

```
        while D<19:
            D=D+3
            A=A+6
        else:
            A=B+14
        if C<=8:
            B=B+5
        else:
            A=B+7
            C=C-2
    print(B+B)
AA()
```

Exercício 8

```
def AA():
    A=3;B=8;C=7
    D=1
    D=D+7+8
    while D>=3:
        A=B-1
        if D<5:
            while C<16:
                B=B+3
                C=C+4
            A=B+13
        else:
            A=B+8
            D=D-3
    print(A+B)
AA()
```

Exercício 9

```
def AA():
    A=4;B=7;C=6
    D=5
    B=B+6+7
    while B>1:
        A=D+1
        while C<19:
            D=A-5
            C=C+1
        A=D-17
        B=B-4
    print(B+B+D)
AA()
```

Exercício 10

```
def AA():
    A=6;B=7;C=3
    if A>15:
        while C<=15:
            A=B+6
            C=C+3
        else:
            while C<=18:
                B=A+1
                C=C+1
            print(A+B+C)
AA()
```

Responda aqui:

1	2
3	4
5	6
7	8
9	10



Repetição em Python

Como qualquer linguagem estruturada moderna, Python tem as ferramentas básicas para interar. São elas: while, for, bem como as saídas break e continue. A mais amigável é while. Já a a mais potente é for, pois funciona como um interador, que é uma extensão e tanto ao conceito de indexador. Já os comandos break permitem o abandono de um ciclo, enquanto continue serve para abandonar a interação atual. Veja alguns exemplos:

```
i=3
while i<10:
    print(i,' ')
    if i%7==0:
        break
    i=i+1
print('acabou')
```

```
i=3
while i<10:
    print(i,' ')
    if i%7==0:
        continue
    i=i+1
print('acabou')
```

```
s="Curitiba - Paraná - Brasil"
qtd=0
for c in s:
    if c=="a":
        qtd=qtd+1
print("achei ",qtd," a")
```

Exercícios Nos exercícios a seguir, você deve simular o interpretador Python e descobrir qual valor vai ser impresso ao final. Todos os códigos estão sintaticamente corretos. A sugestão é que depois de ter calculado o valor na ponta do lápis, você confira o resultado com um computador real.

Exemplos

Seja o programa

```
def AA():
    A=5;B=5;C=3
    if C>=18:
        while B<=18:
            C=C-3
            B=B+4
        else:
            C=C+4
            B=B+2
            print(C+C+B)
AA()
```

Cujo resultado é 21 e seja o seguinte código

```
def AA():
    A=4;B=3;C=6
    D=1
    C=C+6+3
    while C>6:
        B=D+7
        while A<=14:
            D=B+4
            A=A+2
            D=B+19
            C=C-1
            print(C+B+B)
AA()
```

cujo resultado é 438.

🔗 Para você fazer

Exercício 1

```
def AA():
    A=5;B=8;C=2
    while B<15:
        C=A+6
        A=C-7
        B=B+4
        print(C+A+B)
AA()
```

Informe no quadro próprio o valor que foi impresso.

Exercício 2

```
def AA():
    A=2;B=6;C=1
    if A!=16:
        while B<16:
            A=A-4
            B=B+4
        else:
            A=A+1
            B=B+2
            print(A+A+B)
AA()
```

Exercício 3

```
def AA():
    A=8;B=7;C=5
    if B<15:
        while C<15:
            B=B+7
            C=C+1
        else:
            while C<14:
                B=A-5
                C=C+1
                print(B+B+C)
AA()
```

Exercício 4

```
def AA():
    A=5;B=3;C=1
    D=6
    D=D+1+3
    while D>=6:
        A=D-4
        while C<15:
            B=A+4
            C=C+2
            B=B+15
            D=D-3
            print(C+D+C)
AA()
```

Exercício 5

```
def AA():
    A=4;B=2;C=1
    D=3
    D=D+1+2
    while D>=3:
        B=D+2
        if A<=9:
            while A<=18:
                B=C-4
                A=A+5
                B=C+15
            else:
                B=B+8
                D=D-1
                print(D+C)
AA()
```

Exercício 6

```
def AA():
    A=4;B=5;C=8
    D=1
    D=D+8+5
    while D>=7:
        A=D+2
        if C<10:
            while B<=13:
                B=B+6
                A=C+1
            else:
                C=A+15
        if A<6:
            C=C+10
        else:
            A=C+4
            D=D-1
            print(D+B)
AA()
```

Exercício 7

```
def AA():
    A=1;B=6;C=5
    D=2
    C=C+5+6
    while C>=3:
        A=B+4
        if C>10:
```

```
        while B<12:
            B=B+2
            D=D+4
        else:
            A=A-19
        if C<=8:
            D=D+4
        else:
            A=D-4
            C=C-1
        print(B+A)
AA()
```

Exercício 8

```
def AA():
    A=3;B=8;C=6
    D=2
    A=A+6+8
    while A>=4:
        B=D-4
        if C<5:
            while C<17:
                D=B-3
                C=C+2
                B=B+14
            else:
                B=D+5
                A=A-4
                print(B+C)
AA()
```

Exercício 9

```
def AA():
    A=2;B=1;C=3
    D=8
    C=C+3+1
    while C>=4:
        A=A+6
        while B<19:
            A=A-4
            B=B+6
            D=A-13
            C=C-1
            print(A+B+D)
AA()
```

Exercício 10

```
def AA():
    A=4;B=2;C=5
    if A!=13:
        while B<13:
            A=C-6
            B=B+4
        else:
            while B<14:
                A=A+1
                B=B+3
                print(A+C+B)
AA()
```

Responda aqui:

1	2
3	4
5	6
7	8
9	10



Repetição em Python

Como qualquer linguagem estruturada moderna, Python tem as ferramentas básicas para interar. São elas: while, for, bem como as saídas break e continue. A mais amigável é while. Já a a mais potente é for, pois funciona como um interador, que é uma extensão e tanto ao conceito de indexador. Já os comandos break permitem o abandono de um ciclo, enquanto continue serve para abandonar a interação atual. Veja alguns exemplos:

```
i=3
while i<10:
    print(i,' ')
    if i%7==0:
        break
    i=i+1
print('acabou')
```

```
i=3
while i<10:
    print(i,' ')
    if i%7==0:
        continue
    i=i+1
print('acabou')
```

```
s="Curitiba - Paraná - Brasil"
qtd=0
for c in s:
    if c=="a":
        qtd=qtd+1
print("achei ",qtd," a")
```

Exercícios Nos exercícios a seguir, você deve simular o interpretador Python e descobrir qual valor vai ser impresso ao final. Todos os códigos estão sintaticamente corretos. A sugestão é que depois de ter calculado o valor na ponta do lápis, você confira o resultado com um computador real.

Exemplos

Seja o programa

```
def AA():
    A=5;B=5;C=3
    if C>=18:
        while B<=18:
            C=C-3
            B=B+4
        else:
            C=C+4
            B=B+2
        print(C+C+B)
AA()
```

Cujo resultado é 21 e seja o seguinte código

```
def AA():
    A=4;B=3;C=6
    D=1
    C=C+6+3
    while C>6:
        B=D+7
        while A<=14:
            D=B+4
            A=A+2
            D=B+19
            C=C-1
        print(C+B+B)
AA()
```

cujo resultado é 438.

🔗 Para você fazer

Exercício 1

```
def AA():
    A=1;B=2;C=5
    while B<17:
        A=A+3
        A=C-2
        B=B+3
        print(A+A+B)
AA()
```

Informe no quadro próprio o valor que foi impresso.

Exercício 2

```
def AA():
    A=7;B=5;C=5
    if B<19:
        while C<19:
            B=A+6
            C=C+4
        else:
            A=A-7
            C=C+1
        print(B+A+C)
AA()
```

Exercício 3

```
def AA():
    A=2;B=7;C=3
    if A<20:
        while B<20:
            A=A+1
            B=B+2
        else:
            while B<12:
                A=C+2
                B=B+2
            print(A+A+B)
AA()
```

Exercício 4

```
def AA():
    A=7;B=8;C=3
    D=4
    B=B+3+8
    while B>=5:
        A=C+1
        while D<=12:
            C=A+3
            D=D+2
            A=A+19
            B=B-3
        print(B+D+B)
AA()
```

Exercício 5

```
def AA():
    A=4;B=3;C=6
    D=2
    B=B+6+3
    while B>6:
        A=C+5
        if B>5:
            while C<=12:
                D=A+6
                C=C+1
                A=A+12
            else:
                D=A+5
                B=B-1
        print(C+D)
AA()
```

Exercício 6

```
def AA():
    A=1;B=7;C=2
    D=3
    D=D+2+7
    while D>3:
        A=B+8
        if C<7:
            while C<17:
                C=C+2
                A=A+8
            else:
                A=B-19
        if D>9:
            A=A+8
        else:
            B=A-7
            D=D-2
        print(D+C)
AA()
```

Exercício 7

```
def AA():
    A=7;B=6;C=4
    D=5
    B=B+4+6
    while B>=5:
        A=B+7
        if B>=9:
```

```
        while D<13:
            D=D+5
            C=A-5
        else:
            C=A-16
        if C<=6:
            C=C+10
        else:
            A=C-5
            B=B-1
        print(B+D)
AA()
```

Exercício 8

```
def AA():
    A=8;B=2;C=3
    D=1
    B=B+3+2
    while B>=5:
        A=D-8
        if A>7:
            while D<12:
                C=A-7
                D=D+3
                C=A+15
            else:
                A=C+9
                B=B-1
        print(B+A)
AA()
```

Exercício 9

```
def AA():
    A=7;B=3;C=6
    D=5
    C=C+6+3
    while C>=6:
        A=D+1
        while B<=13:
            A=A+6
            B=B+4
            D=D+19
            C=C-1
        print(D+B+B)
AA()
```

Exercício 10

```
def AA():
    A=7;B=4;C=1
    if B>19:
        while C<=19:
            B=A+8
            C=C+4
        else:
            while C<=13:
                A=A+5
                C=C+3
            print(B+A+C)
AA()
```

Responda aqui:

1	2
3	4
5	6
7	8
9	10



Repetição em Python

Como qualquer linguagem estruturada moderna, Python tem as ferramentas básicas para interar. São elas: while, for, bem como as saídas break e continue. A mais amigável é while. Já a a mais potente é for, pois funciona como um interador, que é uma extensão e tanto ao conceito de indexador. Já os comandos break permitem o abandono de um ciclo, enquanto continue serve para abandonar a interação atual. Veja alguns exemplos:

```
i=3
while i<10:
    print(i,' ')
    if i%7==0:
        break
    i=i+1
print('acabou')
```

```
i=3
while i<10:
    print(i,' ')
    if i%7==0:
        continue
    i=i+1
print('acabou')
```

```
s="Curitiba - Paraná - Brasil"
qtd=0
for c in s:
    if c=="a":
        qtd=qtd+1
print("achei ",qtd," a")
```

Exercícios Nos exercícios a seguir, você deve simular o interpretador Python e descobrir qual valor vai ser impresso ao final. Todos os códigos estão sintaticamente corretos. A sugestão é que depois de ter calculado o valor na ponta do lápis, você confira o resultado com um computador real.

Exemplos

Seja o programa

```
def AA():
    A=5;B=5;C=3
    if C>=18:
        while B<=18:
            C=C-3
            B=B+4
        else:
            C=C+4
            B=B+2
    print(C+C+B)
AA()
```

Cujo resultado é 21 e seja o seguinte código

```
def AA():
    A=4;B=3;C=6
    D=1
    C=C+6+3
    while C>6:
        B=D+7
        while A<=14:
            D=B+4
            A=A+2
            D=B+19
            C=C-1
        print(C+B+B)
AA()
```

cujo resultado é 438.

🔗 Para você fazer

Exercício 1

```
def AA():
    A=5;B=7;C=8
    while A<=19:
        C=C+8
        C=B-3
        A=A+4
    print(C+C+A)
AA()
```

Informe no quadro próprio o valor que foi impresso.

Exercício 2

```
def AA():
    A=2;B=5;C=7
    if B>18:
        while C<18:
            B=A+7
            C=C+4
        else:
            B=A+2
            C=C+4
    print(B+A+C)
AA()
```

Exercício 3

```
def AA():
    A=8;B=3;C=5
    if B>=12:
        while A<=12:
            B=C+1
            A=A+2
        else:
            while A<=17:
                B=B-4
                A=A+3
            print(B+C+A)
AA()
```

Exercício 4

```
def AA():
    A=3;B=2;C=8
    D=7
    D=D+8+2
    while D>7:
        A=C+2
        while C<=14:
            B=B+5
            C=C+6
        A=B+15
        D=D-2
    print(D+A+C)
AA()
```

Exercício 5

```
def AA():
    A=5;B=3;C=2
    D=8
    D=D+2+3
    while D>=4:
        A=C+6
        if C>=9:
            while C<=17:
                A=B+8
                C=C+5
            B=A+16
        else:
            B=A+10
            D=D-3
    print(B+B)
AA()
```

Exercício 6

```
def AA():
    A=7;B=3;C=1
    D=2
    C=C+1+3
    while C>4:
        A=C-6
        if C<=7:
            while B<14:
                B=B+6
            A=D+8
        else:
            D=A-17
    if D>4:
        A=A+9
    else:
        A=A-6
        C=C-4
    print(D+A)
AA()
```

Exercício 7

```
def AA():
    A=6;B=2;C=4
    D=1
    B=B+4+2
    while B>3:
        A=B-8
        if C<=6:
```

```
        while C<14:
            C=C+5
            A=A+8
        else:
            A=D-13
    if B<=10:
        A=A+6
    else:
        D=A-6
        B=B-3
    print(D+A)
AA()
```

Exercício 8

```
def AA():
    A=1;B=4;C=8
    D=3
    A=A+8+4
    while A>2:
        B=D+1
        if C<4:
            while D<=17:
                B=C-3
                D=D+5
            B=C-13
        else:
            B=B-4
            A=A-3
    print(D+A)
AA()
```

Exercício 9

```
def AA():
    A=3;B=1;C=2
    D=5
    C=C+2+1
    while C>=1:
        B=B+8
        while A<18:
            B=D-2
            A=A+1
        D=B+17
        C=C-2
    print(B+B+B)
AA()
```

Exercício 10

```
def AA():
    A=4;B=3;C=3
    if A<=13:
        while C<13:
            A=B+5
            C=C+1
        else:
            while C<15:
                A=A-7
                C=C+1
            print(A+B+C)
AA()
```

Responda aqui:

1	2
3	4
5	6
7	8
9	10



Repetição em Python

Como qualquer linguagem estruturada moderna, Python tem as ferramentas básicas para interar. São elas: while, for, bem como as saídas break e continue. A mais amigável é while. Já a a mais potente é for, pois funciona como um interador, que é uma extensão e tanto ao conceito de indexador. Já os comandos break permitem o abandono de um ciclo, enquanto continue serve para abandonar a interação atual. Veja alguns exemplos:

```
i=3
while i<10:
    print(i,' ')
    if i%7==0:
        break
    i=i+1
print('acabou')
```

```
i=3
while i<10:
    print(i,' ')
    if i%7==0:
        continue
    i=i+1
print('acabou')
```

```
s="Curitiba - Paraná - Brasil"
qtd=0
for c in s:
    if c=="a":
        qtd=qtd+1
print("achei ",qtd," a")
```

Exercícios Nos exercícios a seguir, você deve simular o interpretador Python e descobrir qual valor vai ser impresso ao final. Todos os códigos estão sintaticamente corretos. A sugestão é que depois de ter calculado o valor na ponta do lápis, você confira o resultado com um computador real.

Exemplos

Seja o programa

```
def AA():
    A=5;B=5;C=3
    if C>=18:
        while B<=18:
            C=C-3
            B=B+4
        else:
            C=C+4
            B=B+2
    print(C+C+B)
AA()
```

Cujo resultado é 21 e seja o seguinte código

```
def AA():
    A=4;B=3;C=6
    D=1
    C=C+6+3
    while C>6:
        B=D+7
        while A<=14:
            D=B+4
            A=A+2
            D=B+19
            C=C-1
        print(C+B+B)
AA()
```

cujo resultado é 438.

🔗 Para você fazer

Exercício 1

```
def AA():
    A=5;B=2;C=8
    while A<=14:
        B=B+6
        B=C+2
        A=A+2
    print(B+B+A)
AA()
```

Informe no quadro próprio o valor que foi impresso.

Exercício 2

```
def AA():
    A=5;B=6;C=2
    if C<=15:
        while B<=15:
            C=C+5
            B=B+4
        else:
            A=C-6
            B=B+2
    print(C+C+B)
AA()
```

Exercício 3

```
def AA():
    A=8;B=4;C=1
    if B<=18:
        while C<=18:
            B=B+1
            C=C+1
        else:
            while C<=17:
                A=B+8
                C=C+1
            print(B+B+C)
AA()
```

Exercício 4

```
def AA():
    A=3;B=8;C=2
    D=1
    A=A+2+8
    while A>=4:
        B=A+6
        while C<17:
            B=D+3
            C=C+3
            D=B+13
            A=A-2
        print(C+D+A)
AA()
```

Exercício 5

```
def AA():
    A=5;B=1;C=2
    D=7
    C=C+2+1
    while C>2:
        B=B+1
        if A>8:
            while A<12:
                D=B+6
                A=A+3
                D=D-14
            else:
                D=B-10
                C=C-1
        print(C+C)
AA()
```

Exercício 6

```
def AA():
    A=3;B=5;C=1
    D=2
    B=B+1+5
    while B>1:
        A=D+4
        if B<7:
            while C<=17:
                C=C+2
                A=D-4
            else:
                D=D+17
        if C<=8:
            A=A+7
        else:
            D=A+4
            B=B-3
        print(D+C)
AA()
```

Exercício 7

```
def AA():
    A=4;B=6;C=7
    D=5
    D=D+7+6
    while D>1:
        A=D-2
        if C<4:
```

```
        while B<16:
            B=B+2
            A=C-7
        else:
            A=A-13
        if D>=6:
            A=A+4
        else:
            C=A-9
            D=D-3
        print(D+D)
AA()
```

Exercício 8

```
def AA():
    A=8;B=7;C=4
    D=5
    D=D+4+7
    while D>=5:
        A=D+7
        if D>=7:
            while B<=13:
                A=C+6
                B=B+3
                C=C+13
            else:
                A=A+7
                D=D-2
        print(B+D)
AA()
```

Exercício 9

```
def AA():
    A=5;B=3;C=7
    D=2
    D=D+7+3
    while D>6:
        A=A+2
        while C<15:
            B=A-8
            C=C+1
            B=B+19
            D=D-1
        print(B+A+B)
AA()
```

Exercício 10

```
def AA():
    A=7;B=2;C=4
    if A>13:
        while B<13:
            A=C+3
            B=B+3
        else:
            while B<13:
                A=C-3
                B=B+3
            print(A+C+B)
AA()
```

Responda aqui:

1	2
3	4
5	6
7	8
9	10



Repetição em Python

Como qualquer linguagem estruturada moderna, Python tem as ferramentas básicas para interar. São elas: while, for, bem como as saídas break e continue. A mais amigável é while. Já a a mais potente é for, pois funciona como um interador, que é uma extensão e tanto ao conceito de indexador. Já os comandos break permitem o abandono de um ciclo, enquanto continue serve para abandonar a interação atual. Veja alguns exemplos:

```
i=3
while i<10:
    print(i,' ')
    if i%7==0:
        break
    i=i+1
print('acabou')
```

```
i=3
while i<10:
    print(i,' ')
    if i%7==0:
        continue
    i=i+1
print('acabou')
```

```
s="Curitiba - Paraná - Brasil"
qtd=0
for c in s:
    if c=="a":
        qtd=qtd+1
print("achei ",qtd," a")
```

Exercícios Nos exercícios a seguir, você deve simular o interpretador Python e descobrir qual valor vai ser impresso ao final. Todos os códigos estão sintaticamente corretos. A sugestão é que depois de ter calculado o valor na ponta do lápis, você confira o resultado com um computador real.

Exemplos

Seja o programa

```
def AA():
    A=5;B=5;C=3
    if C>=18:
        while B<=18:
            C=C-3
            B=B+4
        else:
            C=C+4
            B=B+2
    print(C+C+B)
AA()
```

Cujo resultado é 21 e seja o seguinte código

```
def AA():
    A=4;B=3;C=6
    D=1
    C=C+6+3
    while C>6:
        B=D+7
        while A<=14:
            D=B+4
            A=A+2
            D=B+19
            C=C-1
        print(C+B+B)
AA()
```

cujo resultado é 438.

🔗 Para você fazer

Exercício 1

```
def AA():
    A=6;B=3;C=1
    while A<15:
        B=B+4
        B=B-3
        A=A+2
    print(B+B+A)
AA()
```

Informe no quadro próprio o valor que foi impresso.

Exercício 2

```
def AA():
    A=2;B=2;C=5
    if C==17:
        while B<17:
            C=C+2
            B=B+4
        else:
            C=A-5
            B=B+1
    print(C+C+B)
AA()
```

Exercício 3

```
def AA():
    A=7;B=2;C=6
    if A==14:
        while C<14:
            A=B+1
            C=C+1
        else:
            while C<14:
                B=B-1
                C=C+3
            print(A+B+C)
AA()
```

Exercício 4

```
def AA():
    A=5;B=3;C=2
    D=6
    B=B+2+3
    while B>1:
        A=D-7
        while D<=12:
            A=A+2
            D=D+5
            C=C-15
            B=B-3
        print(D+B+D)
AA()
```

Exercício 5

```
def AA():
    A=4;B=5;C=7
    D=3
    A=A+7+5
    while A>2:
        B=D+1
        if D<=8:
            while C<13:
                D=D+3
                C=C+6
            B=D+14
        else:
            B=B+10
            A=A-1
    print(A+B)
AA()
```

Exercício 6

```
def AA():
    A=7;B=3;C=2
    D=4
    D=D+2+3
    while D>2:
        A=A-7
        if C>4:
            while C<16:
                C=C+3
            B=B+6
        else:
            A=B+13
        if C>8:
            A=A+7
        else:
            B=A+6
            D=D-3
    print(B+D)
AA()
```

Exercício 7

```
def AA():
    A=3;B=8;C=6
    D=1
    A=A+6+8
    while A>=3:
        B=B+3
        if A<=6:
```

```
        while D<=13:
            D=D+4
            B=B-1
        else:
            B=C+15
        if C>=6:
            C=C+4
        else:
            C=C+7
            A=A-4
    print(D+C)
AA()
```

Exercício 8

```
def AA():
    A=4;B=7;C=5
    D=8
    A=A+5+7
    while A>=2:
        B=D-2
        if A>8:
            while C<14:
                D=D-7
                C=C+3
            B=B+15
        else:
            D=D+5
            A=A-2
    print(A+C)
AA()
```

Exercício 9

```
def AA():
    A=1;B=4;C=7
    D=6
    D=D+7+4
    while D>1:
        A=C+2
        while B<=15:
            C=C+1
            B=B+3
            C=A+13
            D=D-1
        print(C+C+C)
AA()
```

Exercício 10

```
def AA():
    A=5;B=4;C=3
    if A<15:
        while C<=15:
            A=B-8
            C=C+2
        else:
            while C<=14:
                B=B-3
                C=C+2
            print(A+B+C)
AA()
```

Responda aqui:

1	2
3	4
5	6
7	8
9	10



Repetição em Python

Como qualquer linguagem estruturada moderna, Python tem as ferramentas básicas para interar. São elas: while, for, bem como as saídas break e continue. A mais amigável é while. Já a a mais potente é for, pois funciona como um interador, que é uma extensão e tanto ao conceito de indexador. Já os comandos break permitem o abandono de um ciclo, enquanto continue serve para abandonar a interação atual. Veja alguns exemplos:

```
i=3
while i<10:
    print(i,' ')
    if i%7==0:
        break
    i=i+1
print('acabou')
```

```
i=3
while i<10:
    print(i,' ')
    if i%7==0:
        continue
    i=i+1
print('acabou')
```

```
s="Curitiba - Paraná - Brasil"
qtd=0
for c in s:
    if c=="a":
        qtd=qtd+1
print("achei ",qtd," a")
```

Exercícios Nos exercícios a seguir, você deve simular o interpretador Python e descobrir qual valor vai ser impresso ao final. Todos os códigos estão sintaticamente corretos. A sugestão é que depois de ter calculado o valor na ponta do lápis, você confira o resultado com um computador real.

Exemplos

Seja o programa

```
def AA():
    A=5;B=5;C=3
    if C>=18:
        while B<=18:
            C=C-3
            B=B+4
        else:
            C=C+4
            B=B+2
    print(C+C+B)
AA()
```

Cujo resultado é 21 e seja o seguinte código

```
def AA():
    A=4;B=3;C=6
    D=1
    C=C+6+3
    while C>6:
        B=D+7
        while A<=14:
            D=B+4
            A=A+2
            D=B+19
            C=C-1
        print(C+B+B)
AA()
```

cujo resultado é 438.

☞ Para você fazer

Exercício 1

```
def AA():
    A=7;B=8;C=7
    while B<=19:
        C=C-8
        A=A+1
        B=B+1
    print(C+C+B)
AA()
```

Informe no quadro próprio o valor que foi impresso.

Exercício 2

```
def AA():
    A=5;B=8;C=2
    if B>16:
        while C<16:
            B=B+5
            C=C+4
        else:
            A=A-4
            C=C+1
    print(B+B+C)
AA()
```

Exercício 3

```
def AA():
    A=7;B=6;C=2
    if A>14:
        while B<14:
            A=C+8
            B=B+4
        else:
            while B<13:
                C=A-4
                B=B+3
            print(A+C+B)
AA()
```

Exercício 4

```
def AA():
    A=7;B=8;C=1
    D=4
    A=A+1+8
    while A>4:
        C=A+6
        while B<15:
            D=C+2
            B=B+2
            D=D+14
            A=A-1
        print(D+D+D)
AA()
```

Exercício 5

```
def AA():
    A=4;B=5;C=8
    D=3
    B=B+8+5
    while B>1:
        A=A+2
        if B<=8:
            while C<19:
                D=D-4
                C=C+3
                D=D+18
            else:
                A=A+10
                B=B-4
        print(C+B)
AA()
```

Exercício 6

```
def AA():
    A=5;B=1;C=7
    D=4
    D=D+7+1
    while D>=3:
        A=A-3
        if B>8:
            while C<19:
                C=C+2
                B=A-8
            else:
                B=B-13
        if D<5:
            B=B+6
        else:
            B=A-10
            D=D-1
    print(C+B)
AA()
```

Exercício 7

```
def AA():
    A=1;B=3;C=8
    D=2
    C=C+8+3
    while C>2:
        B=B+4
        if D>=5:
```

```
        while A<=14:
            A=A+6
            D=B-6
        else:
            D=D+19
        if B<=10:
            D=D+5
        else:
            B=B+7
            C=C-1
    print(D+D)
AA()
```

Exercício 8

```
def AA():
    A=5;B=4;C=1
    D=2
    D=D+1+4
    while D>1:
        A=C+6
        if B<=9:
            while C<=12:
                B=B+6
                C=C+5
            A=A+15
        else:
            A=A+5
            D=D-4
    print(A+C)
AA()
```

Exercício 9

```
def AA():
    A=7;B=4;C=6
    D=3
    D=D+6+4
    while D>7:
        A=B+6
        while B<16:
            A=C+8
            B=B+6
            C=A+13
            D=D-3
        print(D+A+A)
AA()
```

Exercício 10

```
def AA():
    A=8;B=8;C=3
    if B!=17:
        while A<17:
            B=C+3
            A=A+3
        else:
            while A<12:
                C=C-6
                A=A+2
            print(B+C+A)
AA()
```

Responda aqui:

1	2
3	4
5	6
7	8
9	10



Repetição em Python

Como qualquer linguagem estruturada moderna, Python tem as ferramentas básicas para interar. São elas: while, for, bem como as saídas break e continue. A mais amigável é while. Já a a mais potente é for, pois funciona como um interador, que é uma extensão e tanto ao conceito de indexador. Já os comandos break permitem o abandono de um ciclo, enquanto continue serve para abandonar a interação atual. Veja alguns exemplos:

```
i=3
while i<10:
    print(i,' ')
    if i%7==0:
        break
    i=i+1
print('acabou')
```

```
i=3
while i<10:
    print(i,' ')
    if i%7==0:
        continue
    i=i+1
print('acabou')
```

```
s="Curitiba - Paraná - Brasil"
qtd=0
for c in s:
    if c=="a":
        qtd=qtd+1
print("achei ",qtd," a")
```

Exercícios Nos exercícios a seguir, você deve simular o interpretador Python e descobrir qual valor vai ser impresso ao final. Todos os códigos estão sintaticamente corretos. A sugestão é que depois de ter calculado o valor na ponta do lápis, você confira o resultado com um computador real.

Exemplos

Seja o programa

```
def AA():
    A=5;B=5;C=3
    if C>=18:
        while B<=18:
            C=C-3
            B=B+4
        else:
            C=C+4
            B=B+2
            print(C+C+B)
AA()
```

Cujo resultado é 21 e seja o seguinte código

```
def AA():
    A=4;B=3;C=6
    D=1
    C=C+6+3
    while C>6:
        B=D+7
        while A<=14:
            D=B+4
            A=A+2
            D=B+19
            C=C-1
            print(C+B+B)
AA()
```

cujo resultado é 438.

🔗 Para você fazer

Exercício 1

```
def AA():
    A=2;B=6;C=8
    while B<15:
        A=C+2
        C=A+6
        B=B+1
        print(A+C+B)
AA()
```

Informe no quadro próprio o valor que foi impresso.

Exercício 2

```
def AA():
    A=5;B=1;C=2
    if B<19:
        while A<=19:
            B=B+1
            A=A+4
        else:
            C=C+4
            A=A+2
            print(B+B+A)
AA()
```

Exercício 3

```
def AA():
    A=5;B=3;C=3
    if B>12:
        while A<12:
            B=C+2
            A=A+1
        else:
            while A<13:
                B=C-2
                A=A+2
                print(B+C+A)
AA()
```

Exercício 4

```
def AA():
    A=1;B=2;C=3
    D=5
    A=A+3+2
    while A>6:
        B=B+6
        while C<12:
            B=B+2
            C=C+1
            D=D+12
            A=A-4
            print(B+C+C)
AA()
```

Exercício 5

```
def AA():
    A=8;B=5;C=4
    D=3
    C=C+4+5
    while C>3:
        B=C-3
        if A>7:
            while A<=17:
                B=D+4
                A=A+1
                B=D+19
            else:
                B=D+4
                C=C-1
                print(B+A)
AA()
```

Exercício 6

```
def AA():
    A=7;B=3;C=1
    D=4
    B=B+1+3
    while B>2:
        A=B+6
        if D>9:
            while C<=16:
                C=C+3
                A=A+4
            else:
                D=A-19
            if B<=9:
                A=A+5
            else:
                D=A-7
                B=B-3
                print(C+D)
AA()
```

Exercício 7

```
def AA():
    A=6;B=3;C=5
    D=7
    D=D+5+3
    while D>=7:
        A=A+2
        if A<4:
```

```
        while B<18:
            B=B+1
            A=A+6
        else:
            A=A+18
        if B>9:
            C=C+9
        else:
            A=A+5
            D=D-1
            print(A+D)
AA()
```

Exercício 8

```
def AA():
    A=2;B=5;C=8
    D=7
    B=B+8+5
    while B>7:
        A=B+2
        if A<=8:
            while C<=17:
                D=D-3
                C=C+6
                A=A+13
            else:
                A=A+7
                B=B-2
                print(D+A)
AA()
```

Exercício 9

```
def AA():
    A=5;B=3;C=8
    D=4
    B=B+8+3
    while B>=6:
        A=B+5
        while D<=19:
            A=A-7
            D=D+2
            A=A+19
            B=B-1
            print(D+B+D)
AA()
```

Exercício 10

```
def AA():
    A=2;B=6;C=5
    if A==20:
        while C<=20:
            A=B+2
            C=C+1
        else:
            while C<=15:
                B=A-5
                C=C+2
                print(A+B+C)
AA()
```

Responda aqui:

1	2
3	4
5	6
7	8
9	10



Repetição em Python

Como qualquer linguagem estruturada moderna, Python tem as ferramentas básicas para interar. São elas: while, for, bem como as saídas break e continue. A mais amigável é while. Já a a mais potente é for, pois funciona como um interador, que é uma extensão e tanto ao conceito de indexador. Já os comandos break permitem o abandono de um ciclo, enquanto continue serve para abandonar a interação atual. Veja alguns exemplos:

```
i=3
while i<10:
    print(i,' ')
    if i%7==0:
        break
    i=i+1
print('acabou')
```

```
i=3
while i<10:
    print(i,' ')
    if i%7==0:
        continue
    i=i+1
print('acabou')
```

```
s="Curitiba - Paraná - Brasil"
qtd=0
for c in s:
    if c=="a":
        qtd=qtd+1
print("achei ",qtd," a")
```

Exercícios Nos exercícios a seguir, você deve simular o interpretador Python e descobrir qual valor vai ser impresso ao final. Todos os códigos estão sintaticamente corretos. A sugestão é que depois de ter calculado o valor na ponta do lápis, você confira o resultado com um computador real.

Exemplos

Seja o programa

```
def AA():
    A=5;B=5;C=3
    if C>=18:
        while B<=18:
            C=C-3
            B=B+4
        else:
            C=C+4
            B=B+2
    print(C+C+B)
AA()
```

Cujo resultado é 21 e seja o seguinte código

```
def AA():
    A=4;B=3;C=6
    D=1
    C=C+6+3
    while C>6:
        B=D+7
        while A<=14:
            D=B+4
            A=A+2
            D=B+19
            C=C-1
        print(C+B+B)
AA()
```

cujo resultado é 438.

☞ Para você fazer

Exercício 1

```
def AA():
    A=5;B=4;C=7
    while A<17:
        C=C+6
        B=C+4
        A=A+1
    print(C+C+A)
AA()
```

Informe no quadro próprio o valor que foi impresso.

Exercício 2

```
def AA():
    A=3;B=2;C=8
    if B<16:
        while C<16:
            B=B-6
            C=C+4
        else:
            A=A+5
            C=C+1
    print(B+B+C)
AA()
```

Exercício 3

```
def AA():
    A=4;B=5;C=4
    if B>=15:
        while C<=15:
            B=B+5
            C=C+4
        else:
            while C<=15:
                B=A+4
                C=C+2
            print(B+B+C)
AA()
```

Exercício 4

```
def AA():
    A=6;B=4;C=1
    D=7
    B=B+1+4
    while B>2:
        A=B+1
        while D<17:
            A=C+8
            D=D+4
        A=A+18
        B=B-2
    print(A+C+B)
AA()
```

Exercício 5

```
def AA():
    A=7;B=8;C=5
    D=4
    B=B+5+8
    while B>2:
        C=C+6
        if A>=5:
            while A<=13:
                D=C+2
                A=A+6
            D=D+17
        else:
            D=D+9
            B=B-2
    print(D+D)
AA()
```

Exercício 6

```
def AA():
    A=3;B=8;C=5
    D=7
    D=D+5+8
    while D>4:
        A=B+8
        if B>4:
            while B<=17:
                B=B+5
                C=A-8
            else:
                A=C-17
        if D>7:
            C=C+7
        else:
            C=C-9
            D=D-1
    print(C+B)
AA()
```

Exercício 7

```
def AA():
    A=6;B=7;C=2
    D=5
    C=C+2+7
    while C>7:
        A=D+1
        if C<9:
```

```
        while D<=12:
            D=D+6
            A=A+1
        else:
            B=A+19
        if C<9:
            B=B+4
        else:
            A=A+5
            C=C-2
    print(D+D)
AA()
```

Exercício 8

```
def AA():
    A=5;B=6;C=8
    D=4
    D=D+8+6
    while D>=4:
        B=B+6
        if C<9:
            while A<13:
                B=B-3
                A=A+6
            C=B+12
        else:
            B=C+6
            D=D-3
    print(A+C)
AA()
```

Exercício 9

```
def AA():
    A=7;B=5;C=8
    D=2
    D=D+8+5
    while D>=6:
        A=A-4
        while B<=13:
            C=A+1
            B=B+2
        A=A+17
        D=D-3
    print(C+A+C)
AA()
```

Exercício 10

```
def AA():
    A=3;B=4;C=3
    if A=18:
        while C<18:
            A=B+2
            C=C+2
        else:
            while C<12:
                B=B+3
                C=C+2
            print(A+B+C)
AA()
```

Responda aqui:

1	2
3	4
5	6
7	8
9	10



Repetição em Python

Como qualquer linguagem estruturada moderna, Python tem as ferramentas básicas para interar. São elas: while, for, bem como as saídas break e continue. A mais amigável é while. Já a a mais potente é for, pois funciona como um interador, que é uma extensão e tanto ao conceito de indexador. Já os comandos break permitem o abandono de um ciclo, enquanto continue serve para abandonar a interação atual. Veja alguns exemplos:

```
i=3
while i<10:
    print(i,' ')
    if i%7==0:
        break
    i=i+1
print('acabou')
```

```
i=3
while i<10:
    print(i,' ')
    if i%7==0:
        continue
    i=i+1
print('acabou')
```

```
s="Curitiba - Paraná - Brasil"
qtd=0
for c in s:
    if c=="a":
        qtd=qtd+1
print("achei ",qtd," a")
```

Exercícios Nos exercícios a seguir, você deve simular o interpretador Python e descobrir qual valor vai ser impresso ao final. Todos os códigos estão sintaticamente corretos. A sugestão é que depois de ter calculado o valor na ponta do lápis, você confira o resultado com um computador real.

Exemplos

Seja o programa

```
def AA():
    A=5;B=5;C=3
    if C>=18:
        while B<=18:
            C=C-3
            B=B+4
        else:
            C=C+4
            B=B+2
        print(C+C+B)
AA()
```

Cujo resultado é 21 e seja o seguinte código

```
def AA():
    A=4;B=3;C=6
    D=1
    C=C+6+3
    while C>6:
        B=D+7
        while A<=14:
            D=B+4
            A=A+2
            D=B+19
            C=C-1
        print(C+B+B)
AA()
```

cujo resultado é 438.

🔗 Para você fazer

Exercício 1

```
def AA():
    A=5;B=3;C=4
    while A<=12:
        B=B-3
        B=B+8
        A=A+3
        print(B+B+A)
AA()
```

Informe no quadro próprio o valor que foi impresso.

Exercício 2

```
def AA():
    A=3;B=6;C=1
    if A>=12:
        while C<=12:
            A=A-4
            C=C+4
        else:
            B=B+2
            C=C+4
        print(A+A+C)
AA()
```

Exercício 3

```
def AA():
    A=6;B=2;C=1
    if B>=17:
        while A<=17:
            B=C-3
            A=A+1
        else:
            while A<=17:
                B=C+6
                A=A+3
            print(B+C+A)
AA()
```

Exercício 4

```
def AA():
    A=4;B=6;C=2
    D=1
    A=A+2+6
    while A>=6:
        B=C+8
        while D<16:
            B=C-2
            D=D+2
            B=C+12
            A=A-2
        print(C+B+B)
AA()
```

Exercício 5

```
def AA():
    A=6;B=8;C=4
    D=5
    D=D+4+8
    while D>=5:
        A=D-5
        if A<8:
            while B<=13:
                A=A+8
                B=B+4
                A=C+18
            else:
                A=C+8
                D=D-3
            print(A+D)
AA()
```

Exercício 6

```
def AA():
    A=5;B=1;C=7
    D=2
    B=B+7+1
    while B>5:
        A=D+2
        if B<=9:
            while D<=13:
                D=D+4
                C=A-2
            else:
                C=C-15
        if A>=9:
            A=A+10
        else:
            A=A-8
            B=B-2
        print(D+B)
AA()
```

Exercício 7

```
def AA():
    A=3;B=7;C=4
    D=2
    C=C+4+7
    while C>=3:
        A=B+1
        if C>=5:
```

```
        while D<17:
            D=D+4
            B=B-3
        else:
            A=B+13
        if D>10:
            B=B+5
        else:
            A=B+4
            C=C-4
        print(A+A)
AA()
```

Exercício 8

```
def AA():
    A=5;B=7;C=3
    D=4
    A=A+3+7
    while A>=4:
        B=B-7
        if D<4:
            while D<16:
                B=C-7
                D=D+6
                C=C+17
            else:
                B=C+9
                A=A-2
            print(B+A)
AA()
```

Exercício 9

```
def AA():
    A=7;B=8;C=5
    D=2
    C=C+5+8
    while C>=7:
        B=D+2
        while A<12:
            D=B-5
            A=A+6
            B=D-13
            C=C-2
        print(D+A+A)
AA()
```

Exercício 10

```
def AA():
    A=3;B=2;C=7
    if C>=18:
        while B<=18:
            C=C+6
            B=B+2
        else:
            while B<=18:
                A=C+1
                B=B+3
            print(C+C+B)
AA()
```

Responda aqui:

1	2
3	4
5	6
7	8
9	10



Repetição em Python

Como qualquer linguagem estruturada moderna, Python tem as ferramentas básicas para interar. São elas: while, for, bem como as saídas break e continue. A mais amigável é while. Já a a mais potente é for, pois funciona como um interador, que é uma extensão e tanto ao conceito de indexador. Já os comandos break permitem o abandono de um ciclo, enquanto continue serve para abandonar a interação atual. Veja alguns exemplos:

```
i=3
while i<10:
    print(i,' ')
    if i%7==0:
        break
    i=i+1
print('acabou')
```

```
i=3
while i<10:
    print(i,' ')
    if i%7==0:
        continue
    i=i+1
print('acabou')
```

```
s="Curitiba - Paraná - Brasil"
qtd=0
for c in s:
    if c=="a":
        qtd=qtd+1
print("achei ",qtd," a")
```

Exercícios Nos exercícios a seguir, você deve simular o interpretador Python e descobrir qual valor vai ser impresso ao final. Todos os códigos estão sintaticamente corretos. A sugestão é que depois de ter calculado o valor na ponta do lápis, você confira o resultado com um computador real.

Exemplos

Seja o programa

```
def AA():
    A=5;B=5;C=3
    if C>=18:
        while B<=18:
            C=C-3
            B=B+4
        else:
            C=C+4
            B=B+2
        print(C+C+B)
AA()
```

Cujo resultado é 21 e seja o seguinte código

```
def AA():
    A=4;B=3;C=6
    D=1
    C=C+6+3
    while C>6:
        B=D+7
        while A<=14:
            D=B+4
            A=A+2
            D=B+19
            C=C-1
        print(C+B+B)
AA()
```

cujo resultado é 438.

🔗 Para você fazer

Exercício 1

```
def AA():
    A=8;B=8;C=5
    while B<=14:
        A=C+7
        C=C-8
        B=B+2
        print(A+C+B)
AA()
```

Informe no quadro próprio o valor que foi impresso.

Exercício 2

```
def AA():
    A=4;B=1;C=3
    if A==18:
        while C<=18:
            A=B+8
            C=C+4
        else:
            B=B-2
            C=C+3
        print(A+B+C)
AA()
```

Exercício 3

```
def AA():
    A=6;B=5;C=4
    if B<18:
        while C<18:
            B=A-3
            C=C+3
        else:
            while C<18:
                A=B+2
                C=C+1
            print(B+A+C)
AA()
```

Exercício 4

```
def AA():
    A=6;B=5;C=8
    D=4
    B=B+8+5
    while B>=6:
        A=A+5
        while C<15:
            D=A-3
            C=C+5
            A=D-12
            B=B-3
        print(C+A+C)
AA()
```

Exercício 5

```
def AA():
    A=1;B=5;C=8
    D=7
    B=B+8+5
    while B>6:
        A=C+7
        if C>9:
            while C<18:
                A=D+8
                C=C+5
                D=A+14
            else:
                A=A+7
                B=B-3
        print(C+C)
AA()
```

Exercício 6

```
def AA():
    A=5;B=8;C=7
    D=6
    D=D+7+8
    while D>=7:
        A=B+1
        if A>=6:
            while C<=16:
                C=C+3
                A=B+1
            else:
                A=A+19
        if D<=6:
            B=B+4
        else:
            B=A+8
            D=D-1
        print(C+A)
AA()
```

Exercício 7

```
def AA():
    A=6;B=1;C=4
    D=2
    B=B+4+1
    while B>3:
        A=A+6
        if D>5:
```

```
        while D<19:
            D=D+4
            A=C+5
        else:
            A=C+17
        if B<=5:
            A=A+8
        else:
            A=A+8
            B=B-3
        print(A+B)
AA()
```

Exercício 8

```
def AA():
    A=2;B=8;C=1
    D=6
    A=A+1+8
    while A>=6:
        B=B-3
        if B<=9:
            while C<18:
                D=B+5
                C=C+4
                D=D+13
            else:
                D=D+4
                A=A-2
            print(C+C)
AA()
```

Exercício 9

```
def AA():
    A=7;B=6;C=4
    D=5
    D=D+4+6
    while D>=6:
        B=C-3
        while A<=16:
            B=B-8
            A=A+3
            C=C+14
            D=D-1
        print(C+A+A)
AA()
```

Exercício 10

```
def AA():
    A=7;B=1;C=6
    if C>=19:
        while B<19:
            C=C-6
            B=B+1
        else:
            while B<15:
                C=A+1
                B=B+2
            print(C+C+B)
AA()
```

Responda aqui:

1	2
3	4
5	6
7	8
9	10



Repetição em Python

Como qualquer linguagem estruturada moderna, Python tem as ferramentas básicas para interar. São elas: while, for, bem como as saídas break e continue. A mais amigável é while. Já a a mais potente é for, pois funciona como um interador, que é uma extensão e tanto ao conceito de indexador. Já os comandos break permitem o abandono de um ciclo, enquanto continue serve para abandonar a interação atual. Veja alguns exemplos:

```
i=3
while i<10:
    print(i,' ')
    if i%7==0:
        break
    i=i+1
print('acabou')
```

```
i=3
while i<10:
    print(i,' ')
    if i%7==0:
        continue
    i=i+1
print('acabou')
```

```
s="Curitiba - Paraná - Brasil"
qtd=0
for c in s:
    if c=="a":
        qtd=qtd+1
print("achei ",qtd," a")
```

Exercícios Nos exercícios a seguir, você deve simular o interpretador Python e descobrir qual valor vai ser impresso ao final. Todos os códigos estão sintaticamente corretos. A sugestão é que depois de ter calculado o valor na ponta do lápis, você confira o resultado com um computador real.

Exemplos

Seja o programa

```
def AA():
    A=5;B=5;C=3
    if C>=18:
        while B<=18:
            C=C-3
            B=B+4
        else:
            C=C+4
            B=B+2
            print(C+C+B)
AA()
```

Cujo resultado é 21 e seja o seguinte código

```
def AA():
    A=4;B=3;C=6
    D=1
    C=C+6+3
    while C>6:
        B=D+7
        while A<=14:
            D=B+4
            A=A+2
            D=B+19
            C=C-1
        print(C+B+B)
AA()
```

cujo resultado é 438.

🔗 Para você fazer

Exercício 1

```
def AA():
    A=5;B=1;C=1
    while B<=15:
        A=A+4
        C=A+1
        B=B+4
        print(A+A+B)
AA()
```

Informe no quadro próprio o valor que foi impresso.

Exercício 2

```
def AA():
    A=8;B=2;C=8
    if C>=18:
        while A<=18:
            C=C+6
            A=A+4
        else:
            C=C-5
            A=A+3
            print(C+C+A)
AA()
```

Exercício 3

```
def AA():
    A=3;B=8;C=2
    if A!=17:
        while C<=17:
            A=A-2
            C=C+4
        else:
            while C<=18:
                B=A+4
                C=C+3
                print(A+A+C)
AA()
```

Exercício 4

```
def AA():
    A=2;B=6;C=3
    D=1
    D=D+3+6
    while D>=6:
        B=D+3
        while A<13:
            B=C-8
            A=A+1
            B=B-16
            D=D-3
        print(A+C+C)
AA()
```

Exercício 5

```
def AA():
    A=6;B=1;C=3
    D=8
    B=B+3+1
    while B>6:
        A=D+7
        if D>=5:
            while C<15:
                A=D+1
                C=C+1
                A=D-14
            else:
                D=D-10
                B=B-3
                print(D+B)
AA()
```

Exercício 6

```
def AA():
    A=6;B=1;C=8
    D=4
    D=D+8+1
    while D>5:
        A=D+2
        if C>4:
            while C<=15:
                C=C+6
                A=A+5
            else:
                A=A+18
            if D<=5:
                B=B+7
            else:
                B=A+5
                D=D-4
            print(B+C)
AA()
```

Exercício 7

```
def AA():
    A=2;B=4;C=6
    D=8
    C=C+6+4
    while C>=4:
        B=C+7
        if A>=10:
```

```
        while A<=17:
            A=A+3
            B=B+2
        else:
            B=B-18
        if C<=6:
            D=D+7
        else:
            B=B-9
            C=C-2
        print(A+B)
AA()
```

Exercício 8

```
def AA():
    A=3;B=7;C=6
    D=4
    A=A+6+7
    while A>=3:
        C=A+3
        if C<7:
            while B<=13:
                D=C+7
                B=B+3
                D=D+19
            else:
                D=C+6
                A=A-3
                print(C+A)
AA()
```

Exercício 9

```
def AA():
    A=1;B=7;C=8
    D=4
    D=D+8+7
    while D>=1:
        A=A+4
        while B<15:
            A=C+2
            B=B+4
            A=C+19
            D=D-3
        print(A+C+A)
AA()
```

Exercício 10

```
def AA():
    A=5;B=7;C=8
    if B>=19:
        while A<19:
            B=B+6
            A=A+2
        else:
            while A<12:
                C=C-4
                A=A+1
                print(B+B+A)
AA()
```

Responda aqui:

1	2
3	4
5	6
7	8
9	10



Repetição em Python

Como qualquer linguagem estruturada moderna, Python tem as ferramentas básicas para interar. São elas: while, for, bem como as saídas break e continue. A mais amigável é while. Já a a mais potente é for, pois funciona como um interador, que é uma extensão e tanto ao conceito de indexador. Já os comandos break permitem o abandono de um ciclo, enquanto continue serve para abandonar a interação atual. Veja alguns exemplos:

```
i=3
while i<10:
    print(i,' ')
    if i%7==0:
        break
    i=i+1
print('acabou')
```

```
i=3
while i<10:
    print(i,' ')
    if i%7==0:
        continue
    i=i+1
print('acabou')
```

```
s="Curitiba - Paraná - Brasil"
qtd=0
for c in s:
    if c=="a":
        qtd=qtd+1
print("achei ",qtd," a")
```

Exercícios Nos exercícios a seguir, você deve simular o interpretador Python e descobrir qual valor vai ser impresso ao final. Todos os códigos estão sintaticamente corretos. A sugestão é que depois de ter calculado o valor na ponta do lápis, você confira o resultado com um computador real.

Exemplos

Seja o programa

```
def AA():
    A=5;B=5;C=3
    if C>=18:
        while B<=18:
            C=C-3
            B=B+4
        else:
            C=C+4
            B=B+2
    print(C+C+B)
AA()
```

Cujo resultado é 21 e seja o seguinte código

```
def AA():
    A=4;B=3;C=6
    D=1
    C=C+6+3
    while C>6:
        B=D+7
        while A<=14:
            D=B+4
            A=A+2
            D=B+19
            C=C-1
        print(C+B+B)
AA()
```

cujo resultado é 438.

🔗 Para você fazer

Exercício 1

```
def AA():
    A=3;B=8;C=5
    while B<=13:
        C=A-7
        A=A+4
        B=B+4
    print(C+A+B)
AA()
```

Informe no quadro próprio o valor que foi impresso.

Exercício 2

```
def AA():
    A=7;B=8;C=7
    if B>17:
        while C<17:
            B=A+5
            C=C+4
        else:
            A=A+7
            C=C+2
    print(B+A+C)
AA()
```

Exercício 3

```
def AA():
    A=3;B=7;C=5
    if A==15:
        while B<15:
            A=A+1
            B=B+1
        else:
            while B<17:
                C=C-5
                B=B+3
            print(A+A+B)
AA()
```

Exercício 4

```
def AA():
    A=3;B=6;C=8
    D=4
    B=B+8+6
    while B>=4:
        C=B-6
        while A<19:
            C=D+8
            A=A+5
            C=C+18
            B=B-2
        print(D+A+D)
AA()
```

Exercício 5

```
def AA():
    A=5;B=8;C=4
    D=3
    D=D+4+8
    while D>6:
        A=D+7
        if A<=5:
            while C<=19:
                A=B-6
                C=C+5
                B=A+16
            else:
                A=A+4
                D=D-3
        print(C+C)
AA()
```

Exercício 6

```
def AA():
    A=6;B=4;C=3
    D=8
    C=C+3+4
    while C>1:
        B=B+7
        if A<=9:
            while A<=19:
                A=A+3
                B=B+7
            else:
                D=B-17
        if D>=6:
            B=B+10
        else:
            B=D-9
            C=C-1
        print(C+C)
AA()
```

Exercício 7

```
def AA():
    A=6;B=7;C=2
    D=4
    B=B+2+7
    while B>1:
        A=D+2
        if D<7:
```

```
        while C<=13:
            C=C+1
            A=D+1
        else:
            D=A+18
        if A<8:
            A=A+9
        else:
            D=A+7
            B=B-4
        print(B+C)
AA()
```

Exercício 8

```
def AA():
    A=7;B=2;C=5
    D=3
    B=B+5+2
    while B>=5:
        A=C-6
        if A>=7:
            while D<=14:
                C=C+7
                D=D+4
            A=C+15
        else:
            C=A+5
            B=B-2
        print(A+D)
AA()
```

Exercício 9

```
def AA():
    A=3;B=2;C=1
    D=7
    B=B+1+2
    while B>=1:
        A=A-6
        while C<=18:
            A=A-6
            C=C+5
            A=A+16
            B=B-3
        print(C+B+B)
AA()
```

Exercício 10

```
def AA():
    A=3;B=8;C=5
    if C<20:
        while A<=20:
            C=B+1
            A=A+1
        else:
            while A<=16:
                C=B-8
                A=A+2
            print(C+B+A)
AA()
```

Responda aqui:

1	2
3	4
5	6
7	8
9	10



Repetição em Python

Como qualquer linguagem estruturada moderna, Python tem as ferramentas básicas para interar. São elas: while, for, bem como as saídas break e continue. A mais amigável é while. Já a a mais potente é for, pois funciona como um interador, que é uma extensão e tanto ao conceito de indexador. Já os comandos break permitem o abandono de um ciclo, enquanto continue serve para abandonar a interação atual. Veja alguns exemplos:

```
i=3
while i<10:
    print(i,' ')
    if i%7==0:
        break
    i=i+1
print('acabou')
```

```
i=3
while i<10:
    print(i,' ')
    if i%7==0:
        continue
    i=i+1
print('acabou')
```

```
s="Curitiba - Paraná - Brasil"
qtd=0
for c in s:
    if c=="a":
        qtd=qtd+1
print("achei ",qtd," a")
```

Exercícios Nos exercícios a seguir, você deve simular o interpretador Python e descobrir qual valor vai ser impresso ao final. Todos os códigos estão sintaticamente corretos. A sugestão é que depois de ter calculado o valor na ponta do lápis, você confira o resultado com um computador real.

Exemplos

Seja o programa

```
def AA():
    A=5;B=5;C=3
    if C>=18:
        while B<=18:
            C=C-3
            B=B+4
        else:
            C=C+4
            B=B+2
    print(C+C+B)
AA()
```

Cujo resultado é 21 e seja o seguinte código

```
def AA():
    A=4;B=3;C=6
    D=1
    C=C+6+3
    while C>6:
        B=D+7
        while A<=14:
            D=B+4
            A=A+2
            D=B+19
            C=C-1
        print(C+B+B)
AA()
```

cujo resultado é 438.

🔗 Para você fazer

Exercício 1

```
def AA():
    A=1;B=2;C=5
    while B<=14:
        C=A+7
        C=A-8
        B=B+2
    print(C+A+B)
AA()
```

Informe no quadro próprio o valor que foi impresso.

Exercício 2

```
def AA():
    A=1;B=5;C=5
    if C!=14:
        while A<=14:
            C=C+4
            A=A+4
        else:
            C=B-6
            A=A+4
    print(C+C+A)
AA()
```

Exercício 3

```
def AA():
    A=7;B=1;C=8
    if A<18:
        while C<18:
            A=A-3
            C=C+4
        else:
            while C<14:
                B=B+3
                C=C+3
            print(A+A+C)
AA()
```

Exercício 4

```
def AA():
    A=2;B=8;C=7
    D=4
    B=B+7+8
    while B>=3:
        A=C+4
        while D<18:
            A=C+6
            D=D+1
        C=A+16
        B=B-2
    print(C+A+D)
AA()
```

Exercício 5

```
def AA():
    A=6;B=5;C=4
    D=7
    B=B+4+5
    while B>=7:
        A=C+1
        if A>=10:
            while D<15:
                C=C-7
                D=D+6
            A=A+13
        else:
            C=C+5
            B=B-3
    print(D+A)
AA()
```

Exercício 6

```
def AA():
    A=4;B=3;C=8
    D=1
    C=C+8+3
    while C>=6:
        A=A+3
        if C>4:
            while D<=16:
                D=D+4
            A=B+2
        else:
            A=B+12
        if C>=10:
            B=B+4
        else:
            B=A+7
            C=C-3
    print(B+C)
AA()
```

Exercício 7

```
def AA():
    A=4;B=8;C=2
    D=3
    A=A+2+8
    while A>1:
        C=D+5
        if D>=5:
```

```
        while B<=19:
            B=B+4
            C=D+3
        else:
            C=C+15
        if C<=7:
            C=C+6
        else:
            D=D+10
        A=A-4
    print(B+A)
AA()
```

Exercício 8

```
def AA():
    A=2;B=5;C=4
    D=6
    A=A+4+5
    while A>2:
        B=B+6
        if A<=4:
            while C<=17:
                D=B-1
                C=C+5
            B=D+14
        else:
            B=D+9
        A=A-2
    print(C+D)
AA()
```

Exercício 9

```
def AA():
    A=6;B=8;C=7
    D=1
    A=A+7+8
    while A>2:
        B=C+3
        while D<17:
            B=B+4
            D=D+1
        B=C+18
        A=A-2
    print(C+C+D)
AA()
```

Exercício 10

```
def AA():
    A=1;B=8;C=3
    if C=18:
        while B<18:
            C=C+5
            B=B+2
        else:
            while B<17:
                A=C+3
                B=B+3
            print(C+C+B)
AA()
```

Responda aqui:

1	2
3	4
5	6
7	8
9	10



Repetição em Python

Como qualquer linguagem estruturada moderna, Python tem as ferramentas básicas para interar. São elas: while, for, bem como as saídas break e continue. A mais amigável é while. Já a a mais potente é for, pois funciona como um interador, que é uma extensão e tanto ao conceito de indexador. Já os comandos break permitem o abandono de um ciclo, enquanto continue serve para abandonar a interação atual. Veja alguns exemplos:

```
i=3
while i<10:
    print(i,' ')
    if i%7==0:
        break
    i=i+1
print('acabou')
```

```
i=3
while i<10:
    print(i,' ')
    if i%7==0:
        continue
    i=i+1
print('acabou')
```

```
s="Curitiba - Paraná - Brasil"
qtd=0
for c in s:
    if c=="a":
        qtd=qtd+1
print("achei ",qtd," a")
```

Exercícios Nos exercícios a seguir, você deve simular o interpretador Python e descobrir qual valor vai ser impresso ao final. Todos os códigos estão sintaticamente corretos. A sugestão é que depois de ter calculado o valor na ponta do lápis, você confira o resultado com um computador real.

Exemplos

Seja o programa

```
def AA():
    A=5;B=5;C=3
    if C>=18:
        while B<=18:
            C=C-3
            B=B+4
    else:
        C=C+4
        B=B+2
    print(C+C+B)
AA()
```

Cujo resultado é 21 e seja o seguinte código

```
def AA():
    A=4;B=3;C=6
    D=1
    C=C+6+3
    while C>6:
        B=D+7
        while A<=14:
            D=B+4
            A=A+2
            D=B+19
            C=C-1
        print(C+B+B)
AA()
```

cujo resultado é 438.

🔗 Para você fazer

Exercício 1

```
def AA():
    A=4;B=3;C=5
    while A<=13:
        C=B+1
        B=C+3
        A=A+2
        print(C+B+A)
AA()
```

Informe no quadro próprio o valor que foi impresso.

Exercício 2

```
def AA():
    A=6;B=7;C=5
    if B<19:
        while A<19:
            B=B+3
            A=A+4
        else:
            C=C+6
            A=A+2
        print(B+B+A)
AA()
```

Exercício 3

```
def AA():
    A=7;B=5;C=5
    if C<=12:
        while A<=12:
            C=C+2
            A=A+1
        else:
            while A<=16:
                B=C+2
                A=A+3
            print(C+C+A)
AA()
```

Exercício 4

```
def AA():
    A=8;B=2;C=3
    D=5
    D=D+3+2
    while D>=6:
        B=B+6
        while A<=12:
            C=C-2
            A=A+2
            B=B+18
            D=D-3
        print(A+D+C)
AA()
```

Exercício 5

```
def AA():
    A=7;B=2;C=5
    D=1
    D=D+5+2
    while D>2:
        A=A-3
        if C<=6:
            while B<17:
                A=C-8
                B=B+1
            A=A+17
        else:
            C=A+6
            D=D-4
        print(B+C)
AA()
```

Exercício 6

```
def AA():
    A=3;B=7;C=1
    D=6
    C=C+1+7
    while C>4:
        A=C+4
        if D<=8:
            while B<13:
                B=B+1
            D=D-8
        else:
            D=D-14
        if C<7:
            D=D+4
        else:
            A=D-9
            C=C-1
        print(B+A)
AA()
```

Exercício 7

```
def AA():
    A=1;B=4;C=8
    D=6
    A=A+8+4
    while A>4:
        B=B+6
        if D<=8:
```

```
        while D<17:
            D=D+5
            C=C+1
        else:
            C=C+16
        if C<4:
            C=C+10
        else:
            B=B+6
            A=A-1
        print(A+C)
AA()
```

Exercício 8

```
def AA():
    A=2;B=6;C=4
    D=3
    A=A+4+6
    while A>=3:
        B=B-2
        if B<=5:
            while C<19:
                D=D+6
                C=C+2
            B=B+15
        else:
            B=D+7
            A=A-3
        print(A+B)
AA()
```

Exercício 9

```
def AA():
    A=3;B=1;C=6
    D=2
    A=A+6+1
    while A>1:
        C=D+6
        while B<=15:
            C=D+2
            B=B+5
            C=D-12
            A=A-1
        print(C+A+D)
AA()
```

Exercício 10

```
def AA():
    A=1;B=4;C=8
    if B>=15:
        while A<15:
            B=C+2
            A=A+4
    else:
        while A<18:
            B=C+3
            A=A+2
        print(B+C+A)
AA()
```

Responda aqui:

1	2
3	4
5	6
7	8
9	10



Repetição em Python

Como qualquer linguagem estruturada moderna, Python tem as ferramentas básicas para interar. São elas: while, for, bem como as saídas break e continue. A mais amigável é while. Já a a mais potente é for, pois funciona como um interador, que é uma extensão e tanto ao conceito de indexador. Já os comandos break permitem o abandono de um ciclo, enquanto continue serve para abandonar a interação atual. Veja alguns exemplos:

```
i=3
while i<10:
    print(i,' ')
    if i%7==0:
        break
    i=i+1
print('acabou')
```

```
i=3
while i<10:
    print(i,' ')
    if i%7==0:
        continue
    i=i+1
print('acabou')
```

```
s="Curitiba - Paraná - Brasil"
qtd=0
for c in s:
    if c=="a":
        qtd=qtd+1
print("achei ",qtd," a")
```

Exercícios Nos exercícios a seguir, você deve simular o interpretador Python e descobrir qual valor vai ser impresso ao final. Todos os códigos estão sintaticamente corretos. A sugestão é que depois de ter calculado o valor na ponta do lápis, você confira o resultado com um computador real.

Exemplos

Seja o programa

```
def AA():
    A=5;B=5;C=3
    if C>=18:
        while B<=18:
            C=C-3
            B=B+4
        else:
            C=C+4
            B=B+2
        print(C+C+B)
AA()
```

Cujo resultado é 21 e seja o seguinte código

```
def AA():
    A=4;B=3;C=6
    D=1
    C=C+6+3
    while C>6:
        B=D+7
        while A<=14:
            D=B+4
            A=A+2
            D=B+19
            C=C-1
        print(C+B+B)
AA()
```

cujo resultado é 438.

🔗 Para você fazer

Exercício 1

```
def AA():
    A=7;B=1;C=1
    while C<16:
        B=B+5
        A=A+1
        C=C+1
        print(B+B+C)
AA()
```

Informe no quadro próprio o valor que foi impresso.

Exercício 2

```
def AA():
    A=7;B=4;C=2
    if A>=19:
        while B<19:
            A=A+2
            B=B+4
        else:
            C=C+1
            B=B+1
        print(A+A+B)
AA()
```

Exercício 3

```
def AA():
    A=6;B=4;C=2
    if A<=20:
        while C<=20:
            A=A+4
            C=C+1
        else:
            while C<=18:
                A=A+8
                C=C+3
            print(A+A+C)
AA()
```

Exercício 4

```
def AA():
    A=1;B=5;C=4
    D=2
    A=A+4+5
    while A>5:
        C=C+7
        while B<15:
            C=C+8
            B=B+6
        D=C-12
        A=A-3
        print(C+C+C)
AA()
```

Exercício 5

```
def AA():
    A=5;B=1;C=4
    D=7
    A=A+4+1
    while A>3:
        B=A+6
        if B<=6:
            while C<=18:
                B=D+4
                C=C+4
                D=D+15
            else:
                D=D+4
                A=A-2
        print(A+C)
AA()
```

Exercício 6

```
def AA():
    A=7;B=1;C=6
    D=8
    A=A+6+1
    while A>7:
        B=B-8
        if C>10:
            while C<=12:
                C=C+2
                D=B-6
            else:
                D=D+15
        if A<7:
            D=D+4
        else:
            B=B+8
            A=A-3
        print(A+D)
AA()
```

Exercício 7

```
def AA():
    A=3;B=8;C=5
    D=2
    B=B+5+8
    while B>=3:
        A=B+7
        if A<=5:
```

```
        while C<=17:
            C=C+2
            D=D-2
        else:
            D=A-19
        if D<5:
            D=D+9
        else:
            A=A-7
            B=B-3
        print(B+A)
AA()
```

Exercício 8

```
def AA():
    A=2;B=1;C=8
    D=4
    C=C+8+1
    while C>6:
        B=C-2
        if C<7:
            while A<12:
                D=B-1
                A=A+3
                B=B+16
            else:
                B=B+10
                C=C-1
        print(B+A)
AA()
```

Exercício 9

```
def AA():
    A=4;B=3;C=6
    D=2
    C=C+6+3
    while C>1:
        A=C+4
        while D<18:
            B=B-3
            D=D+2
            B=B+19
            C=C-2
        print(C+D+D)
AA()
```

Exercício 10

```
def AA():
    A=7;B=8;C=2
    if A!=16:
        while C<=16:
            A=A+5
            C=C+2
        else:
            while C<=13:
                A=A+2
                C=C+3
            print(A+A+C)
AA()
```

Responda aqui:

1	2
3	4
5	6
7	8
9	10



Repetição em Python

Como qualquer linguagem estruturada moderna, Python tem as ferramentas básicas para interar. São elas: while, for, bem como as saídas break e continue. A mais amigável é while. Já a a mais potente é for, pois funciona como um interador, que é uma extensão e tanto ao conceito de indexador. Já os comandos break permitem o abandono de um ciclo, enquanto continue serve para abandonar a interação atual. Veja alguns exemplos:

```
i=3
while i<10:
    print(i,' ')
    if i%7==0:
        break
    i=i+1
print('acabou')
```

```
i=3
while i<10:
    print(i,' ')
    if i%7==0:
        continue
    i=i+1
print('acabou')
```

```
s="Curitiba - Paraná - Brasil"
qtd=0
for c in s:
    if c=="a":
        qtd=qtd+1
print("achei ",qtd," a")
```

Exercícios Nos exercícios a seguir, você deve simular o interpretador Python e descobrir qual valor vai ser impresso ao final. Todos os códigos estão sintaticamente corretos. A sugestão é que depois de ter calculado o valor na ponta do lápis, você confira o resultado com um computador real.

Exemplos

Seja o programa

```
def AA():
    A=5;B=5;C=3
    if C>=18:
        while B<=18:
            C=C-3
            B=B+4
        else:
            C=C+4
            B=B+2
    print(C+C+B)
AA()
```

Cujo resultado é 21 e seja o seguinte código

```
def AA():
    A=4;B=3;C=6
    D=1
    C=C+6+3
    while C>6:
        B=D+7
        while A<=14:
            D=B+4
            A=A+2
            D=B+19
            C=C-1
        print(C+B+B)
AA()
```

cujo resultado é 438.

☞ Para você fazer

Exercício 1

```
def AA():
    A=7;B=5;C=2
    while A<=15:
        B=C-3
        B=B-5
        A=A+3
        print(B+C+A)
AA()
```

Informe no quadro próprio o valor que foi impresso.

Exercício 2

```
def AA():
    A=3;B=5;C=8
    if C>13:
        while A<=13:
            C=B-1
            A=A+4
        else:
            C=B-8
            A=A+2
        print(C+B+A)
AA()
```

Exercício 3

```
def AA():
    A=2;B=7;C=7
    if C<=15:
        while A<15:
            C=B+1
            A=A+2
        else:
            while A<15:
                B=B-8
                A=A+3
            print(C+B+A)
AA()
```

Exercício 4

```
def AA():
    A=7;B=8;C=1
    D=3
    C=C+1+8
    while C>=6:
        A=C-5
        while D<=19:
            A=B-2
            D=D+6
            A=B+13
            C=C-4
        print(D+D+A)
AA()
```

Exercício 5

```
def AA():
    A=6;B=3;C=2
    D=5
    C=C+2+3
    while C>=6:
        B=A+1
        if A>9:
            while A<17:
                B=D+1
                A=A+3
                D=D+16
            else:
                D=B+6
                C=C-4
        print(B+B)
AA()
```

Exercício 6

```
def AA():
    A=8;B=2;C=6
    D=5
    B=B+6+2
    while B>=7:
        A=B+2
        if C<=4:
            while D<=16:
                D=D+4
                A=C+1
            else:
                C=C+17
        if C<5:
            A=A+8
        else:
            A=A+8
            B=B-2
        print(D+A)
AA()
```

Exercício 7

```
def AA():
    A=5;B=3;C=8
    D=7
    C=C+8+3
    while C>=2:
        B=D+7
        if B>9:
```

```
        while A<15:
            A=A+2
            D=D-7
        else:
            B=D-16
        if C<6:
            B=B+7
        else:
            D=D-9
            C=C-3
        print(C+A)
AA()
```

Exercício 8

```
def AA():
    A=7;B=4;C=5
    D=6
    A=A+5+4
    while A>7:
        B=A+4
        if B<=8:
            while D<=14:
                C=B+5
                D=D+4
                C=B-19
            else:
                C=C-6
                A=A-2
            print(B+A)
AA()
```

Exercício 9

```
def AA():
    A=2;B=7;C=3
    D=8
    C=C+3+7
    while C>=1:
        A=B+6
        while B<16:
            D=D-7
            B=B+3
            D=D-16
            C=C-3
        print(C+A+D)
AA()
```

Exercício 10

```
def AA():
    A=5;B=7;C=6
    if A=16:
        while C<16:
            A=A+8
            C=C+3
        else:
            while C<15:
                A=B+8
                C=C+3
            print(A+A+C)
AA()
```

Responda aqui:

1	2
3	4
5	6
7	8
9	10



Repetição em Python

Como qualquer linguagem estruturada moderna, Python tem as ferramentas básicas para interar. São elas: while, for, bem como as saídas break e continue. A mais amigável é while. Já a a mais potente é for, pois funciona como um interador, que é uma extensão e tanto ao conceito de indexador. Já os comandos break permitem o abandono de um ciclo, enquanto continue serve para abandonar a interação atual. Veja alguns exemplos:

```
i=3
while i<10:
    print(i,' ')
    if i%7==0:
        break
    i=i+1
print('acabou')
```

```
i=3
while i<10:
    print(i,' ')
    if i%7==0:
        continue
    i=i+1
print('acabou')
```

```
s="Curitiba - Paraná - Brasil"
qtd=0
for c in s:
    if c=="a":
        qtd=qtd+1
print("achei ",qtd," a")
```

Exercícios Nos exercícios a seguir, você deve simular o interpretador Python e descobrir qual valor vai ser impresso ao final. Todos os códigos estão sintaticamente corretos. A sugestão é que depois de ter calculado o valor na ponta do lápis, você confira o resultado com um computador real.

Exemplos

Seja o programa

```
def AA():
    A=5;B=5;C=3
    if C>=18:
        while B<=18:
            C=C-3
            B=B+4
        else:
            C=C+4
            B=B+2
            print(C+C+B)
AA()
```

Cujo resultado é 21 e seja o seguinte código

```
def AA():
    A=4;B=3;C=6
    D=1
    C=C+6+3
    while C>6:
        B=D+7
        while A<=14:
            D=B+4
            A=A+2
            D=B+19
            C=C-1
        print(C+B+B)
AA()
```

cujo resultado é 438.

🔗 Para você fazer

Exercício 1

```
def AA():
    A=2;B=4;C=3
    while B<=17:
        C=A-6
        C=A-4
        B=B+3
        print(C+A+B)
AA()
```

Informe no quadro próprio o valor que foi impresso.

Exercício 2

```
def AA():
    A=4;B=3;C=3
    if C==18:
        while B<18:
            C=A+5
            B=B+4
        else:
            C=C+7
            B=B+1
            print(C+A+B)
AA()
```

Exercício 3

```
def AA():
    A=2;B=1;C=5
    if A<=18:
        while B<18:
            A=A+6
            B=B+2
        else:
            while B<18:
                C=A+5
                B=B+2
                print(A+A+B)
AA()
```

Exercício 4

```
def AA():
    A=8;B=7;C=6
    D=4
    D=D+6+7
    while D>6:
        A=B+2
        while C<17:
            B=B+3
            C=C+2
            A=A+18
            D=D-4
            print(D+A+C)
AA()
```

Exercício 5

```
def AA():
    A=7;B=2;C=3
    D=8
    D=D+3+2
    while D>=7:
        A=C+5
        if B>10:
            while B<=13:
                A=C+4
                B=B+4
            A=C+12
        else:
            A=A+4
            D=D-2
            print(D+D)
AA()
```

Exercício 6

```
def AA():
    A=8;B=2;C=1
    D=3
    B=B+1+2
    while B>5:
        A=A-3
        if B>=4:
            while C<=12:
                C=C+5
                A=A+4
            else:
                D=D+13
        if B<4:
            A=A+4
        else:
            A=D+5
            B=B-4
            print(D+C)
AA()
```

Exercício 7

```
def AA():
    A=8;B=5;C=6
    D=1
    D=D+6+5
    while D>3:
        A=A-5
        if C>7:
```

```
        while B<=16:
            B=B+6
            A=C-7
        else:
            A=A-16
        if C<=4:
            C=C+4
        else:
            C=C-5
            D=D-3
        print(B+A)
AA()
```

Exercício 8

```
def AA():
    A=2;B=3;C=7
    D=6
    A=A+7+3
    while A>2:
        C=D+6
        if B<=9:
            while B<=17:
                D=C+5
                B=B+5
                D=D+13
            else:
                C=C+7
                A=A-3
                print(C+C)
AA()
```

Exercício 9

```
def AA():
    A=7;B=8;C=5
    D=6
    D=D+5+8
    while D>7:
        A=B+8
        while B<13:
            A=A+1
            B=B+2
            C=C+13
            D=D-4
            print(C+B+A)
AA()
```

Exercício 10

```
def AA():
    A=8;B=4;C=2
    if C==12:
        while B<12:
            C=A+8
            B=B+3
        else:
            while B<18:
                A=C+1
                B=B+3
                print(C+A+B)
AA()
```

Responda aqui:

1	2
3	4
5	6
7	8
9	10



Repetição em Python

Como qualquer linguagem estruturada moderna, Python tem as ferramentas básicas para interar. São elas: while, for, bem como as saídas break e continue. A mais amigável é while. Já a a mais potente é for, pois funciona como um interador, que é uma extensão e tanto ao conceito de indexador. Já os comandos break permitem o abandono de um ciclo, enquanto continue serve para abandonar a interação atual. Veja alguns exemplos:

```
i=3
while i<10:
    print(i,' ')
    if i%7==0:
        break
    i=i+1
print('acabou')
```

```
i=3
while i<10:
    print(i,' ')
    if i%7==0:
        continue
    i=i+1
print('acabou')
```

```
s="Curitiba - Paraná - Brasil"
qtd=0
for c in s:
    if c=="a":
        qtd=qtd+1
print("achei ",qtd," a")
```

Exercícios Nos exercícios a seguir, você deve simular o interpretador Python e descobrir qual valor vai ser impresso ao final. Todos os códigos estão sintaticamente corretos. A sugestão é que depois de ter calculado o valor na ponta do lápis, você confira o resultado com um computador real.

Exemplos

Seja o programa

```
def AA():
    A=5;B=5;C=3
    if C>=18:
        while B<=18:
            C=C-3
            B=B+4
        else:
            C=C+4
            B=B+2
    print(C+C+B)
AA()
```

Cujo resultado é 21 e seja o seguinte código

```
def AA():
    A=4;B=3;C=6
    D=1
    C=C+6+3
    while C>6:
        B=D+7
        while A<=14:
            D=B+4
            A=A+2
            D=B+19
            C=C-1
        print(C+B+B)
AA()
```

cujo resultado é 438.

🔗 Para você fazer
Exercício 1

```
def AA():
    A=3;B=3;C=8
    while B<17:
        A=C-5
        C=A+5
        B=B+1
    print(A+C+B)
AA()
```

Informe no quadro próprio o valor que foi impresso.

Exercício 2

```
def AA():
    A=7;B=4;C=8
    if B!=14:
        while A<=14:
            B=B+2
            A=A+4
        else:
            B=C+5
            A=A+1
    print(B+B+A)
AA()
```

Exercício 3

```
def AA():
    A=6;B=2;C=1
    if C>=17:
        while A<17:
            C=B+2
            A=A+3
        else:
            while A<15:
                B=C+6
                A=A+3
            print(C+B+A)
AA()
```

Exercício 4

```
def AA():
    A=4;B=7;C=2
    D=3
    A=A+2+7
    while A>5:
        B=B+6
        while D<=16:
            B=B+4
            D=D+2
            C=C-14
            A=A-3
        print(D+B+D)
AA()
```

Exercício 5

```
def AA():
    A=5;B=8;C=6
    D=3
    B=B+6+8
    while B>3:
        A=C+3
        if A>5:
            while C<12:
                D=A+3
                C=C+3
            A=D+16
        else:
            A=D+4
            B=B-1
    print(B+A)
AA()
```

Exercício 6

```
def AA():
    A=2;B=3;C=5
    D=7
    B=B+5+3
    while B>2:
        C=D-8
        if B<=5:
            while A<=15:
                A=A+3
                D=C+8
            else:
                D=D+12
        if A>=6:
            D=D+6
        else:
            C=D+8
            B=B-3
    print(D+D)
AA()
```

Exercício 7

```
def AA():
    A=2;B=4;C=8
    D=3
    B=B+8+4
    while B>2:
        A=C-1
        if B>5:
```

```
        while D<19:
            D=D+3
            A=A-2
        else:
            A=C+13
        if B>=8:
            A=A+9
        else:
            A=A+9
            B=B-1
    print(A+A)
AA()
```

Exercício 8

```
def AA():
    A=2;B=5;C=7
    D=3
    B=B+7+5
    while B>=5:
        C=B+7
        if A<8:
            while A<17:
                C=C+5
                A=A+3
                C=D-18
            else:
                D=C-9
                B=B-3
        print(C+C)
AA()
```

Exercício 9

```
def AA():
    A=5;B=7;C=6
    D=3
    B=B+6+7
    while B>=3:
        A=A-4
        while C<12:
            A=A+3
            C=C+6
            D=A-14
            B=B-2
        print(B+A+A)
AA()
```

Exercício 10

```
def AA():
    A=8;B=6;C=1
    if C!=19:
        while A<19:
            C=B+3
            A=A+4
        else:
            while A<12:
                C=C+1
                A=A+3
            print(C+B+A)
AA()
```

Responda aqui:

1	2
3	4
5	6
7	8
9	10



Repetição em Python

Como qualquer linguagem estruturada moderna, Python tem as ferramentas básicas para interar. São elas: while, for, bem como as saídas break e continue. A mais amigável é while. Já a a mais potente é for, pois funciona como um interador, que é uma extensão e tanto ao conceito de indexador. Já os comandos break permitem o abandono de um ciclo, enquanto continue serve para abandonar a interação atual. Veja alguns exemplos:

```
i=3
while i<10:
    print(i,' ')
    if i%7==0:
        break
    i=i+1
print('acabou')
```

```
i=3
while i<10:
    print(i,' ')
    if i%7==0:
        continue
    i=i+1
print('acabou')
```

```
s="Curitiba - Paraná - Brasil"
qtd=0
for c in s:
    if c=="a":
        qtd=qtd+1
print("achei ",qtd," a")
```

Exercícios Nos exercícios a seguir, você deve simular o interpretador Python e descobrir qual valor vai ser impresso ao final. Todos os códigos estão sintaticamente corretos. A sugestão é que depois de ter calculado o valor na ponta do lápis, você confira o resultado com um computador real.

Exemplos

Seja o programa

```
def AA():
    A=5;B=5;C=3
    if C>=18:
        while B<=18:
            C=C-3
            B=B+4
        else:
            C=C+4
            B=B+2
        print(C+C+B)
AA()
```

Cujo resultado é 21 e seja o seguinte código

```
def AA():
    A=4;B=3;C=6
    D=1
    C=C+6+3
    while C>6:
        B=D+7
        while A<=14:
            D=B+4
            A=A+2
            D=B+19
            C=C-1
        print(C+B+B)
AA()
```

cujo resultado é 438.

🔗 Para você fazer

Exercício 1

```
def AA():
    A=5;B=7;C=7
    while C<=14:
        A=A+1
        A=B+4
        C=C+1
        print(A+A+C)
AA()
```

Informe no quadro próprio o valor que foi impresso.

Exercício 2

```
def AA():
    A=2;B=8;C=7
    if C<=13:
        while A<13:
            C=B+2
            A=A+4
        else:
            C=B+8
            A=A+1
        print(C+B+A)
AA()
```

Exercício 3

```
def AA():
    A=8;B=8;C=8
    if C>16:
        while A<16:
            C=B-1
            A=A+1
        else:
            while A<15:
                B=B-2
                A=A+1
            print(C+B+A)
AA()
```

Exercício 4

```
def AA():
    A=5;B=2;C=4
    D=3
    B=B+4+2
    while B>4:
        A=B+7
        while C<18:
            A=D+5
            C=C+5
            D=D+17
            B=B-3
        print(A+D+D)
AA()
```

Exercício 5

```
def AA():
    A=8;B=7;C=6
    D=3
    C=C+6+7
    while C>3:
        B=A+2
        if A<=5:
            while A<16:
                B=B+2
                A=A+4
                D=B+15
            else:
                B=D+7
                C=C-4
        print(C+B)
AA()
```

Exercício 6

```
def AA():
    A=5;B=4;C=6
    D=1
    C=C+6+4
    while C>4:
        B=A+5
        if C>=6:
            while A<17:
                A=A+1
                B=B+3
            else:
                B=D-15
        if B<=4:
            B=B+9
        else:
            B=D-5
            C=C-4
        print(B+D)
AA()
```

Exercício 7

```
def AA():
    A=8;B=1;C=6
    D=5
    A=A+6+1
    while A>3:
        B=B+3
        if D<=8:
```

```
        while D<19:
            D=D+1
            B=C+2
        else:
            B=B-16
        if C>=8:
            B=B+8
        else:
            C=C-10
            A=A-3
        print(D+D)
AA()
```

Exercício 8

```
def AA():
    A=2;B=8;C=5
    D=4
    D=D+5+8
    while D>1:
        B=B+3
        if C>=5:
            while A<=13:
                B=B+5
                A=A+1
                B=C+15
            else:
                B=C+7
                D=D-4
            print(C+A)
AA()
```

Exercício 9

```
def AA():
    A=5;B=3;C=1
    D=8
    D=D+1+3
    while D>7:
        A=A+1
        while B<17:
            C=A+2
            B=B+2
            A=C+15
            D=D-2
        print(D+A+C)
AA()
```

Exercício 10

```
def AA():
    A=4;B=5;C=4
    if C<18:
        while B<=18:
            C=A+5
            B=B+2
        else:
            while B<=14:
                A=C-1
                B=B+3
            print(C+A+B)
AA()
```

Responda aqui:

1	2
3	4
5	6
7	8
9	10



Repetição em Python

Como qualquer linguagem estruturada moderna, Python tem as ferramentas básicas para interar. São elas: while, for, bem como as saídas break e continue. A mais amigável é while. Já a a mais potente é for, pois funciona como um interador, que é uma extensão e tanto ao conceito de indexador. Já os comandos break permitem o abandono de um ciclo, enquanto continue serve para abandonar a interação atual. Veja alguns exemplos:

```
i=3
while i<10:
    print(i,' ')
    if i%7==0:
        break
    i=i+1
print('acabou')
```

```
i=3
while i<10:
    print(i,' ')
    if i%7==0:
        continue
    i=i+1
print('acabou')
```

```
s="Curitiba - Paraná - Brasil"
qtd=0
for c in s:
    if c=="a":
        qtd=qtd+1
print("achei ",qtd," a")
```

Exercícios Nos exercícios a seguir, você deve simular o interpretador Python e descobrir qual valor vai ser impresso ao final. Todos os códigos estão sintaticamente corretos. A sugestão é que depois de ter calculado o valor na ponta do lápis, você confira o resultado com um computador real.

Exemplos

Seja o programa

```
def AA():
    A=5;B=5;C=3
    if C>=18:
        while B<=18:
            C=C-3
            B=B+4
        else:
            C=C+4
            B=B+2
        print(C+C+B)
AA()
```

Cujo resultado é 21 e seja o seguinte código

```
def AA():
    A=4;B=3;C=6
    D=1
    C=C+6+3
    while C>6:
        B=D+7
        while A<=14:
            D=B+4
            A=A+2
            D=B+19
            C=C-1
        print(C+B+B)
AA()
```

cujo resultado é 438.

🔗 Para você fazer

Exercício 1

```
def AA():
    A=7;B=1;C=3
    while C<=17:
        A=B-4
        A=B+4
        C=C+1
        print(A+B+C)
AA()
```

Informe no quadro próprio o valor que foi impresso.

Exercício 2

```
def AA():
    A=5;B=2;C=5
    if B>18:
        while C<=18:
            B=A-6
            C=C+4
        else:
            A=A+8
            C=C+4
        print(B+A+C)
AA()
```

Exercício 3

```
def AA():
    A=6;B=6;C=5
    if B!=15:
        while A<15:
            B=B+2
            A=A+4
        else:
            while A<12:
                C=B-1
                A=A+1
            print(B+B+A)
AA()
```

Exercício 4

```
def AA():
    A=2;B=7;C=1
    D=8
    A=A+1+7
    while A>=2:
        B=D+2
        while D<19:
            C=B-8
            D=D+1
        B=B+16
        A=A-1
        print(D+A+D)
AA()
```

Exercício 5

```
def AA():
    A=1;B=8;C=3
    D=4
    C=C+3+8
    while C>=2:
        A=D+6
        if C<8:
            while B<19:
                A=A+7
                B=B+5
            D=D-18
        else:
            A=A-5
            C=C-4
        print(C+B)
AA()
```

Exercício 6

```
def AA():
    A=5;B=8;C=3
    D=6
    B=B+3+8
    while B>=3:
        C=B+5
        if C<6:
            while A<=12:
                A=A+2
                D=D-7
            else:
                C=C+18
        if A>=4:
            C=C+10
        else:
            D=D+6
            B=B-1
        print(A+D)
AA()
```

Exercício 7

```
def AA():
    A=2;B=8;C=6
    D=7
    B=B+6+8
    while B>7:
        A=A+6
        if C<=4:
```

```
        while C<=13:
            C=C+2
            D=A+8
        else:
            D=A+18
        if A>6:
            A=A+4
        else:
            A=D+4
            B=B-4
        print(A+D)
AA()
```

Exercício 8

```
def AA():
    A=2;B=5;C=8
    D=1
    C=C+8+5
    while C>=2:
        B=C-1
        if C>9:
            while A<13:
                D=D+4
                A=A+6
            B=B-18
        else:
            D=B-9
            C=C-3
        print(C+C)
AA()
```

Exercício 9

```
def AA():
    A=8;B=7;C=2
    D=5
    B=B+2+7
    while B>=2:
        A=C+5
        while C<=13:
            D=D+5
            C=C+6
        A=D+14
        B=B-3
        print(C+B+D)
AA()
```

Exercício 10

```
def AA():
    A=5;B=7;C=3
    if B>=15:
        while A<=15:
            B=C+2
            A=A+2
        else:
            while A<=12:
                B=B+3
                A=A+3
            print(B+C+A)
AA()
```

Responda aqui:

1	2
3	4
5	6
7	8
9	10



Repetição em Python

Como qualquer linguagem estruturada moderna, Python tem as ferramentas básicas para interar. São elas: while, for, bem como as saídas break e continue. A mais amigável é while. Já a a mais potente é for, pois funciona como um interador, que é uma extensão e tanto ao conceito de indexador. Já os comandos break permitem o abandono de um ciclo, enquanto continue serve para abandonar a interação atual. Veja alguns exemplos:

```
i=3
while i<10:
    print(i,' ')
    if i%7==0:
        break
    i=i+1
print('acabou')
```

```
i=3
while i<10:
    print(i,' ')
    if i%7==0:
        continue
    i=i+1
print('acabou')
```

```
s="Curitiba - Paraná - Brasil"
qtd=0
for c in s:
    if c=="a":
        qtd=qtd+1
print("achei ",qtd," a")
```

Exercícios Nos exercícios a seguir, você deve simular o interpretador Python e descobrir qual valor vai ser impresso ao final. Todos os códigos estão sintaticamente corretos. A sugestão é que depois de ter calculado o valor na ponta do lápis, você confira o resultado com um computador real.

Exemplos

Seja o programa

```
def AA():
    A=5;B=5;C=3
    if C>=18:
        while B<=18:
            C=C-3
            B=B+4
        else:
            C=C+4
            B=B+2
    print(C+C+B)
AA()
```

Cujo resultado é 21 e seja o seguinte código

```
def AA():
    A=4;B=3;C=6
    D=1
    C=C+6+3
    while C>6:
        B=D+7
        while A<=14:
            D=B+4
            A=A+2
            D=B+19
            C=C-1
        print(C+B+B)
AA()
```

cujo resultado é 438.

🔗 Para você fazer

Exercício 1

```
def AA():
    A=5;B=8;C=7
    while A<15:
        B=C+6
        C=C+4
        A=A+3
    print(B+C+A)
AA()
```

Informe no quadro próprio o valor que foi impresso.

Exercício 2

```
def AA():
    A=8;B=4;C=1
    if A>15:
        while B<=15:
            A=C+7
            B=B+4
        else:
            A=C+7
            B=B+4
    print(A+C+B)
AA()
```

Exercício 3

```
def AA():
    A=8;B=2;C=3
    if A>=16:
        while C<16:
            A=B+8
            C=C+4
        else:
            while C<14:
                A=B-2
                C=C+1
            print(A+B+C)
AA()
```

Exercício 4

```
def AA():
    A=8;B=1;C=6
    D=4
    A=A+6+1
    while A>4:
        B=B-5
        while D<15:
            B=B+4
            D=D+4
        B=C-17
        A=A-3
    print(A+C+D)
AA()
```

Exercício 5

```
def AA():
    A=6;B=5;C=8
    D=1
    B=B+8+5
    while B>6:
        A=B-2
        if A<=10:
            while D<=13:
                A=A+8
                D=D+5
            A=C+15
        else:
            A=C+4
            B=B-2
    print(C+C)
AA()
```

Exercício 6

```
def AA():
    A=8;B=5;C=7
    D=3
    A=A+7+5
    while A>1:
        B=B+1
        if C<=7:
            while D<=16:
                D=D+5
                B=C-3
            else:
                B=B+13
        if C<5:
            C=C+9
        else:
            C=B+7
            A=A-1
    print(B+D)
AA()
```

Exercício 7

```
def AA():
    A=8;B=4;C=6
    D=5
    A=A+6+4
    while A>=5:
        C=C+6
        if A<7:
```

```
        while B<=15:
            B=B+2
            D=D+2
        else:
            D=D+18
        if D<=10:
            D=D+6
        else:
            D=C+5
            A=A-2
    print(C+B)
AA()
```

Exercício 8

```
def AA():
    A=6;B=8;C=2
    D=5
    D=D+2+8
    while D>2:
        B=D+7
        if C<9:
            while A<17:
                C=B+5
                A=A+1
                B=B+13
            else:
                B=C+4
                D=D-4
        print(B+D)
AA()
```

Exercício 9

```
def AA():
    A=2;B=7;C=8
    D=3
    B=B+8+7
    while B>=6:
        A=D+1
        while D<=19:
            A=A-7
            D=D+5
        A=A-15
        B=B-2
    print(B+C+C)
AA()
```

Exercício 10

```
def AA():
    A=7;B=5;C=5
    if B==19:
        while A<=19:
            B=C+1
            A=A+1
        else:
            while A<=12:
                B=B+4
                A=A+3
            print(B+C+A)
AA()
```

Responda aqui:

1	2
3	4
5	6
7	8
9	10



Repetição em Python

Como qualquer linguagem estruturada moderna, Python tem as ferramentas básicas para interar. São elas: while, for, bem como as saídas break e continue. A mais amigável é while. Já a a mais potente é for, pois funciona como um interador, que é uma extensão e tanto ao conceito de indexador. Já os comandos break permitem o abandono de um ciclo, enquanto continue serve para abandonar a interação atual. Veja alguns exemplos:

```
i=3
while i<10:
    print(i,' ')
    if i%7==0:
        break
    i=i+1
print('acabou')
```

```
i=3
while i<10:
    print(i,' ')
    if i%7==0:
        continue
    i=i+1
print('acabou')
```

```
s="Curitiba - Paraná - Brasil"
qtd=0
for c in s:
    if c=="a":
        qtd=qtd+1
print("achei ",qtd," a")
```

Exercícios Nos exercícios a seguir, você deve simular o interpretador Python e descobrir qual valor vai ser impresso ao final. Todos os códigos estão sintaticamente corretos. A sugestão é que depois de ter calculado o valor na ponta do lápis, você confira o resultado com um computador real.

Exemplos

Seja o programa

```
def AA():
    A=5;B=5;C=3
    if C>=18:
        while B<=18:
            C=C-3
            B=B+4
        else:
            C=C+4
            B=B+2
        print(C+C+B)
AA()
```

Cujo resultado é 21 e seja o seguinte código

```
def AA():
    A=4;B=3;C=6
    D=1
    C=C+6+3
    while C>6:
        B=D+7
        while A<=14:
            D=B+4
            A=A+2
            D=B+19
            C=C-1
        print(C+B+B)
AA()
```

cujo resultado é 438.

🔗 Para você fazer

Exercício 1

```
def AA():
    A=8;B=7;C=7
    while A<=20:
        B=B+2
        B=B-1
        A=A+4
        print(B+B+A)
AA()
```

Informe no quadro próprio o valor que foi impresso.

Exercício 2

```
def AA():
    A=2;B=3;C=2
    if C==13:
        while A<13:
            C=C+4
            A=A+4
        else:
            B=B-1
            A=A+3
        print(C+C+A)
AA()
```

Exercício 3

```
def AA():
    A=1;B=4;C=7
    if C==15:
        while B<=15:
            C=A+6
            B=B+2
        else:
            while B<=15:
                C=A+3
                B=B+2
            print(C+A+B)
AA()
```

Exercício 4

```
def AA():
    A=7;B=8;C=3
    D=4
    A=A+3+8
    while A>=4:
        B=D+5
        while C<=14:
            D=D+7
            C=C+6
            B=D+17
            A=A-4
        print(B+C+C)
AA()
```

Exercício 5

```
def AA():
    A=3;B=6;C=5
    D=7
    C=C+5+6
    while C>5:
        B=D+5
        if A>6:
            while A<=15:
                D=B+6
                A=A+5
            B=D+16
        else:
            B=D+8
            C=C-3
        print(D+B)
AA()
```

Exercício 6

```
def AA():
    A=7;B=8;C=2
    D=5
    B=B+2+8
    while B>=7:
        A=B+5
        if D<7:
            while D<14:
                D=D+6
                A=A+1
            else:
                C=C+12
        if B>9:
            A=A+8
        else:
            A=A+9
            B=B-4
        print(D+B)
AA()
```

Exercício 7

```
def AA():
    A=3;B=4;C=1
    D=7
    D=D+1+4
    while D>5:
        A=C+1
        if A>9:
```

```
        while C<13:
            C=C+3
            A=B-1
        else:
            A=A+15
        if A<4:
            A=A+10
        else:
            B=A+7
            D=D-3
        print(C+B)
AA()
```

Exercício 8

```
def AA():
    A=6;B=8;C=5
    D=4
    B=B+5+8
    while B>6:
        C=B+1
        if B>=10:
            while A<16:
                C=C+2
                A=A+2
            D=C-19
        else:
            D=D-4
            B=B-1
        print(C+D)
AA()
```

Exercício 9

```
def AA():
    A=7;B=8;C=1
    D=4
    B=B+1+8
    while B>4:
        A=D-1
        while D<=15:
            C=A+3
            D=D+5
            A=C-13
            B=B-2
        print(C+D+B)
AA()
```

Exercício 10

```
def AA():
    A=7;B=6;C=4
    if B!=16:
        while C<=16:
            B=B+4
            C=C+1
        else:
            while C<=16:
                A=A-3
                C=C+2
            print(B+B+C)
AA()
```

Responda aqui:

1	2
3	4
5	6
7	8
9	10



Repetição em Python

Como qualquer linguagem estruturada moderna, Python tem as ferramentas básicas para interar. São elas: while, for, bem como as saídas break e continue. A mais amigável é while. Já a a mais potente é for, pois funciona como um interador, que é uma extensão e tanto ao conceito de indexador. Já os comandos break permitem o abandono de um ciclo, enquanto continue serve para abandonar a interação atual. Veja alguns exemplos:

```
i=3
while i<10:
    print(i,' ')
    if i%7==0:
        break
    i=i+1
print('acabou')
```

```
i=3
while i<10:
    print(i,' ')
    if i%7==0:
        continue
    i=i+1
print('acabou')
```

```
s="Curitiba - Paraná - Brasil"
qtd=0
for c in s:
    if c=="a":
        qtd=qtd+1
print("achei ",qtd," a")
```

Exercícios Nos exercícios a seguir, você deve simular o interpretador Python e descobrir qual valor vai ser impresso ao final. Todos os códigos estão sintaticamente corretos. A sugestão é que depois de ter calculado o valor na ponta do lápis, você confira o resultado com um computador real.

Exemplos

Seja o programa

```
def AA():
    A=5;B=5;C=3
    if C>=18:
        while B<=18:
            C=C-3
            B=B+4
        else:
            C=C+4
            B=B+2
        print(C+C+B)
AA()
```

Cujo resultado é 21 e seja o seguinte código

```
def AA():
    A=4;B=3;C=6
    D=1
    C=C+6+3
    while C>6:
        B=D+7
        while A<=14:
            D=B+4
            A=A+2
            D=B+19
            C=C-1
        print(C+B+B)
AA()
```

cujo resultado é 438.

🔗 Para você fazer

Exercício 1

```
def AA():
    A=4;B=2;C=7
    while A<=15:
        C=C-7
        C=C-1
        A=A+4
        print(C+C+A)
AA()
```

Informe no quadro próprio o valor que foi impresso.

Exercício 2

```
def AA():
    A=1;B=7;C=2
    if A!=20:
        while C<20:
            A=A+1
            C=C+4
        else:
            A=B-1
            C=C+3
        print(A+A+C)
AA()
```

Exercício 3

```
def AA():
    A=7;B=7;C=3
    if C<=20:
        while A<=20:
            C=C+4
            A=A+1
        else:
            while A<=16:
                B=B-4
                A=A+1
            print(C+C+A)
AA()
```

Exercício 4

```
def AA():
    A=1;B=4;C=8
    D=6
    D=D+8+4
    while D>=5:
        A=A+1
        while C<=15:
            B=A-5
            C=C+4
            A=A+15
            D=D-4
        print(D+A+A)
AA()
```

Exercício 5

```
def AA():
    A=7;B=8;C=3
    D=2
    C=C+3+8
    while C>=4:
        A=D-6
        if B<10:
            while D<=13:
                B=A-3
                D=D+1
            A=A+16
        else:
            B=B+8
            C=C-3
        print(A+A)
AA()
```

Exercício 6

```
def AA():
    A=5;B=7;C=6
    D=2
    D=D+6+7
    while D>=4:
        B=B+3
        if C<8:
            while A<18:
                A=A+3
                C=B+5
            else:
                C=C+16
        if D<=8:
            B=B+6
        else:
            B=B+8
            D=D-2
        print(C+D)
AA()
```

Exercício 7

```
def AA():
    A=1;B=6;C=3
    D=8
    D=D+3+6
    while D>=5:
        A=D-1
        if A<4:
```

```
        while C<14:
            C=C+1
            A=B-8
        else:
            B=B-16
        if C<4:
            A=A+10
        else:
            A=B-6
            D=D-1
        print(B+C)
AA()
```

Exercício 8

```
def AA():
    A=8;B=7;C=4
    D=2
    C=C+4+7
    while C>=3:
        A=D-2
        if C<=7:
            while D<=13:
                A=A+7
                D=D+1
            A=A+16
        else:
            A=B+9
            C=C-2
        print(C+C)
AA()
```

Exercício 9

```
def AA():
    A=8;B=5;C=7
    D=6
    B=B+7+5
    while B>=4:
        A=C+4
        while C<13:
            A=D+3
            C=C+5
            D=D+15
            B=B-3
        print(B+B+B)
AA()
```

Exercício 10

```
def AA():
    A=8;B=4;C=8
    if C<18:
        while B<=18:
            C=C+5
            B=B+3
        else:
            while B<=18:
                C=C+2
                B=B+2
            print(C+C+B)
AA()
```

Responda aqui:

1	2
3	4
5	6
7	8
9	10



Repetição em Python

Como qualquer linguagem estruturada moderna, Python tem as ferramentas básicas para interar. São elas: while, for, bem como as saídas break e continue. A mais amigável é while. Já a a mais potente é for, pois funciona como um interador, que é uma extensão e tanto ao conceito de indexador. Já os comandos break permitem o abandono de um ciclo, enquanto continue serve para abandonar a interação atual. Veja alguns exemplos:

```
i=3
while i<10:
    print(i,' ')
    if i%7==0:
        break
    i=i+1
print('acabou')
```

```
i=3
while i<10:
    print(i,' ')
    if i%7==0:
        continue
    i=i+1
print('acabou')
```

```
s="Curitiba - Paraná - Brasil"
qtd=0
for c in s:
    if c=="a":
        qtd=qtd+1
print("achei ",qtd," a")
```

Exercícios Nos exercícios a seguir, você deve simular o interpretador Python e descobrir qual valor vai ser impresso ao final. Todos os códigos estão sintaticamente corretos. A sugestão é que depois de ter calculado o valor na ponta do lápis, você confira o resultado com um computador real.

Exemplos

Seja o programa

```
def AA():
    A=5;B=5;C=3
    if C>=18:
        while B<=18:
            C=C-3
            B=B+4
    else:
        C=C+4
        B=B+2
    print(C+C+B)
AA()
```

Cujo resultado é 21 e seja o seguinte código

```
def AA():
    A=4;B=3;C=6
    D=1
    C=C+6+3
    while C>6:
        B=D+7
        while A<=14:
            D=B+4
            A=A+2
            D=B+19
            C=C-1
        print(C+B+B)
AA()
```

cujo resultado é 438.

🔗 Para você fazer

Exercício 1

```
def AA():
    A=6;B=7;C=1
    while B<19:
        A=A+5
        C=A+4
        B=B+1
    print(A+A+B)
AA()
```

Informe no quadro próprio o valor que foi impresso.

Exercício 2

```
def AA():
    A=5;B=4;C=4
    if A>=13:
        while C<=13:
            A=A-2
            C=C+4
        else:
            A=A-6
            C=C+4
    print(A+A+C)
AA()
```

Exercício 3

```
def AA():
    A=7;B=3;C=1
    if B<14:
        while A<=14:
            B=B-5
            A=A+4
        else:
            while A<=14:
                B=B-3
                A=A+3
            print(B+B+A)
AA()
```

Exercício 4

```
def AA():
    A=5;B=3;C=2
    D=4
    C=C+2+3
    while C>=2:
        A=B-1
        while B<=13:
            D=D+7
            B=B+6
        A=A+19
        C=C-4
    print(C+D+D)
AA()
```

Exercício 5

```
def AA():
    A=2;B=7;C=4
    D=3
    D=D+4+7
    while D>5:
        A=C+6
        if D>=6:
            while C<=15:
                B=B+7
                C=C+1
            B=A-15
        else:
            A=A-5
            D=D-2
    print(C+B)
AA()
```

Exercício 6

```
def AA():
    A=7;B=1;C=3
    D=2
    A=A+3+1
    while A>=5:
        B=D-4
        if B<4:
            while D<16:
                D=D+5
                B=C+5
            else:
                C=C-18
        if B<=7:
            B=B+10
        else:
            C=C-6
            A=A-4
    print(C+C)
AA()
```

Exercício 7

```
def AA():
    A=4;B=6;C=2
    D=7
    D=D+2+6
    while D>=5:
        B=B+2
        if B<=7:
```

```
        while A<=18:
            A=A+4
            B=C+4
        else:
            C=B+13
        if B>=8:
            B=B+5
        else:
            B=B+8
            D=D-4
    print(B+D)
AA()
```

Exercício 8

```
def AA():
    A=1;B=8;C=3
    D=6
    D=D+3+8
    while D>7:
        A=B-4
        if B<=8:
            while C<=16:
                A=B+2
                C=C+1
            A=B+14
        else:
            B=B+4
            D=D-1
    print(C+D)
AA()
```

Exercício 9

```
def AA():
    A=6;B=2;C=5
    D=1
    A=A+5+2
    while A>=2:
        B=A+6
        while D<18:
            C=C+7
            D=D+5
        C=B+16
        A=A-2
    print(D+A+D)
AA()
```

Exercício 10

```
def AA():
    A=7;B=5;C=1
    if A<12:
        while B<12:
            A=A-5
            B=B+2
    else:
        while B<12:
            C=A-1
            B=B+2
    print(A+A+B)
AA()
```

Responda aqui:

1	2
3	4
5	6
7	8
9	10



Repetição em Python

Como qualquer linguagem estruturada moderna, Python tem as ferramentas básicas para interar. São elas: while, for, bem como as saídas break e continue. A mais amigável é while. Já a a mais potente é for, pois funciona como um interador, que é uma extensão e tanto ao conceito de indexador. Já os comandos break permitem o abandono de um ciclo, enquanto continue serve para abandonar a interação atual. Veja alguns exemplos:

```
i=3
while i<10:
    print(i,' ')
    if i%7==0:
        break
    i=i+1
print('acabou')
```

```
i=3
while i<10:
    print(i,' ')
    if i%7==0:
        continue
    i=i+1
print('acabou')
```

```
s="Curitiba - Paraná - Brasil"
qtd=0
for c in s:
    if c=="a":
        qtd=qtd+1
print("achei ",qtd," a")
```

Exercícios Nos exercícios a seguir, você deve simular o interpretador Python e descobrir qual valor vai ser impresso ao final. Todos os códigos estão sintaticamente corretos. A sugestão é que depois de ter calculado o valor na ponta do lápis, você confira o resultado com um computador real.

Exemplos

Seja o programa

```
def AA():
    A=5;B=5;C=3
    if C>=18:
        while B<=18:
            C=C-3
            B=B+4
        else:
            C=C+4
            B=B+2
            print(C+C+B)
AA()
```

Cujo resultado é 21 e seja o seguinte código

```
def AA():
    A=4;B=3;C=6
    D=1
    C=C+6+3
    while C>6:
        B=D+7
        while A<=14:
            D=B+4
            A=A+2
            D=B+19
            C=C-1
        print(C+B+B)
AA()
```

cujo resultado é 438.

🔗 Para você fazer

Exercício 1

```
def AA():
    A=6;B=3;C=6
    while A<16:
        B=C+3
        C=C+4
        A=A+3
        print(B+C+A)
AA()
```

Informe no quadro próprio o valor que foi impresso.

Exercício 2

```
def AA():
    A=5;B=2;C=1
    if C<14:
        while B<14:
            C=A+7
            B=B+4
        else:
            C=A+2
            B=B+1
            print(C+A+B)
AA()
```

Exercício 3

```
def AA():
    A=7;B=6;C=5
    if A==18:
        while C<=18:
            A=B+6
            C=C+1
        else:
            while C<=13:
                B=B-6
                C=C+3
            print(A+B+C)
AA()
```

Exercício 4

```
def AA():
    A=6;B=4;C=7
    D=3
    C=C+7+4
    while C>=4:
        A=C-2
        while D<=18:
            A=A+7
            D=D+4
            B=A-17
            C=C-4
        print(A+D+B)
AA()
```

Exercício 5

```
def AA():
    A=2;B=7;C=3
    D=8
    D=D+3+7
    while D>=3:
        A=A-6
        if D<8:
            while B<12:
                A=A+6
                B=B+2
                C=A+14
            else:
                A=A+5
                D=D-4
            print(B+D)
AA()
```

Exercício 6

```
def AA():
    A=7;B=3;C=8
    D=6
    C=C+8+3
    while C>=4:
        A=A+6
        if D>=9:
            while D<17:
                D=D+5
                A=B+1
            else:
                A=B+15
        if A<5:
            A=A+5
        else:
            A=A+8
            C=C-2
        print(D+C)
AA()
```

Exercício 7

```
def AA():
    A=6;B=1;C=2
    D=5
    B=B+2+1
    while B>4:
        A=A+6
        if B<9:
```

```
        while C<=15:
            C=C+3
            A=D+8
        else:
            A=D-17
        if A<7:
            D=D+9
        else:
            D=A-7
            B=B-1
        print(A+A)
AA()
```

Exercício 8

```
def AA():
    A=4;B=7;C=3
    D=8
    A=A+3+7
    while A>4:
        B=C+3
        if B>=9:
            while C<16:
                B=D-1
                C=C+1
                B=B+12
            else:
                B=D+5
                A=A-4
            print(D+A)
AA()
```

Exercício 9

```
def AA():
    A=8;B=7;C=2
    D=3
    B=B+2+7
    while B>1:
        A=D+7
        while D<=14:
            C=C+3
            D=D+1
            A=A-15
            B=B-1
        print(D+C+C)
AA()
```

Exercício 10

```
def AA():
    A=8;B=8;C=1
    if A<20:
        while B<20:
            A=A+5
            B=B+4
        else:
            while B<14:
                C=A+7
                B=B+1
            print(A+A+B)
AA()
```

Responda aqui:

1	2
3	4
5	6
7	8
9	10



Repetição em Python

Como qualquer linguagem estruturada moderna, Python tem as ferramentas básicas para interar. São elas: while, for, bem como as saídas break e continue. A mais amigável é while. Já a a mais potente é for, pois funciona como um interador, que é uma extensão e tanto ao conceito de indexador. Já os comandos break permitem o abandono de um ciclo, enquanto continue serve para abandonar a interação atual. Veja alguns exemplos:

```
i=3
while i<10:
    print(i,' ')
    if i%7==0:
        break
    i=i+1
print('acabou')
```

```
i=3
while i<10:
    print(i,' ')
    if i%7==0:
        continue
    i=i+1
print('acabou')
```

```
s="Curitiba - Paraná - Brasil"
qtd=0
for c in s:
    if c=="a":
        qtd=qtd+1
print("achei ",qtd," a")
```

Exercícios Nos exercícios a seguir, você deve simular o interpretador Python e descobrir qual valor vai ser impresso ao final. Todos os códigos estão sintaticamente corretos. A sugestão é que depois de ter calculado o valor na ponta do lápis, você confira o resultado com um computador real.

Exemplos

Seja o programa

```
def AA():
    A=5;B=5;C=3
    if C>=18:
        while B<=18:
            C=C-3
            B=B+4
        else:
            C=C+4
            B=B+2
    print(C+C+B)
AA()
```

Cujo resultado é 21 e seja o seguinte código

```
def AA():
    A=4;B=3;C=6
    D=1
    C=C+6+3
    while C>6:
        B=D+7
        while A<=14:
            D=B+4
            A=A+2
            D=B+19
            C=C-1
        print(C+B+B)
AA()
```

cujo resultado é 438.

🔗 Para você fazer

Exercício 1

```
def AA():
    A=4;B=8;C=5
    while A<19:
        C=C+3
        C=B+8
        A=A+2
        print(C+C+A)
AA()
```

Informe no quadro próprio o valor que foi impresso.

Exercício 2

```
def AA():
    A=6;B=1;C=1
    if C!=16:
        while A<=16:
            C=B+3
            A=A+4
        else:
            B=B-4
            A=A+1
        print(C+B+A)
AA()
```

Exercício 3

```
def AA():
    A=2;B=4;C=5
    if C<=19:
        while A<19:
            C=B+1
            A=A+1
        else:
            while A<13:
                C=B+1
                A=A+1
            print(C+B+A)
AA()
```

Exercício 4

```
def AA():
    A=8;B=6;C=2
    D=5
    A=A+2+6
    while A>=3:
        B=C+2
        while D<12:
            B=C+8
            D=D+3
            C=B+15
            A=A-1
        print(C+B+A)
AA()
```

Exercício 5

```
def AA():
    A=2;B=4;C=3
    D=6
    B=B+3+4
    while B>=5:
        C=D+1
        if A>10:
            while A<15:
                D=D+3
                A=A+1
            D=D-15
        else:
            D=D-4
            B=B-3
        print(B+B)
AA()
```

Exercício 6

```
def AA():
    A=3;B=1;C=5
    D=4
    D=D+5+1
    while D>2:
        B=A+5
        if B<6:
            while A<=19:
                A=A+3
                B=B+8
            else:
                B=C+19
        if A<=7:
            C=C+5
        else:
            C=B+9
            D=D-2
        print(B+A)
AA()
```

Exercício 7

```
def AA():
    A=4;B=8;C=1
    D=6
    B=B+1+8
    while B>2:
        C=B+2
        if A<=6:
```

```
        while A<12:
            A=A+3
            C=D+8
        else:
            D=C-19
        if A<=10:
            C=C+9
        else:
            C=D-4
            B=B-1
        print(D+B)
AA()
```

Exercício 8

```
def AA():
    A=5;B=7;C=4
    D=6
    B=B+4+7
    while B>4:
        C=C+3
        if B>6:
            while A<=15:
                C=D-2
                A=A+2
            C=C+16
        else:
            D=D+9
            B=B-3
        print(B+A)
AA()
```

Exercício 9

```
def AA():
    A=4;B=3;C=7
    D=5
    C=C+7+3
    while C>5:
        B=A+6
        while A<19:
            D=B-8
            A=A+4
            B=D-18
            C=C-4
        print(A+A+B)
AA()
```

Exercício 10

```
def AA():
    A=5;B=3;C=6
    if A!=18:
        while B<18:
            A=C+1
            B=B+2
        else:
            while B<14:
                C=C+1
                B=B+1
            print(A+C+B)
AA()
```

Responda aqui:

1	2
3	4
5	6
7	8
9	10



Repetição em Python

Como qualquer linguagem estruturada moderna, Python tem as ferramentas básicas para interar. São elas: while, for, bem como as saídas break e continue. A mais amigável é while. Já a a mais potente é for, pois funciona como um interador, que é uma extensão e tanto ao conceito de indexador. Já os comandos break permitem o abandono de um ciclo, enquanto continue serve para abandonar a interação atual. Veja alguns exemplos:

```
i=3
while i<10:
    print(i,' ')
    if i%7==0:
        break
    i=i+1
print('acabou')
```

```
i=3
while i<10:
    print(i,' ')
    if i%7==0:
        continue
    i=i+1
print('acabou')
```

```
s="Curitiba - Paraná - Brasil"
qtd=0
for c in s:
    if c=="a":
        qtd=qtd+1
print("achei ",qtd," a")
```

Exercícios Nos exercícios a seguir, você deve simular o interpretador Python e descobrir qual valor vai ser impresso ao final. Todos os códigos estão sintaticamente corretos. A sugestão é que depois de ter calculado o valor na ponta do lápis, você confira o resultado com um computador real.

Exemplos

Seja o programa

```
def AA():
    A=5;B=5;C=3
    if C>=18:
        while B<=18:
            C=C-3
            B=B+4
        else:
            C=C+4
            B=B+2
        print(C+C+B)
AA()
```

Cujo resultado é 21 e seja o seguinte código

```
def AA():
    A=4;B=3;C=6
    D=1
    C=C+6+3
    while C>6:
        B=D+7
        while A<=14:
            D=B+4
            A=A+2
            D=B+19
            C=C-1
        print(C+B+B)
AA()
```

cujo resultado é 438.

🔗 Para você fazer

Exercício 1

```
def AA():
    A=6;B=3;C=1
    while A<=14:
        B=B-8
        C=C-6
        A=A+2
        print(B+B+A)
AA()
```

Informe no quadro próprio o valor que foi impresso.

Exercício 2

```
def AA():
    A=1;B=8;C=6
    if A!=13:
        while C<=13:
            A=A+4
            C=C+4
        else:
            A=B+3
            C=C+4
        print(A+A+C)
AA()
```

Exercício 3

```
def AA():
    A=4;B=1;C=8
    if B!=16:
        while A<16:
            B=C-8
            A=A+2
        else:
            while A<16:
                B=C+2
                A=A+2
            print(B+C+A)
AA()
```

Exercício 4

```
def AA():
    A=7;B=8;C=1
    D=3
    D=D+1+8
    while D>=6:
        A=A-8
        while C<=15:
            B=B-5
            C=C+5
            B=B-12
            D=D-1
        print(D+B+D)
AA()
```

Exercício 5

```
def AA():
    A=2;B=3;C=7
    D=4
    C=C+7+3
    while C>=4:
        A=A-3
        if B<=9:
            while B<14:
                A=A+4
                B=B+2
            A=D+19
        else:
            D=A+4
            C=C-3
        print(A+D)
AA()
```

Exercício 6

```
def AA():
    A=5;B=1;C=8
    D=3
    B=B+8+1
    while B>3:
        C=D-4
        if C>=8:
            while A<=13:
                A=A+5
                D=D+2
            else:
                D=C+15
        if D<=7:
            D=D+8
        else:
            C=D+4
            B=B-3
        print(B+B)
AA()
```

Exercício 7

```
def AA():
    A=3;B=7;C=8
    D=6
    D=D+8+7
    while D>=7:
        A=C+2
        if A<=7:
```

```
        while C<16:
            C=C+6
            B=B-4
        else:
            A=B+19
        if A<=10:
            B=B+5
        else:
            B=B+7
            D=D-4
        print(A+A)
AA()
```

Exercício 8

```
def AA():
    A=8;B=5;C=4
    D=2
    D=D+4+5
    while D>6:
        A=D-2
        if D<=6:
            while C<=15:
                B=A+3
                C=C+3
                A=A+17
            else:
                B=B+4
                D=D-2
        print(B+A)
AA()
```

Exercício 9

```
def AA():
    A=3;B=1;C=8
    D=4
    A=A+8+1
    while A>7:
        B=B+7
        while C<15:
            D=D-4
            C=C+4
            B=D+14
            A=A-2
        print(D+C+A)
AA()
```

Exercício 10

```
def AA():
    A=1;B=5;C=8
    if C<=13:
        while B<=13:
            C=C+8
            B=B+2
        else:
            while B<=17:
                A=C+2
                B=B+2
            print(C+C+B)
AA()
```

Responda aqui:

1	2
3	4
5	6
7	8
9	10



Repetição em Python

Como qualquer linguagem estruturada moderna, Python tem as ferramentas básicas para interar. São elas: while, for, bem como as saídas break e continue. A mais amigável é while. Já a a mais potente é for, pois funciona como um interador, que é uma extensão e tanto ao conceito de indexador. Já os comandos break permitem o abandono de um ciclo, enquanto continue serve para abandonar a interação atual. Veja alguns exemplos:

```
i=3
while i<10:
    print(i,' ')
    if i%7==0:
        break
    i=i+1
print('acabou')
```

```
i=3
while i<10:
    print(i,' ')
    if i%7==0:
        continue
    i=i+1
print('acabou')
```

```
s="Curitiba - Paraná - Brasil"
qtd=0
for c in s:
    if c=="a":
        qtd=qtd+1
print("achei ",qtd," a")
```

Exercícios Nos exercícios a seguir, você deve simular o interpretador Python e descobrir qual valor vai ser impresso ao final. Todos os códigos estão sintaticamente corretos. A sugestão é que depois de ter calculado o valor na ponta do lápis, você confira o resultado com um computador real.

Exemplos

Seja o programa

```
def AA():
    A=5;B=5;C=3
    if C>=18:
        while B<=18:
            C=C-3
            B=B+4
        else:
            C=C+4
            B=B+2
            print(C+C+B)
AA()
```

Cujo resultado é 21 e seja o seguinte código

```
def AA():
    A=4;B=3;C=6
    D=1
    C=C+6+3
    while C>6:
        B=D+7
        while A<=14:
            D=B+4
            A=A+2
            D=B+19
            C=C-1
            print(C+B+B)
AA()
```

cujo resultado é 438.

🔗 Para você fazer

Exercício 1

```
def AA():
    A=8;B=6;C=4
    while A<=20:
        C=C-8
        C=B+3
        A=A+2
        print(C+C+A)
AA()
```

Informe no quadro próprio o valor que foi impresso.

Exercício 2

```
def AA():
    A=6;B=2;C=6
    if A<=16:
        while C<16:
            A=B-1
            C=C+4
        else:
            B=B+7
            C=C+2
            print(A+B+C)
AA()
```

Exercício 3

```
def AA():
    A=7;B=2;C=3
    if A>=14:
        while B<=14:
            A=C+5
            B=B+3
        else:
            while B<=17:
                A=C+3
                B=B+1
                print(A+C+B)
AA()
```

Exercício 4

```
def AA():
    A=7;B=5;C=3
    D=2
    B=B+3+5
    while B>=5:
        A=A-8
        while D<15:
            A=A-1
            D=D+2
            C=A-19
            B=B-2
            print(A+B+A)
AA()
```

Exercício 5

```
def AA():
    A=3;B=2;C=4
    D=6
    B=B+4+2
    while B>=4:
        C=D+5
        if D<=9:
            while A<13:
                C=D-2
                A=A+2
                C=D+17
            else:
                D=C+8
                B=B-1
                print(A+B)
AA()
```

Exercício 6

```
def AA():
    A=1;B=5;C=3
    D=6
    C=C+3+5
    while C>5:
        B=A+3
        if B>5:
            while A<=12:
                A=A+4
                D=D+1
            else:
                B=B+19
            if A<6:
                B=B+8
            else:
                D=D+10
                C=C-4
                print(C+B)
AA()
```

Exercício 7

```
def AA():
    A=6;B=3;C=8
    D=7
    A=A+8+3
    while A>=1:
        B=D-3
        if C>=9:
```

```
        while C<19:
            C=C+6
            D=D+6
        else:
            D=B+19
        if C>7:
            D=D+4
        else:
            D=D+5
            A=A-4
            print(A+A)
AA()
```

Exercício 8

```
def AA():
    A=6;B=2;C=8
    D=7
    A=A+8+2
    while A>=1:
        C=D+7
        if B>=7:
            while B<=19:
                C=D+7
                B=B+6
                C=C+16
            else:
                D=D+6
                A=A-2
                print(B+B)
AA()
```

Exercício 9

```
def AA():
    A=1;B=3;C=8
    D=4
    D=D+8+3
    while D>4:
        B=B+2
        while A<=13:
            B=B+7
            A=A+2
            B=B+16
            D=D-2
            print(B+D+C)
AA()
```

Exercício 10

```
def AA():
    A=4;B=3;C=1
    if B<15:
        while C<=15:
            B=B-2
            C=C+3
        else:
            while C<=15:
                A=B+5
                C=C+1
                print(B+B+C)
AA()
```

Responda aqui:

1	2
3	4
5	6
7	8
9	10



Repetição em Python

Como qualquer linguagem estruturada moderna, Python tem as ferramentas básicas para interar. São elas: while, for, bem como as saídas break e continue. A mais amigável é while. Já a a mais potente é for, pois funciona como um interador, que é uma extensão e tanto ao conceito de indexador. Já os comandos break permitem o abandono de um ciclo, enquanto continue serve para abandonar a interação atual. Veja alguns exemplos:

```
i=3
while i<10:
    print(i,' ')
    if i%7==0:
        break
    i=i+1
print('acabou')
```

```
i=3
while i<10:
    print(i,' ')
    if i%7==0:
        continue
    i=i+1
print('acabou')
```

```
s="Curitiba - Paraná - Brasil"
qtd=0
for c in s:
    if c=="a":
        qtd=qtd+1
print("achei ",qtd," a")
```

Exercícios Nos exercícios a seguir, você deve simular o interpretador Python e descobrir qual valor vai ser impresso ao final. Todos os códigos estão sintaticamente corretos. A sugestão é que depois de ter calculado o valor na ponta do lápis, você confira o resultado com um computador real.

Exemplos

Seja o programa

```
def AA():
    A=5;B=5;C=3
    if C>=18:
        while B<=18:
            C=C-3
            B=B+4
        else:
            C=C+4
            B=B+2
    print(C+C+B)
AA()
```

Cujo resultado é 21 e seja o seguinte código

```
def AA():
    A=4;B=3;C=6
    D=1
    C=C+6+3
    while C>6:
        B=D+7
        while A<=14:
            D=B+4
            A=A+2
            D=B+19
            C=C-1
        print(C+B+B)
AA()
```

cujo resultado é 438.

☞ Para você fazer

Exercício 1

```
def AA():
    A=4;B=3;C=6
    while C<13:
        A=A+5
        B=B-8
        C=C+4
    print(A+A+C)
AA()
```

Informe no quadro próprio o valor que foi impresso.

Exercício 2

```
def AA():
    A=7;B=1;C=8
    if A<19:
        while C<19:
            A=B+4
            C=C+4
        else:
            B=B+1
            C=C+4
    print(A+B+C)
AA()
```

Exercício 3

```
def AA():
    A=5;B=6;C=2
    if A==19:
        while C<=19:
            A=B+6
            C=C+1
        else:
            while C<=12:
                B=B-4
                C=C+1
            print(A+B+C)
AA()
```

Exercício 4

```
def AA():
    A=3;B=2;C=6
    D=8
    C=C+6+2
    while C>=7:
        A=C+1
        while D<19:
            A=B+7
            D=D+1
            B=A+14
            C=C-1
        print(B+B+D)
AA()
```

Exercício 5

```
def AA():
    A=5;B=6;C=4
    D=8
    A=A+4+6
    while A>=7:
        C=D+8
        if D<=7:
            while B<17:
                D=D+3
                B=B+6
            D=C+14
        else:
            C=C+10
            A=A-2
    print(A+B)
AA()
```

Exercício 6

```
def AA():
    A=5;B=4;C=1
    D=6
    A=A+1+4
    while A>5:
        B=C+5
        if D>10:
            while C<15:
                C=C+1
                D=D-4
            else:
                B=D+14
        if D>9:
            B=B+6
        else:
            B=B+9
            A=A-1
    print(B+C)
AA()
```

Exercício 7

```
def AA():
    A=7;B=8;C=3
    D=1
    B=B+3+8
    while B>5:
        C=B-3
        if D<=8:
```

```
        while A<=18:
            A=A+4
            C=D-6
        else:
            D=C-19
        if A<=6:
            D=D+10
        else:
            C=D-10
            B=B-3
    print(A+A)
AA()
```

Exercício 8

```
def AA():
    A=6;B=4;C=8
    D=3
    C=C+8+4
    while C>=5:
        B=C-3
        if C>7:
            while A<=18:
                B=D+1
                A=A+6
                D=D+16
            else:
                D=D+7
                C=C-1
            print(C+A)
AA()
```

Exercício 9

```
def AA():
    A=7;B=5;C=4
    D=2
    D=D+4+5
    while D>7:
        B=C+5
        while A<=15:
            B=B+4
            A=A+6
            B=C-16
            D=D-4
        print(A+D+D)
AA()
```

Exercício 10

```
def AA():
    A=5;B=2;C=1
    if C==15:
        while A<15:
            C=C+6
            A=A+3
        else:
            while A<16:
                C=C-4
                A=A+3
            print(C+C+A)
AA()
```

Responda aqui:

1	2
3	4
5	6
7	8
9	10



Repetição em Python

Como qualquer linguagem estruturada moderna, Python tem as ferramentas básicas para interar. São elas: while, for, bem como as saídas break e continue. A mais amigável é while. Já a a mais potente é for, pois funciona como um interador, que é uma extensão e tanto ao conceito de indexador. Já os comandos break permitem o abandono de um ciclo, enquanto continue serve para abandonar a interação atual. Veja alguns exemplos:

```
i=3
while i<10:
    print(i,' ')
    if i%7==0:
        break
    i=i+1
print('acabou')
```

```
i=3
while i<10:
    print(i,' ')
    if i%7==0:
        continue
    i=i+1
print('acabou')
```

```
s="Curitiba - Paraná - Brasil"
qtd=0
for c in s:
    if c=="a":
        qtd=qtd+1
print("achei ",qtd," a")
```

Exercícios Nos exercícios a seguir, você deve simular o interpretador Python e descobrir qual valor vai ser impresso ao final. Todos os códigos estão sintaticamente corretos. A sugestão é que depois de ter calculado o valor na ponta do lápis, você confira o resultado com um computador real.

Exemplos

Seja o programa

```
def AA():
    A=5;B=5;C=3
    if C>=18:
        while B<=18:
            C=C-3
            B=B+4
        else:
            C=C+4
            B=B+2
            print(C+C+B)
AA()
```

Cujo resultado é 21 e seja o seguinte código

```
def AA():
    A=4;B=3;C=6
    D=1
    C=C+6+3
    while C>6:
        B=D+7
        while A<=14:
            D=B+4
            A=A+2
            D=B+19
            C=C-1
        print(C+B+B)
AA()
```

cujo resultado é 438.

🔗 Para você fazer

Exercício 1

```
def AA():
    A=4;B=7;C=7
    while B<=12:
        C=A+1
        C=A+4
        B=B+2
        print(C+A+B)
AA()
```

Informe no quadro próprio o valor que foi impresso.

Exercício 2

```
def AA():
    A=8;B=6;C=5
    if A<=15:
        while B<15:
            A=A+5
            B=B+4
        else:
            C=C+5
            B=B+2
            print(A+A+B)
AA()
```

Exercício 3

```
def AA():
    A=1;B=8;C=1
    if B>12:
        while C<12:
            B=A+6
            C=C+1
        else:
            while C<18:
                B=B+6
                C=C+2
                print(B+A+C)
AA()
```

Exercício 4

```
def AA():
    A=5;B=3;C=7
    D=6
    A=A+7+3
    while A>6:
        B=D-7
        while D<18:
            B=C+5
            D=D+3
            B=C-15
            A=A-1
            print(B+A+D)
AA()
```

Exercício 5

```
def AA():
    A=5;B=4;C=7
    D=6
    B=B+7+4
    while B>1:
        C=D+2
        if B>=9:
            while A<=13:
                D=D+6
                A=A+1
                C=D-19
            else:
                D=C-5
                B=B-4
                print(C+B)
AA()
```

Exercício 6

```
def AA():
    A=8;B=3;C=6
    D=5
    B=B+6+3
    while B>2:
        A=B+5
        if D<4:
            while D<=16:
                D=D+2
                A=A+2
            else:
                A=C+18
            if A>9:
                C=C+7
            else:
                A=A+6
                B=B-1
            print(D+B)
AA()
```

Exercício 7

```
def AA():
    A=4;B=2;C=5
    D=6
    C=C+5+2
    while C>=5:
        B=D+3
        if C>6:
```

```
        while A<16:
            A=A+4
            D=D-8
        else:
            B=D-16
            if D<=9:
                D=D+5
            else:
                D=D-7
            C=C-1
        print(A+D)
AA()
```

Exercício 8

```
def AA():
    A=5;B=7;C=8
    D=4
    A=A+8+7
    while A>1:
        B=A+1
        if D<=6:
            while C<=16:
                D=D+7
                C=C+5
                B=D+14
            else:
                B=B+4
                A=A-1
            print(D+B)
AA()
```

Exercício 9

```
def AA():
    A=2;B=3;C=4
    D=8
    D=D+4+3
    while D>=5:
        B=C-4
        while A<=17:
            B=C-1
            A=A+5
            B=C-16
            D=D-2
        print(A+A+D)
AA()
```

Exercício 10

```
def AA():
    A=1;B=2;C=8
    if A>=19:
        while B<19:
            A=A+1
            B=B+1
        else:
            while B<15:
                A=C+6
                B=B+1
            print(A+A+B)
AA()
```

Responda aqui:

1	2
3	4
5	6
7	8
9	10



Repetição em Python

Como qualquer linguagem estruturada moderna, Python tem as ferramentas básicas para interar. São elas: while, for, bem como as saídas break e continue. A mais amigável é while. Já a a mais potente é for, pois funciona como um interador, que é uma extensão e tanto ao conceito de indexador. Já os comandos break permitem o abandono de um ciclo, enquanto continue serve para abandonar a interação atual. Veja alguns exemplos:

```
i=3
while i<10:
    print(i,' ')
    if i%7==0:
        break
    i=i+1
print('acabou')
```

```
i=3
while i<10:
    print(i,' ')
    if i%7==0:
        continue
    i=i+1
print('acabou')
```

```
s="Curitiba - Paraná - Brasil"
qtd=0
for c in s:
    if c=="a":
        qtd=qtd+1
print("achei ",qtd," a")
```

Exercícios Nos exercícios a seguir, você deve simular o interpretador Python e descobrir qual valor vai ser impresso ao final. Todos os códigos estão sintaticamente corretos. A sugestão é que depois de ter calculado o valor na ponta do lápis, você confira o resultado com um computador real.

Exemplos

Seja o programa

```
def AA():
    A=5;B=5;C=3
    if C>=18:
        while B<=18:
            C=C-3
            B=B+4
        else:
            C=C+4
            B=B+2
            print(C+C+B)
AA()
```

Cujo resultado é 21 e seja o seguinte código

```
def AA():
    A=4;B=3;C=6
    D=1
    C=C+6+3
    while C>6:
        B=D+7
        while A<=14:
            D=B+4
            A=A+2
            D=B+19
            C=C-1
        print(C+B+B)
AA()
```

cujo resultado é 438.

🔗 Para você fazer

Exercício 1

```
def AA():
    A=1;B=7;C=8
    while A<=18:
        B=C+6
        C=B+7
        A=A+2
        print(B+C+A)
AA()
```

Informe no quadro próprio o valor que foi impresso.

Exercício 2

```
def AA():
    A=5;B=7;C=5
    if B>=18:
        while A<=18:
            B=B+5
            A=A+4
        else:
            B=C-1
            A=A+1
            print(B+B+A)
AA()
```

Exercício 3

```
def AA():
    A=4;B=7;C=2
    if C==12:
        while A<=12:
            C=C-1
            A=A+4
        else:
            while A<=13:
                C=B-7
                A=A+3
                print(C+C+A)
AA()
```

Exercício 4

```
def AA():
    A=3;B=4;C=2
    D=1
    A=A+2+4
    while A>=6:
        B=B+1
        while C<18:
            B=D-6
            C=C+3
            B=B+15
            A=A-3
            print(D+A+D)
AA()
```

Exercício 5

```
def AA():
    A=5;B=6;C=8
    D=7
    A=A+8+6
    while A>=5:
        C=A+3
        if B>=4:
            while B<=13:
                C=C+3
                B=B+3
                C=C+16
            else:
                C=D+4
                A=A-1
            print(B+A)
AA()
```

Exercício 6

```
def AA():
    A=3;B=6;C=1
    D=4
    A=A+1+6
    while A>2:
        B=C+8
        if C<=7:
            while C<19:
                C=C+3
                B=B+4
            else:
                B=D+14
        if C>=6:
            B=B+4
        else:
            D=B+7
            A=A-2
            print(A+D)
AA()
```

Exercício 7

```
def AA():
    A=1;B=8;C=2
    D=3
    D=D+2+8
    while D>3:
        B=B-6
        if D>=4:
```

```
        while A<13:
            A=A+1
            B=C+4
        else:
            C=C+13
        if B<=6:
            C=C+9
        else:
            B=C+4
            D=D-4
        print(B+D)
AA()
```

Exercício 8

```
def AA():
    A=4;B=6;C=8
    D=5
    B=B+8+6
    while B>=7:
        A=C-7
        if C>5:
            while D<=18:
                C=C+7
                D=D+5
                A=A+14
            else:
                C=C+8
                B=B-4
                print(D+B)
AA()
```

Exercício 9

```
def AA():
    A=7;B=3;C=6
    D=1
    A=A+6+3
    while A>5:
        C=D+5
        while B<=16:
            C=C-2
            B=B+3
            C=C-13
            A=A-1
            print(A+A+C)
AA()
```

Exercício 10

```
def AA():
    A=1;B=2;C=1
    if C>=13:
        while A<13:
            C=B+4
            A=A+3
        else:
            while A<17:
                B=C+5
                A=A+1
                print(C+B+A)
AA()
```

Responda aqui:

1	2
3	4
5	6
7	8
9	10



Repetição em Python

Como qualquer linguagem estruturada moderna, Python tem as ferramentas básicas para interar. São elas: while, for, bem como as saídas break e continue. A mais amigável é while. Já a a mais potente é for, pois funciona como um interador, que é uma extensão e tanto ao conceito de indexador. Já os comandos break permitem o abandono de um ciclo, enquanto continue serve para abandonar a interação atual. Veja alguns exemplos:

```
i=3
while i<10:
    print(i,' ')
    if i%7==0:
        break
    i=i+1
print('acabou')
```

```
i=3
while i<10:
    print(i,' ')
    if i%7==0:
        continue
    i=i+1
print('acabou')
```

```
s="Curitiba - Paraná - Brasil"
qtd=0
for c in s:
    if c=="a":
        qtd=qtd+1
print("achei ",qtd," a")
```

Exercícios Nos exercícios a seguir, você deve simular o interpretador Python e descobrir qual valor vai ser impresso ao final. Todos os códigos estão sintaticamente corretos. A sugestão é que depois de ter calculado o valor na ponta do lápis, você confira o resultado com um computador real.

Exemplos

Seja o programa

```
def AA():
    A=5;B=5;C=3
    if C>=18:
        while B<=18:
            C=C-3
            B=B+4
    else:
        C=C+4
        B=B+2
    print(C+C+B)
AA()
```

Cujo resultado é 21 e seja o seguinte código

```
def AA():
    A=4;B=3;C=6
    D=1
    C=C+6+3
    while C>6:
        B=D+7
        while A<=14:
            D=B+4
            A=A+2
            D=B+19
            C=C-1
        print(C+B+B)
AA()
```

cujo resultado é 438.

🔗 Para você fazer

Exercício 1

```
def AA():
    A=6;B=4;C=2
    while C<17:
        A=B-7
        B=A+1
        C=C+1
        print(A+B+C)
AA()
```

Informe no quadro próprio o valor que foi impresso.

Exercício 2

```
def AA():
    A=3;B=5;C=5
    if A!=19:
        while B<19:
            A=C+2
            B=B+4
        else:
            A=A+3
            B=B+4
        print(A+C+B)
AA()
```

Exercício 3

```
def AA():
    A=1;B=7;C=3
    if A<15:
        while C<=15:
            A=A+8
            C=C+1
        else:
            while C<=14:
                A=A+3
                C=C+1
            print(A+A+C)
AA()
```

Exercício 4

```
def AA():
    A=7;B=6;C=3
    D=2
    B=B+3+6
    while B>=7:
        A=D+5
        while C<12:
            D=A-5
            C=C+4
        A=A-15
        B=B-3
        print(C+B+C)
AA()
```

Exercício 5

```
def AA():
    A=6;B=1;C=8
    D=2
    A=A+8+1
    while A>5:
        B=B-5
        if B<9:
            while D<13:
                C=B+5
                D=D+3
            B=C-16
        else:
            C=B-8
            A=A-1
        print(B+C)
AA()
```

Exercício 6

```
def AA():
    A=7;B=6;C=1
    D=8
    A=A+1+6
    while A>7:
        B=A+3
        if D<=4:
            while D<18:
                D=D+3
                B=C+1
        else:
            C=B+18
        if B<7:
            C=C+9
        else:
            B=C+5
            A=A-1
        print(C+C)
AA()
```

Exercício 7

```
def AA():
    A=1;B=5;C=2
    D=4
    B=B+2+5
    while B>3:
        A=C-1
        if A>=9:
```

```
        while D<15:
            D=D+6
            C=A+4
        else:
            A=C-19
        if B>=10:
            C=C+10
        else:
            C=C-8
            B=B-3
        print(A+C)
AA()
```

Exercício 8

```
def AA():
    A=2;B=8;C=1
    D=7
    C=C+1+8
    while C>5:
        A=A-2
        if C<=8:
            while D<14:
                B=A+3
                D=D+2
            B=A+13
        else:
            B=A+10
            C=C-3
        print(D+A)
AA()
```

Exercício 9

```
def AA():
    A=4;B=2;C=1
    D=7
    C=C+1+2
    while C>2:
        A=B+2
        while D<16:
            B=B+8
            D=D+2
        B=A+17
        C=C-3
        print(B+D+C)
AA()
```

Exercício 10

```
def AA():
    A=6;B=5;C=4
    if B!=17:
        while C<=17:
            B=A+5
            C=C+4
        else:
            while C<=14:
                B=B-6
                C=C+1
            print(B+A+C)
AA()
```

Responda aqui:

1	2
3	4
5	6
7	8
9	10



Repetição em Python

Como qualquer linguagem estruturada moderna, Python tem as ferramentas básicas para interar. São elas: while, for, bem como as saídas break e continue. A mais amigável é while. Já a a mais potente é for, pois funciona como um interador, que é uma extensão e tanto ao conceito de indexador. Já os comandos break permitem o abandono de um ciclo, enquanto continue serve para abandonar a interação atual. Veja alguns exemplos:

```
i=3
while i<10:
    print(i,' ')
    if i%7==0:
        break
    i=i+1
print('acabou')
```

```
i=3
while i<10:
    print(i,' ')
    if i%7==0:
        continue
    i=i+1
print('acabou')
```

```
s="Curitiba - Paraná - Brasil"
qtd=0
for c in s:
    if c=="a":
        qtd=qtd+1
print("achei ",qtd," a")
```

Exercícios Nos exercícios a seguir, você deve simular o interpretador Python e descobrir qual valor vai ser impresso ao final. Todos os códigos estão sintaticamente corretos. A sugestão é que depois de ter calculado o valor na ponta do lápis, você confira o resultado com um computador real.

Exemplos

Seja o programa

```
def AA():
    A=5;B=5;C=3
    if C>=18:
        while B<=18:
            C=C-3
            B=B+4
        else:
            C=C+4
            B=B+2
        print(C+C+B)
AA()
```

Cujo resultado é 21 e seja o seguinte código

```
def AA():
    A=4;B=3;C=6
    D=1
    C=C+6+3
    while C>6:
        B=D+7
        while A<=14:
            D=B+4
            A=A+2
            D=B+19
            C=C-1
        print(C+B+B)
AA()
```

cujo resultado é 438.

☞ Para você fazer

Exercício 1

```
def AA():
    A=3;B=3;C=7
    while B<13:
        C=A-1
        A=A-1
        B=B+3
        print(C+A+B)
AA()
```

Informe no quadro próprio o valor que foi impresso.

Exercício 2

```
def AA():
    A=2;B=2;C=7
    if B<18:
        while C<=18:
            B=A+3
            C=C+4
        else:
            A=B+5
            C=C+2
        print(B+A+C)
AA()
```

Exercício 3

```
def AA():
    A=7;B=5;C=4
    if A<=13:
        while C<=13:
            A=A-6
            C=C+2
        else:
            while C<=16:
                A=B+7
                C=C+3
            print(A+A+C)
AA()
```

Exercício 4

```
def AA():
    A=1;B=4;C=5
    D=7
    B=B+5+4
    while B>=4:
        A=B+2
        while C<19:
            D=D-1
            C=C+4
            D=D+13
            B=B-4
        print(A+B+B)
AA()
```

Exercício 5

```
def AA():
    A=5;B=8;C=3
    D=1
    B=B+3+8
    while B>=5:
        A=C+6
        if A>=5:
            while C<19:
                D=A+7
                C=C+6
                D=D+17
            else:
                D=A+5
                B=B-1
            print(D+D)
AA()
```

Exercício 6

```
def AA():
    A=6;B=4;C=8
    D=3
    D=D+8+4
    while D>2:
        A=A+4
        if C<=5:
            while B<=12:
                B=B+4
                C=C-3
            else:
                C=A+16
        if A>8:
            A=A+9
        else:
            A=A+10
            D=D-4
        print(C+C)
AA()
```

Exercício 7

```
def AA():
    A=1;B=3;C=5
    D=7
    D=D+5+3
    while D>=1:
        A=A-3
        if C>10:
```

```
        while C<17:
            C=C+4
            B=B+5
        else:
            A=B+12
        if A>=7:
            A=A+10
        else:
            B=B+9
            D=D-1
        print(B+C)
AA()
```

Exercício 8

```
def AA():
    A=1;B=2;C=4
    D=7
    C=C+4+2
    while C>=5:
        A=A+7
        if C<=5:
            while B<13:
                A=D+3
                B=B+3
                A=D-17
            else:
                D=D-4
                C=C-4
            print(C+A)
AA()
```

Exercício 9

```
def AA():
    A=4;B=3;C=8
    D=7
    C=C+8+3
    while C>1:
        A=D-2
        while D<=18:
            A=A+2
            D=D+6
            A=B+12
            C=C-3
        print(C+C+A)
AA()
```

Exercício 10

```
def AA():
    A=5;B=1;C=3
    if B>=13:
        while A<13:
            B=C-2
            A=A+4
        else:
            while A<18:
                C=C+3
                A=A+1
            print(B+C+A)
AA()
```

Responda aqui:

1	2
3	4
5	6
7	8
9	10

