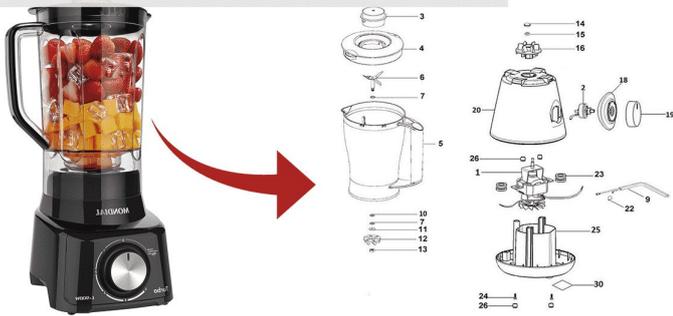


Engenharia Reversa

ENGENHARIA REVERSA



Engenharia Reversa de algoritmos

Se a programação fosse mais engenharia e menos arte, talvez não precisássemos tanto usar os dons de Sherlock Holmes para corrigir programas. Infelizmente (ou será felizmente ??) tal não ocorre, e um programador profissional gasta mais de 50% do seu tempo procurando e corrigindo erros em programas.

É tarefa difícil, sutil, complexa e delicada. s vezes um ponto fora do lugar põe a perder um imenso projeto. É tão importante a tarefa que se diz que bom programador não é aquele de programa bem, mas aquele que depura bem. A propósito, depurar é **identificar, localizar e corrigir erros, de maneira continuada e integrada para produzir software de qualidade.**

O exercício de hoje, visa estimular esta habilidade. Você vai receber alguns códigos e alguns resultados, e deverá inferir quais as inicializações originais das variáveis em cada código.

Por exemplo, no algoritmo

```

1: função EXEMPLO
2: A ← _____
3: B ← 2
4: C ← 3
5: enquanto A < 10
6:   se (B mod 3) = 0
7:     A ← A + 1
8:   senão
9:     A ← A + 2
10:  fim{se}
11:  se (C mod 4) = 0
12:    B ← B + 2
13:  senão
14:    A ← A + 1
15:  fim{se}
16:  imprima A, B, C
17: fim{enquanto}
    
```

Analisando as saídas, que são:

4 2 3
 7 2 3
 10 2 3

Percebe-se que a inicialização original do A foi 1. A resposta deste exemplo portanto, é 1.

Para você fazer

A seguir, diversos algoritmos. Em cada um deles, você deve dizer qual o valor inicial da variável A.

Exercício 1

```

1: função A1
2: A ← _____
3: B ← 6
4: C ← 13
5: enquanto (A + B) > 4
6:   se A > C
7:     A ← A - 1
8:   senão
9:     A ← A + 1
    
```

```

10:  fim{se}
11:  B ← B - 2
12:  imprima A, B, C
13: fim{enquanto}
    
```

Esta execução deu como resultado

9 4 13
 10 2 13
 11 0 13
 12 -2 13
 13 -4 13
 14 -6 13
 13 -8 13
 14 -10 13

Exercício 2

```

1: função A2
2: A ← _____
3: B ← 15
4: C ← 10
5: enquanto (C ≠ 0) ∧ (C ≠ 1)
6:   se (B mod 5) ≠ 0
7:     B ← B - 1
8:   fim{se}
9:   se (A mod 3) = 0
10:    A ← A + 1
11:  fim{se}
12:  se (A > B)
13:    A ← A + 2
14:  fim{se}
15:  C ← C - 1
16:  imprima A, B, C
17: fim{enquanto}
    
```

Esta execução deu como resultado

11 15 9
 11 15 8
 11 15 7
 11 15 6
 11 15 5
 11 15 4
 11 15 3
 11 15 2
 11 15 1

Exercício 3

```

1: função A3
2: A ← _____
3: B ← 1
4: C ← 3
5: repita
6:   A ← A + 1
7:   B ← B + 1
8:   C ← C + 1
9: enquanto C > 10
10:  C ← C - 2
11: fim{enquanto}
12: se (C ≠ 3) ∨ (B ≠ 2)
13:  A ← A + 2
14: fim{se}
15: imprima A, B, C
16: até (A + B + C) > 30
    
```

Esta execução deu como resultado

4 2 4
 7 3 5

10 4 6
 13 5 7
 16 6 8
 19 7 9

Exercício 4

```

1: função A4
2: A ← _____
3: B ← 12
4: C ← 15
5: enquanto (A > 10) ∨ (B > 10) ∨ (C > 10)
6:   se A > B
7:     C ← C - 1
8:   fim{se}
9:   se (B mod 3) = 1
10:    A ← A - 1
11:  fim{se}
12:  se C > B
13:    B ← B - 3
14:  fim{se}
15:  imprima A, B, C
16:  A ← A - 1
17:  B ← B - 1
18:  C ← C - 2
19: fim{enquanto}
    
```

Esta execução deu como resultado

14 9 14
 13 5 11
 11 1 8

Respostas

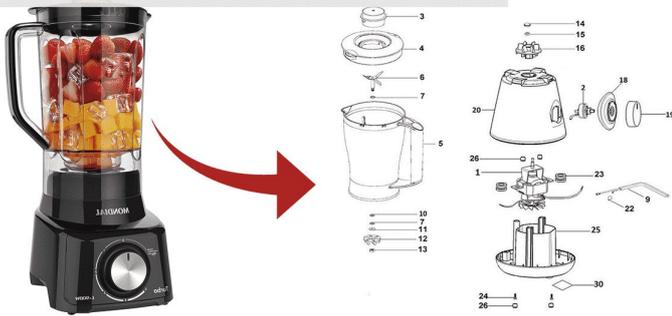
Escreva aqui os valores presumidos para as 4 variáveis "A" de cada um dos 4 algoritmos

1	2	3	4

75789

Engenharia Reversa

ENGENHARIA REVERSA



Engenharia Reversa de algoritmos

Se a programação fosse mais engenharia e menos arte, talvez não precisássemos tanto usar os dons de Sherlock Holmes para corrigir programas. Infelizmente (ou será felizmente ??) tal não ocorre, e um programador profissional gasta mais de 50% do seu tempo procurando e corrigindo erros em programas.

É tarefa difícil, sutil, complexa e delicada. s vezes um ponto fora do lugar põe a perder um imenso projeto. É tão importante a tarefa que se diz que bom programador não é aquele de programa bem, mas aquele que depura bem. A propósito, depurar é **identificar, localizar e corrigir erros, de maneira continuada e integrada para produzir software de qualidade.**

O exercício de hoje, visa estimular esta habilidade. Você vai receber alguns códigos e alguns resultados, e deverá inferir quais as inicializações originais das variáveis em cada código.

Por exemplo, no algoritmo

```

1: função EXEMPLO
2: A ← _____
3: B ← 2
4: C ← 3
5: enquanto A < 10
6:   se (B mod 3) = 0
7:     A ← A + 1
8:   senão
9:     A ← A + 2
10:  fim{se}
11:  se (C mod 4) = 0
12:    B ← B + 2
13:  senão
14:    A ← A + 1
15:  fim{se}
16:  imprima A, B, C
17: fim{enquanto}
    
```

Analisando as saídas, que são:

4 2 3
 7 2 3
 10 2 3

Percebe-se que a inicialização original do A foi 1. A resposta deste exemplo portanto, é 1.

Para você fazer

A seguir, diversos algoritmos. Em cada um deles, você deve dizer qual o valor inicial da variável A.

Exercício 1

```

1: função A1
2: A ← _____
3: B ← 6
4: C ← 13
5: enquanto (A + B) > 4
6:   se A > C
7:     A ← A - 1
8:   senão
9:     A ← A + 1
    
```

```

10:  fim{se}
11:  B ← B - 2
12:  imprima A, B, C
13: fim{enquanto}
    
```

Esta execução deu como resultado

8 4 13
 9 2 13
 10 0 13
 11 -2 13
 12 -4 13
 13 -6 13
 14 -8 13
 13 -10 13

Exercício 2

```

1: função A2
2: A ← _____
3: B ← 9
4: C ← 9
5: enquanto (C ≠ 0) ∧ (C ≠ 1)
6:   se (B mod 5) ≠ 0
7:     B ← B - 1
8:   fim{se}
9:   se (A mod 3) = 0
10:    A ← A + 1
11:  fim{se}
12:  se (A > B)
13:    A ← A + 2
14:  fim{se}
15:  C ← C - 1
16:  imprima A, B, C
17: fim{enquanto}
    
```

Esta execução deu como resultado

16 8 8
 18 7 7
 21 6 6
 24 5 5
 27 5 4
 30 5 3
 33 5 2
 36 5 1

Exercício 3

```

1: função A3
2: A ← _____
3: B ← 2
4: C ← 4
5: repita
6:   A ← A + 1
7:   B ← B + 1
8:   C ← C + 1
9: enquanto C > 10
10:  C ← C - 2
11: fim{enquanto}
12: se (C ≠ 3) ∨ (B ≠ 2)
13:  A ← A + 2
14: fim{se}
15: imprima A, B, C
16: até (A + B + C) > 30
    
```

Esta execução deu como resultado

6 3 5
 9 4 6
 12 5 7

15 6 8
 18 7 9

Exercício 4

```

1: função A4
2: A ← _____
3: B ← 13
4: C ← 12
5: enquanto (A > 10) ∨ (B > 10) ∨ (C > 10)
6:   se A > B
7:     C ← C - 1
8:   fim{se}
9:   se (B mod 3) = 1
10:    A ← A - 1
11:  fim{se}
12:  se C > B
13:    B ← B - 3
14:  fim{se}
15:  imprima A, B, C
16:  A ← A - 1
17:  B ← B - 1
18:  C ← C - 2
19: fim{enquanto}
    
```

Esta execução deu como resultado

10 13 12
 9 12 10
 8 11 8

Respostas

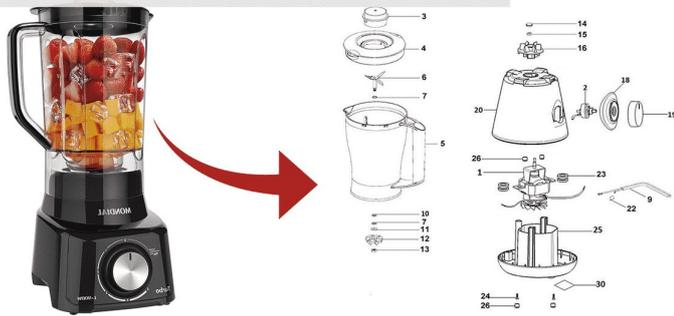
Escreva aqui os valores presumidos para as 4 variáveis "A" de cada um dos 4 algoritmos

1	2	3	4

75796

Engenharia Reversa

ENGENHARIA REVERSA



Engenharia Reversa de algoritmos

Se a programação fosse mais engenharia e menos arte, talvez não precisássemos tanto usar os dons de Sherlock Holmes para corrigir programas. Infelizmente (ou será felizmente ??) tal não ocorre, e um programador profissional gasta mais de 50% do seu tempo procurando e corrigindo erros em programas.

É tarefa difícil, sutil, complexa e delicada. s vezes um ponto fora do lugar põe a perder um imenso projeto. É tão importante a tarefa que se diz que bom programador não é aquele de programa bem, mas aquele que depura bem. A propósito, depurar é **identificar, localizar e corrigir erros, de maneira continuada e integrada para produzir software de qualidade.**

O exercício de hoje, visa estimular esta habilidade. Você vai receber alguns códigos e alguns resultados, e deverá inferir quais as inicializações originais das variáveis em cada código.

Por exemplo, no algoritmo

```

1: função EXEMPLO
2: A ← _____
3: B ← 2
4: C ← 3
5: enquanto A < 10
6:   se (B mod 3) = 0
7:     A ← A + 1
8:   senão
9:     A ← A + 2
10:  fim{se}
11:  se (C mod 4) = 0
12:    B ← B + 2
13:  senão
14:    A ← A + 1
15:  fim{se}
16:  imprima A, B, C
17: fim{enquanto}
    
```

Analisando as saídas, que são:

4 2 3
 7 2 3
 10 2 3

Percebe-se que a inicialização original do A foi 1. A resposta deste exemplo portanto, é 1.

Para você fazer

A seguir, diversos algoritmos. Em cada um deles, você deve dizer qual o valor inicial da variável A.

Exercício 1

```

1: função A1
2: A ← _____
3: B ← 10
4: C ← 8
5: enquanto (A + B) > 4
6:   se A > C
7:     A ← A - 1
8:   senão
9:     A ← A + 1
    
```

```

10:  fim{se}
11:  B ← B - 2
12:  imprima A, B, C
13: fim{enquanto}
    
```

Esta execução deu como resultado

7 8 8
 8 6 8
 9 4 8
 8 2 8
 9 0 8
 8 -2 8
 9 -4 8
 8 -6 8

Exercício 2

```

1: função A2
2: A ← _____
3: B ← 15
4: C ← 10
5: enquanto (C ≠ 0) ∧ (C ≠ 1)
6:   se (B mod 5) ≠ 0
7:     B ← B - 1
8:   fim{se}
9:   se (A mod 3) = 0
10:    A ← A + 1
11:  fim{se}
12:  se (A > B)
13:    A ← A + 2
14:  fim{se}
15:  C ← C - 1
16:  imprima A, B, C
17: fim{enquanto}
    
```

Esta execução deu como resultado

14 15 9
 14 15 8
 14 15 7
 14 15 6
 14 15 5
 14 15 4
 14 15 3
 14 15 2
 14 15 1

Exercício 3

```

1: função A3
2: A ← _____
3: B ← 5
4: C ← 4
5: repita
6:   A ← A + 1
7:   B ← B + 1
8:   C ← C + 1
9: enquanto C > 10
10:  C ← C - 2
11: fim{enquanto}
12: se (C ≠ 3) ∨ (B ≠ 2)
13:  A ← A + 2
14: fim{se}
15: imprima A, B, C
16: até (A + B + C) > 30
    
```

Esta execução deu como resultado

6 6 5
 9 7 6

12 8 7
 15 9 8

Exercício 4

```

1: função A4
2: A ← _____
3: B ← 14
4: C ← 14
5: enquanto (A > 10) ∨ (B > 10) ∨ (C > 10)
6:   se A > B
7:     C ← C - 1
8:   fim{se}
9:   se (B mod 3) = 1
10:    A ← A - 1
11:  fim{se}
12:  se C > B
13:    B ← B - 3
14:  fim{se}
15:  imprima A, B, C
16:  A ← A - 1
17:  B ← B - 1
18:  C ← C - 2
19: fim{enquanto}
    
```

Esta execução deu como resultado

11 14 14
 9 13 12
 8 12 10
 7 11 8

Respostas

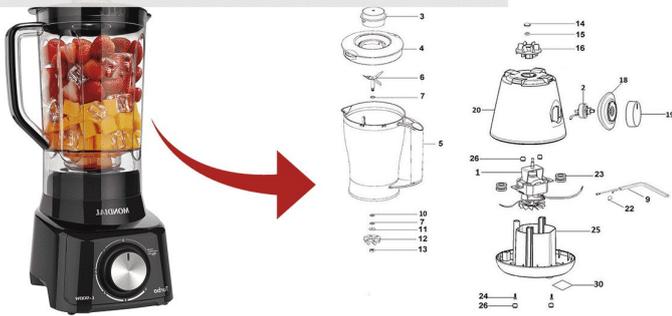
Escreva aqui os valores presumidos para as 4 variáveis "A" de cada um dos 4 algoritmos

1	2	3	4

75808

Engenharia Reversa

ENGENHARIA REVERSA



Engenharia Reversa de algoritmos

Se a programação fosse mais engenharia e menos arte, talvez não precisássemos tanto usar os dons de Sherlock Holmes para corrigir programas. Infelizmente (ou será felizmente ??) tal não ocorre, e um programador profissional gasta mais de 50% do seu tempo procurando e corrigindo erros em programas.

É tarefa difícil, sutil, complexa e delicada. s vezes um ponto fora do lugar põe a perder um imenso projeto. É tão importante a tarefa que se diz que bom programador não é aquele de programa bem, mas aquele que depura bem. A propósito, depurar é **identificar, localizar e corrigir erros, de maneira continuada e integrada para produzir software de qualidade.**

O exercício de hoje, visa estimular esta habilidade. Você vai receber alguns códigos e alguns resultados, e deverá inferir quais as inicializações originais das variáveis em cada código.

Por exemplo, no algoritmo

```

1: função EXEMPLO
2: A ← _____
3: B ← 2
4: C ← 3
5: enquanto A < 10
6:   se (B mod 3) = 0
7:     A ← A + 1
8:   senão
9:     A ← A + 2
10:  fim{se}
11:  se (C mod 4) = 0
12:    B ← B + 2
13:  senão
14:    A ← A + 1
15:  fim{se}
16:  imprima A, B, C
17: fim{enquanto}
    
```

Analisando as saídas, que são:

4 2 3
 7 2 3
 10 2 3

Percebe-se que a inicialização original do A foi 1. A resposta deste exemplo portanto, é 1.

Para você fazer

A seguir, diversos algoritmos. Em cada um deles, você deve dizer qual o valor inicial da variável A.

Exercício 1

```

1: função A1
2: A ← _____
3: B ← 8
4: C ← 7
5: enquanto (A + B) > 4
6:   se A > C
7:     A ← A - 1
8:   senão
9:     A ← A + 1
    
```

```

10:  fim{se}
11:  B ← B - 2
12:  imprima A, B, C
13: fim{enquanto}

Esta execução deu como resultado
6 6 7
7 4 7
8 2 7
7 0 7
8 -2 7
7 -4 7
    
```

Exercício 2

```

1: função A2
2: A ← _____
3: B ← 12
4: C ← 10
5: enquanto (C ≠ 0) ∧ (C ≠ 1)
6:   se (B mod 5) ≠ 0
7:     B ← B - 1
8:   fim{se}
9:   se (A mod 3) = 0
10:    A ← A + 1
11:  fim{se}
12:  se (A > B)
13:    A ← A + 2
14:  fim{se}
15:  C ← C - 1
16:  imprima A, B, C
17: fim{enquanto}
    
```

Esta execução deu como resultado

16 11 9
 18 10 8
 21 10 7
 24 10 6
 27 10 5
 30 10 4
 33 10 3
 36 10 2
 39 10 1

Exercício 3

```

1: função A3
2: A ← _____
3: B ← 4
4: C ← 5
5: repita
6:   A ← A + 1
7:   B ← B + 1
8:   C ← C + 1
9: enquanto C > 10
10:  C ← C - 2
11: fim{enquanto}
12: se (C ≠ 3) ∨ (B ≠ 2)
13:  A ← A + 2
14: fim{se}
15: imprima A, B, C
16: até (A + B + C) > 30
    
```

Esta execução deu como resultado

5 5 6
 8 6 7
 11 7 8
 14 8 9

Exercício 4

```

1: função A4
2: A ← _____
3: B ← 15
4: C ← 11
5: enquanto (A > 10) ∨ (B > 10) ∨ (C > 10)
6:   se A > B
7:     C ← C - 1
8:   fim{se}
9:   se (B mod 3) = 1
10:    A ← A - 1
11:  fim{se}
12:  se C > B
13:    B ← B - 3
14:  fim{se}
15:  imprima A, B, C
16:  A ← A - 1
17:  B ← B - 1
18:  C ← C - 2
19: fim{enquanto}
    
```

Esta execução deu como resultado

15 15 11
 14 14 9
 12 13 7
 11 12 5
 10 11 3

Respostas

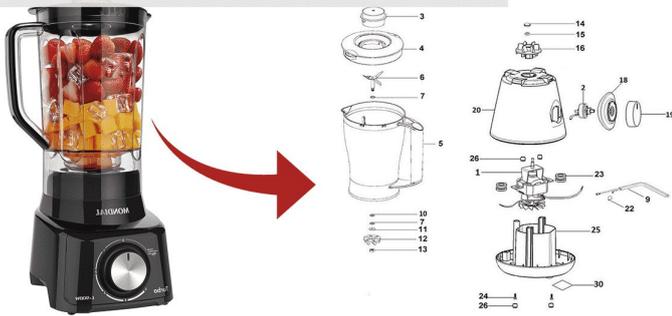
Escreva aqui os valores presumidos para as 4 variáveis "A" de cada um dos 4 algoritmos

1	2	3	4

75815

Engenharia Reversa

ENGENHARIA REVERSA



Engenharia Reversa de algoritmos

Se a programação fosse mais engenharia e menos arte, talvez não precisássemos tanto usar os dons de Sherlock Holmes para corrigir programas. Infelizmente (ou será felizmente ??) tal não ocorre, e um programador profissional gasta mais de 50% do seu tempo procurando e corrigindo erros em programas.

É tarefa difícil, sutil, complexa e delicada. s vezes um ponto fora do lugar põe a perder um imenso projeto. É tão importante a tarefa que se diz que bom programador não é aquele de programa bem, mas aquele que depura bem. A propósito, depurar é **identificar, localizar e corrigir erros, de maneira continuada e integrada para produzir software de qualidade.**

O exercício de hoje, visa estimular esta habilidade. Você vai receber alguns códigos e alguns resultados, e deverá inferir quais as inicializações originais das variáveis em cada código.

Por exemplo, no algoritmo

```

1: função EXEMPLO
2: A ← _____
3: B ← 2
4: C ← 3
5: enquanto A < 10
6:   se (B mod 3) = 0
7:     A ← A + 1
8:   senão
9:     A ← A + 2
10:  fim{se}
11:  se (C mod 4) = 0
12:    B ← B + 2
13:  senão
14:    A ← A + 1
15:  fim{se}
16:  imprima A, B, C
17: fim{enquanto}
    
```

Analisando as saídas, que são:

4 2 3
 7 2 3
 10 2 3

Percebe-se que a inicialização original do A foi 1. A resposta deste exemplo portanto, é 1.

Para você fazer

A seguir, diversos algoritmos. Em cada um deles, você deve dizer qual o valor inicial da variável A.

Exercício 1

```

1: função A1
2: A ← _____
3: B ← 6
4: C ← 13
5: enquanto (A + B) > 4
6:   se A > C
7:     A ← A - 1
8:   senão
9:     A ← A + 1
    
```

```

10:  fim{se}
11:  B ← B - 2
12:  imprima A, B, C
13: fim{enquanto}
    
```

Esta execução deu como resultado

7 4 13
 8 2 13
 9 0 13
 10 -2 13
 11 -4 13
 12 -6 13
 13 -8 13
 14 -10 13

Exercício 2

```

1: função A2
2: A ← _____
3: B ← 14
4: C ← 9
5: enquanto (C ≠ 0) ∧ (C ≠ 1)
6:   se (B mod 5) ≠ 0
7:     B ← B - 1
8:   fim{se}
9:   se (A mod 3) = 0
10:    A ← A + 1
11:  fim{se}
12:  se (A > B)
13:    A ← A + 2
14:  fim{se}
15:  C ← C - 1
16:  imprima A, B, C
17: fim{enquanto}
    
```

Esta execução deu como resultado

16 13 8
 18 12 7
 21 11 6
 24 10 5
 27 10 4
 30 10 3
 33 10 2
 36 10 1

Exercício 3

```

1: função A3
2: A ← _____
3: B ← 3
4: C ← 2
5: repita
6:   A ← A + 1
7:   B ← B + 1
8:   C ← C + 1
9: enquanto C > 10
10:  C ← C - 2
11: fim{enquanto}
12: se (C ≠ 3) ∨ (B ≠ 2)
13:  A ← A + 2
14: fim{se}
15: imprima A, B, C
16: até (A + B + C) > 30
    
```

Esta execução deu como resultado

4 4 3
 7 5 4
 10 6 5

13 7 6
 16 8 7

Exercício 4

```

1: função A4
2: A ← _____
3: B ← 11
4: C ← 13
5: enquanto (A > 10) ∨ (B > 10) ∨ (C > 10)
6:   se A > B
7:     C ← C - 1
8:   fim{se}
9:   se (B mod 3) = 1
10:    A ← A - 1
11:  fim{se}
12:  se C > B
13:    B ← B - 3
14:  fim{se}
15:  imprima A, B, C
16:  A ← A - 1
17:  B ← B - 1
18:  C ← C - 2
19: fim{enquanto}
    
```

Esta execução deu como resultado

13 8 12
 11 4 9

Respostas

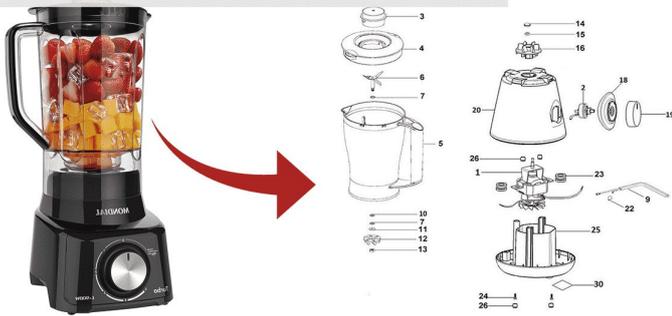
Escreva aqui os valores presumidos para as 4 variáveis "A" de cada um dos 4 algoritmos

1	2	3	4

75822

Engenharia Reversa

ENGENHARIA REVERSA



Engenharia Reversa de algoritmos

Se a programação fosse mais engenharia e menos arte, talvez não precisássemos tanto usar os dons de Sherlock Holmes para corrigir programas. Infelizmente (ou será felizmente ??) tal não ocorre, e um programador profissional gasta mais de 50% do seu tempo procurando e corrigindo erros em programas.

É tarefa difícil, sutil, complexa e delicada. s vezes um ponto fora do lugar põe a perder um imenso projeto. É tão importante a tarefa que se diz que bom programador não é aquele de programa bem, mas aquele que depura bem. A propósito, depurar é **identificar, localizar e corrigir erros, de maneira continuada e integrada para produzir software de qualidade.**

O exercício de hoje, visa estimular esta habilidade. Você vai receber alguns códigos e alguns resultados, e deverá inferir quais as inicializações originais das variáveis em cada código.

Por exemplo, no algoritmo

```

1: função EXEMPLO
2: A ← _____
3: B ← 2
4: C ← 3
5: enquanto A < 10
6:   se (B mod 3) = 0
7:     A ← A + 1
8:   senão
9:     A ← A + 2
10:  fim{se}
11:  se (C mod 4) = 0
12:    B ← B + 2
13:  senão
14:    A ← A + 1
15:  fim{se}
16:  imprima A, B, C
17: fim{enquanto}
    
```

Analisando as saídas, que são:

4 2 3
 7 2 3
 10 2 3

Percebe-se que a inicialização original do A foi 1. A resposta deste exemplo portanto, é 1.

Para você fazer

A seguir, diversos algoritmos. Em cada um deles, você deve dizer qual o valor inicial da variável A.

Exercício 1

```

1: função A1
2: A ← _____
3: B ← 9
4: C ← 7
5: enquanto (A + B) > 4
6:   se A > C
7:     A ← A - 1
8:   senão
9:     A ← A + 1
    
```

```

10:  fim{se}
11:  B ← B - 2
12:  imprima A, B, C
13: fim{enquanto}

Esta execução deu como resultado
6 7 7
7 5 7
8 3 7
7 1 7
8 -1 7
7 -3 7
    
```

Exercício 2

```

1: função A2
2: A ← _____
3: B ← 12
4: C ← 11
5: enquanto (C ≠ 0) ∧ (C ≠ 1)
6:   se (B mod 5) ≠ 0
7:     B ← B - 1
8:   fim{se}
9:   se (A mod 3) = 0
10:    A ← A + 1
11:  fim{se}
12:  se (A > B)
13:    A ← A + 2
14:  fim{se}
15:  C ← C - 1
16:  imprima A, B, C
17: fim{enquanto}
    
```

Esta execução deu como resultado

11 11 10
 13 10 9
 15 10 8
 18 10 7
 21 10 6
 24 10 5
 27 10 4
 30 10 3
 33 10 2
 36 10 1

Exercício 3

```

1: função A3
2: A ← _____
3: B ← 1
4: C ← 4
5: repita
6:   A ← A + 1
7:   B ← B + 1
8:   C ← C + 1
9: enquanto C > 10
10:  C ← C - 2
11: fim{enquanto}
12: se (C ≠ 3) ∨ (B ≠ 2)
13:  A ← A + 2
14: fim{se}
15: imprima A, B, C
16: até (A + B + C) > 30
    
```

Esta execução deu como resultado

7 2 5
 10 3 6
 13 4 7

16 5 8
 19 6 9

Exercício 4

```

1: função A4
2: A ← _____
3: B ← 14
4: C ← 12
5: enquanto (A > 10) ∨ (B > 10) ∨ (C > 10)
6:   se A > B
7:     C ← C - 1
8:   fim{se}
9:   se (B mod 3) = 1
10:    A ← A - 1
11:  fim{se}
12:  se C > B
13:    B ← B - 3
14:  fim{se}
15:  imprima A, B, C
16:  A ← A - 1
17:  B ← B - 1
18:  C ← C - 2
19: fim{enquanto}
    
```

Esta execução deu como resultado

15 14 11
 13 13 8
 12 12 6
 11 11 4

Respostas

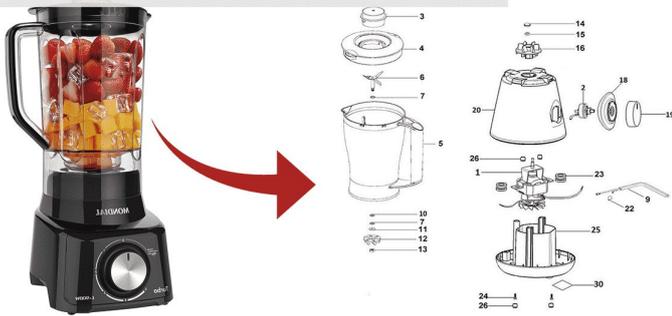
Escreva aqui os valores presumidos para as 4 variáveis "A" de cada um dos 4 algoritmos

1	2	3	4

75839

Engenharia Reversa

ENGENHARIA REVERSA



Engenharia Reversa de algoritmos

Se a programação fosse mais engenharia e menos arte, talvez não precisássemos tanto usar os dons de Sherlock Holmes para corrigir programas. Infelizmente (ou será felizmente ??) tal não ocorre, e um programador profissional gasta mais de 50% do seu tempo procurando e corrigindo erros em programas.

É tarefa difícil, sutil, complexa e delicada. s vezes um ponto fora do lugar põe a perder um imenso projeto. É tão importante a tarefa que se diz que bom programador não é aquele de programa bem, mas aquele que depura bem. A propósito, depurar é **identificar, localizar e corrigir erros, de maneira continuada e integrada para produzir software de qualidade.**

O exercício de hoje, visa estimular esta habilidade. Você vai receber alguns códigos e alguns resultados, e deverá inferir quais as inicializações originais das variáveis em cada código.

Por exemplo, no algoritmo

```

1: função EXEMPLO
2: A ← _____
3: B ← 2
4: C ← 3
5: enquanto A < 10
6:   se (B mod 3) = 0
7:     A ← A + 1
8:   senão
9:     A ← A + 2
10:  fim{se}
11:  se (C mod 4) = 0
12:    B ← B + 2
13:  senão
14:    A ← A + 1
15:  fim{se}
16:  imprima A, B, C
17: fim{enquanto}
  
```

Analisando as saídas, que são:

4 2 3
 7 2 3
 10 2 3

Percebe-se que a inicialização original do A foi 1. A resposta deste exemplo portanto, é 1.

Para você fazer

A seguir, diversos algoritmos. Em cada um deles, você deve dizer qual o valor inicial da variável A.

Exercício 1

```

1: função A1
2: A ← _____
3: B ← 8
4: C ← 10
5: enquanto (A + B) > 4
6:   se A > C
7:     A ← A - 1
8:   senão
9:     A ← A + 1
  
```

```

10:  fim{se}
11:  B ← B - 2
12:  imprima A, B, C
13: fim{enquanto}

Esta execução deu como resultado
8 6 10
9 4 10
10 2 10
11 0 10
10 -2 10
11 -4 10
10 -6 10
  
```

Exercício 2

```

1: função A2
2: A ← _____
3: B ← 12
4: C ← 9
5: enquanto (C ≠ 0) ∧ (C ≠ 1)
6:   se (B mod 5) ≠ 0
7:     B ← B - 1
8:   fim{se}
9:   se (A mod 3) = 0
10:    A ← A + 1
11:  fim{se}
12:  se (A > B)
13:    A ← A + 2
14:  fim{se}
15:  C ← C - 1
16:  imprima A, B, C
17: fim{enquanto}
  
```

Esta execução deu como resultado

16 11 8
 18 10 7
 21 10 6
 24 10 5
 27 10 4
 30 10 3
 33 10 2
 36 10 1

Exercício 3

```

1: função A3
2: A ← _____
3: B ← 4
4: C ← 5
5: repita
6:   A ← A + 1
7:   B ← B + 1
8:   C ← C + 1
9: enquanto C > 10
10:  C ← C - 2
11: fim{enquanto}
12: se (C ≠ 3) ∨ (B ≠ 2)
13:  A ← A + 2
14: fim{se}
15: imprima A, B, C
16: até (A + B + C) > 30
  
```

Esta execução deu como resultado

8 5 6
 11 6 7
 14 7 8
 17 8 9

Exercício 4

```

1: função A4
2: A ← _____
3: B ← 14
4: C ← 12
5: enquanto (A > 10) ∨ (B > 10) ∨ (C > 10)
6:   se A > B
7:     C ← C - 1
8:   fim{se}
9:   se (B mod 3) = 1
10:    A ← A - 1
11:  fim{se}
12:  se C > B
13:    B ← B - 3
14:  fim{se}
15:  imprima A, B, C
16:  A ← A - 1
17:  B ← B - 1
18:  C ← C - 2
19: fim{enquanto}
  
```

Esta execução deu como resultado

15 14 11
 13 13 8
 12 12 6
 11 11 4

Respostas

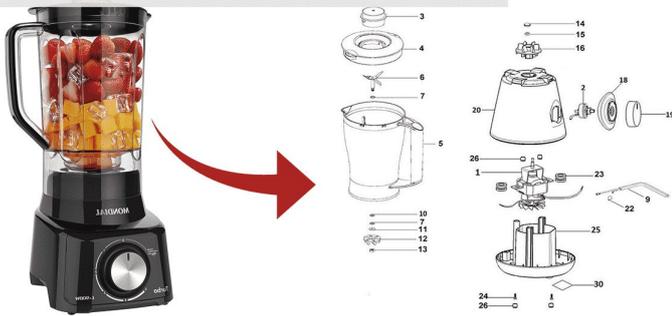
Escreva aqui os valores presumidos para as 4 variáveis "A" de cada um dos 4 algoritmos

1	2	3	4

75846

Engenharia Reversa

ENGENHARIA REVERSA



Engenharia Reversa de algoritmos

Se a programação fosse mais engenharia e menos arte, talvez não precisássemos tanto usar os dons de Sherlock Holmes para corrigir programas. Infelizmente (ou será felizmente ??) tal não ocorre, e um programador profissional gasta mais de 50% do seu tempo procurando e corrigindo erros em programas.

É tarefa difícil, sutil, complexa e delicada. Às vezes um ponto fora do lugar põe a perder um imenso projeto. É tão importante a tarefa que se diz que bom programador não é aquele de programa bem, mas aquele que depura bem. A propósito, depurar é **identificar, localizar e corrigir erros, de maneira continuada e integrada para produzir software de qualidade.**

O exercício de hoje, visa estimular esta habilidade. Você vai receber alguns códigos e alguns resultados, e deverá inferir quais as inicializações originais das variáveis em cada código.

Por exemplo, no algoritmo

```

1: função EXEMPLO
2: A ← _____
3: B ← 2
4: C ← 3
5: enquanto A < 10
6:   se (B mod 3) = 0
7:     A ← A + 1
8:   senão
9:     A ← A + 2
10:  fim{se}
11:  se (C mod 4) = 0
12:    B ← B + 2
13:  senão
14:    A ← A + 1
15:  fim{se}
16:  imprima A, B, C
17: fim{enquanto}
    
```

Analisando as saídas, que são:

4 2 3
 7 2 3
 10 2 3

Percebe-se que a inicialização original do A foi 1. A resposta deste exemplo portanto, é 1.

Para você fazer

A seguir, diversos algoritmos. Em cada um deles, você deve dizer qual o valor inicial da variável A.

Exercício 1

```

1: função A1
2: A ← _____
3: B ← 10
4: C ← 9
5: enquanto (A + B) > 4
6:   se A > C
7:     A ← A - 1
8:   senão
9:     A ← A + 1
    
```

```

10:  fim{se}
11:  B ← B - 2
12:  imprima A, B, C
13: fim{enquanto}
    
```

Esta execução deu como resultado

6 8 9
 7 6 9
 8 4 9
 9 2 9
 10 0 9
 9 -2 9
 10 -4 9
 9 -6 9

Exercício 2

```

1: função A2
2: A ← _____
3: B ← 11
4: C ← 9
5: enquanto (C ≠ 0) ∧ (C ≠ 1)
6:   se (B mod 5) ≠ 0
7:     B ← B - 1
8:   fim{se}
9:   se (A mod 3) = 0
10:    A ← A + 1
11:  fim{se}
12:  se (A > B)
13:    A ← A + 2
14:  fim{se}
15:  C ← C - 1
16:  imprima A, B, C
17: fim{enquanto}
    
```

Esta execução deu como resultado

13 10 8
 15 10 7
 18 10 6
 21 10 5
 24 10 4
 27 10 3
 30 10 2
 33 10 1

Exercício 3

```

1: função A3
2: A ← _____
3: B ← 2
4: C ← 2
5: repita
6:   A ← A + 1
7:   B ← B + 1
8:   C ← C + 1
9: enquanto C > 10
10:  C ← C - 2
11: fim{enquanto}
12: se (C ≠ 3) ∨ (B ≠ 2)
13:  A ← A + 2
14: fim{se}
15: imprima A, B, C
16: até (A + B + C) > 30
    
```

Esta execução deu como resultado

4 3 3
 7 4 4
 10 5 5

13 6 6
 16 7 7
 19 8 8

Exercício 4

```

1: função A4
2: A ← _____
3: B ← 15
4: C ← 13
5: enquanto (A > 10) ∨ (B > 10) ∨ (C > 10)
6:   se A > B
7:     C ← C - 1
8:   fim{se}
9:   se (B mod 3) = 1
10:    A ← A - 1
11:  fim{se}
12:  se C > B
13:    B ← B - 3
14:  fim{se}
15:  imprima A, B, C
16:  A ← A - 1
17:  B ← B - 1
18:  C ← C - 2
19: fim{enquanto}
    
```

Esta execução deu como resultado

12 15 13
 11 14 11
 9 13 9
 8 12 7
 7 11 5

Respostas

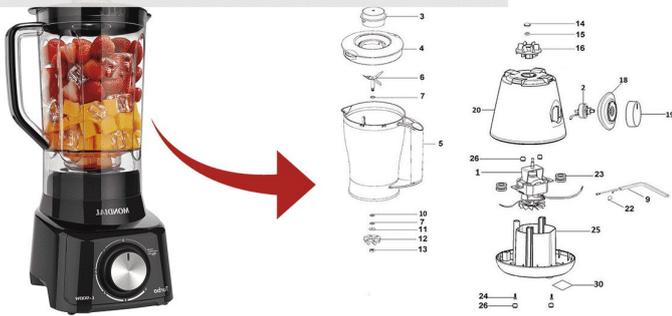
Escreva aqui os valores presumidos para as 4 variáveis "A" de cada um dos 4 algoritmos

1	2	3	4

75853

Engenharia Reversa

ENGENHARIA REVERSA



Engenharia Reversa de algoritmos

Se a programação fosse mais engenharia e menos arte, talvez não precisássemos tanto usar os dons de Sherlock Holmes para corrigir programas. Infelizmente (ou será felizmente ??) tal não ocorre, e um programador profissional gasta mais de 50% do seu tempo procurando e corrigindo erros em programas.

É tarefa difícil, sutil, complexa e delicada. s vezes um ponto fora do lugar põe a perder um imenso projeto. É tão importante a tarefa que se diz que bom programador não é aquele de programa bem, mas aquele que depura bem. A propósito, depurar é **identificar, localizar e corrigir erros, de maneira continuada e integrada para produzir software de qualidade.**

O exercício de hoje, visa estimular esta habilidade. Você vai receber alguns códigos e alguns resultados, e deverá inferir quais as inicializações originais das variáveis em cada código.

Por exemplo, no algoritmo

```

1: função EXEMPLO
2: A ← _____
3: B ← 2
4: C ← 3
5: enquanto A < 10
6:   se (B mod 3) = 0
7:     A ← A + 1
8:   senão
9:     A ← A + 2
10:  fim{se}
11:  se (C mod 4) = 0
12:    B ← B + 2
13:  senão
14:    A ← A + 1
15:  fim{se}
16:  imprima A, B, C
17: fim{enquanto}
  
```

Analizando as saídas, que são:

4 2 3
 7 2 3
 10 2 3

Percebe-se que a inicialização original do A foi 1. A resposta deste exemplo portanto, é 1.

Para você fazer

A seguir, diversos algoritmos. Em cada um deles, você deve dizer qual o valor inicial da variável A.

Exercício 1

```

1: função A1
2: A ← _____
3: B ← 6
4: C ← 14
5: enquanto (A + B) > 4
6:   se A > C
7:     A ← A - 1
8:   senão
9:     A ← A + 1
  
```

```

10:  fim{se}
11:  B ← B - 2
12:  imprima A, B, C
13: fim{enquanto}
  
```

Esta execução deu como resultado

6 4 14
 7 2 14
 8 0 14
 9 -2 14
 10 -4 14
 11 -6 14
 12 -8 14

Exercício 2

```

1: função A2
2: A ← _____
3: B ← 13
4: C ← 10
5: enquanto (C ≠ 0) ∧ (C ≠ 1)
6:   se (B mod 5) ≠ 0
7:     B ← B - 1
8:   fim{se}
9:   se (A mod 3) = 0
10:    A ← A + 1
11:  fim{se}
12:  se (A > B)
13:    A ← A + 2
14:  fim{se}
15:  C ← C - 1
16:  imprima A, B, C
17: fim{enquanto}
  
```

Esta execução deu como resultado

16 12 9
 18 11 8
 21 10 7
 24 10 6
 27 10 5
 30 10 4
 33 10 3
 36 10 2
 39 10 1

Exercício 3

```

1: função A3
2: A ← _____
3: B ← 2
4: C ← 1
5: repita
6:   A ← A + 1
7:   B ← B + 1
8:   C ← C + 1
9: enquanto C > 10
10:  C ← C - 2
11: fim{enquanto}
12: se (C ≠ 3) ∨ (B ≠ 2)
13:  A ← A + 2
14: fim{se}
15: imprima A, B, C
16: até (A + B + C) > 30
  
```

Esta execução deu como resultado

7 3 2
 10 4 3
 13 5 4

16 6 5
 19 7 6

Exercício 4

```

1: função A4
2: A ← _____
3: B ← 11
4: C ← 11
5: enquanto (A > 10) ∨ (B > 10) ∨ (C > 10)
6:   se A > B
7:     C ← C - 1
8:   fim{se}
9:   se (B mod 3) = 1
10:    A ← A - 1
11:  fim{se}
12:  se C > B
13:    B ← B - 3
14:  fim{se}
15:  imprima A, B, C
16:  A ← A - 1
17:  B ← B - 1
18:  C ← C - 2
19: fim{enquanto}
  
```

Esta execução deu como resultado

12 11 10
 10 10 7

Respostas

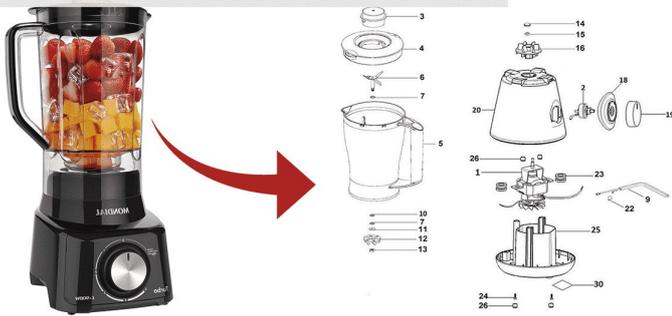
Escreva aqui os valores presumidos para as 4 variáveis "A" de cada um dos 4 algoritmos

1	2	3	4

75860

Engenharia Reversa

ENGENHARIA REVERSA



Engenharia Reversa de algoritmos

Se a programação fosse mais engenharia e menos arte, talvez não precisássemos tanto usar os dons de Sherlock Holmes para corrigir programas. Infelizmente (ou será felizmente ??) tal não ocorre, e um programador profissional gasta mais de 50% do seu tempo procurando e corrigindo erros em programas.

É tarefa difícil, sutil, complexa e delicada. s vezes um ponto fora do lugar põe a perder um imenso projeto. É tão importante a tarefa que se diz que bom programador não é aquele de programa bem, mas aquele que depura bem. A propósito, depurar é **identificar, localizar e corrigir erros, de maneira continuada e integrada para produzir software de qualidade.**

O exercício de hoje, visa estimular esta habilidade. Você vai receber alguns códigos e alguns resultados, e deverá inferir quais as inicializações originais das variáveis em cada código.

Por exemplo, no algoritmo

```

1: função EXEMPLO
2: A ← _____
3: B ← 2
4: C ← 3
5: enquanto A < 10
6:   se (B mod 3) = 0
7:     A ← A + 1
8:   senão
9:     A ← A + 2
10:  fim{se}
11:  se (C mod 4) = 0
12:    B ← B + 2
13:  senão
14:    A ← A + 1
15:  fim{se}
16:  imprima A, B, C
17: fim{enquanto}
    
```

Analisando as saídas, que são:

4 2 3
 7 2 3
 10 2 3

Percebe-se que a inicialização original do A foi 1. A resposta deste exemplo portanto, é 1.

Para você fazer

A seguir, diversos algoritmos. Em cada um deles, você deve dizer qual o valor inicial da variável A.

Exercício 1

```

1: função A1
2: A ← _____
3: B ← 8
4: C ← 11
5: enquanto (A + B) > 4
6:   se A > C
7:     A ← A - 1
8:   senão
9:     A ← A + 1
    
```

```

10:  fim{se}
11:  B ← B - 2
12:  imprima A, B, C
13: fim{enquanto}
    
```

Esta execução deu como resultado

6 6 11
 7 4 11
 8 2 11
 9 0 11
 10 -2 11
 11 -4 11
 12 -6 11
 11 -8 11

Exercício 2

```

1: função A2
2: A ← _____
3: B ← 14
4: C ← 10
5: enquanto (C ≠ 0) ∧ (C ≠ 1)
6:   se (B mod 5) ≠ 0
7:     B ← B - 1
8:   fim{se}
9:   se (A mod 3) = 0
10:    A ← A + 1
11:  fim{se}
12:  se (A > B)
13:    A ← A + 2
14:  fim{se}
15:  C ← C - 1
16:  imprima A, B, C
17: fim{enquanto}
    
```

Esta execução deu como resultado

16 13 9
 18 12 8
 21 11 7
 24 10 6
 27 10 5
 30 10 4
 33 10 3
 36 10 2
 39 10 1

Exercício 3

```

1: função A3
2: A ← _____
3: B ← 5
4: C ← 1
5: repita
6:   A ← A + 1
7:   B ← B + 1
8:   C ← C + 1
9: enquanto C > 10
10:   C ← C - 2
11: fim{enquanto}
12: se (C ≠ 3) ∨ (B ≠ 2)
13:   A ← A + 2
14: fim{se}
15: imprima A, B, C
16: até (A + B + C) > 30
    
```

Esta execução deu como resultado

6 6 2
 9 7 3

12 8 4
 15 9 5
 18 10 6

Exercício 4

```

1: função A4
2: A ← _____
3: B ← 15
4: C ← 11
5: enquanto (A > 10) ∨ (B > 10) ∨ (C > 10)
6:   se A > B
7:     C ← C - 1
8:   fim{se}
9:   se (B mod 3) = 1
10:    A ← A - 1
11:  fim{se}
12:  se C > B
13:    B ← B - 3
14:  fim{se}
15:  imprima A, B, C
16:  A ← A - 1
17:  B ← B - 1
18:  C ← C - 2
19: fim{enquanto}
    
```

Esta execução deu como resultado

13 15 11
 12 14 9
 10 13 7
 9 12 5
 8 11 3

Respostas

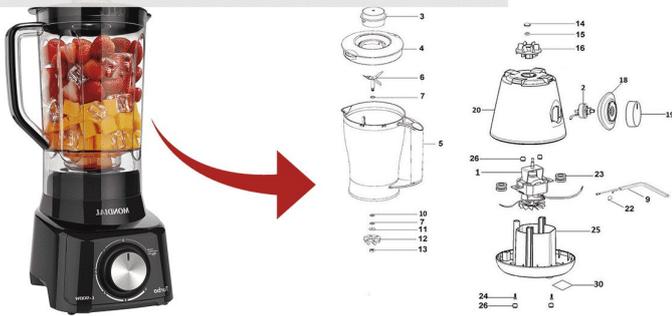
Escreva aqui os valores presumidos para as 4 variáveis "A" de cada um dos 4 algoritmos

1	2	3	4

75877

Engenharia Reversa

ENGENHARIA REVERSA



Engenharia Reversa de algoritmos

Se a programação fosse mais engenharia e menos arte, talvez não precisássemos tanto usar os dons de Sherlock Holmes para corrigir programas. Infelizmente (ou será felizmente ??) tal não ocorre, e um programador profissional gasta mais de 50% do seu tempo procurando e corrigindo erros em programas.

É tarefa difícil, sutil, complexa e delicada. s vezes um ponto fora do lugar põe a perder um imenso projeto. É tão importante a tarefa que se diz que bom programador não é aquele de programa bem, mas aquele que depura bem. A propósito, depurar é **identificar, localizar e corrigir erros, de maneira continuada e integrada para produzir software de qualidade.**

O exercício de hoje, visa estimular esta habilidade. Você vai receber alguns códigos e alguns resultados, e deverá inferir quais as inicializações originais das variáveis em cada código.

Por exemplo, no algoritmo

```

1: função EXEMPLO
2: A ← _____
3: B ← 2
4: C ← 3
5: enquanto A < 10
6:   se (B mod 3) = 0
7:     A ← A + 1
8:   senão
9:     A ← A + 2
10:  fim{se}
11:  se (C mod 4) = 0
12:    B ← B + 2
13:  senão
14:    A ← A + 1
15:  fim{se}
16:  imprima A, B, C
17: fim{enquanto}
  
```

Analizando as saídas, que são:

4 2 3
 7 2 3
 10 2 3

Percebe-se que a inicialização original do A foi 1. A resposta deste exemplo portanto, é 1.

Para você fazer

A seguir, diversos algoritmos. Em cada um deles, você deve dizer qual o valor inicial da variável A.

Exercício 1

```

1: função A1
2: A ← _____
3: B ← 9
4: C ← 10
5: enquanto (A + B) > 4
6:   se A > C
7:     A ← A - 1
8:   senão
9:     A ← A + 1
  
```

```

10:  fim{se}
11:  B ← B - 2
12:  imprima A, B, C
13: fim{enquanto}
  
```

Esta execução deu como resultado

7 7 10
 8 5 10
 9 3 10
 10 1 10
 11 -1 10
 10 -3 10
 11 -5 10
 10 -7 10

Exercício 2

```

1: função A2
2: A ← _____
3: B ← 10
4: C ← 9
5: enquanto (C ≠ 0) ∧ (C ≠ 1)
6:   se (B mod 5) ≠ 0
7:     B ← B - 1
8:   fim{se}
9:   se (A mod 3) = 0
10:    A ← A + 1
11:  fim{se}
12:  se (A > B)
13:    A ← A + 2
14:  fim{se}
15:  C ← C - 1
16:  imprima A, B, C
17: fim{enquanto}
  
```

Esta execução deu como resultado

16 10 8
 18 10 7
 21 10 6
 24 10 5
 27 10 4
 30 10 3
 33 10 2
 36 10 1

Exercício 3

```

1: função A3
2: A ← _____
3: B ← 5
4: C ← 1
5: repita
6:   A ← A + 1
7:   B ← B + 1
8:   C ← C + 1
9: enquanto C > 10
10:  C ← C - 2
11: fim{enquanto}
12: se (C ≠ 3) ∨ (B ≠ 2)
13:  A ← A + 2
14: fim{se}
15: imprima A, B, C
16: até (A + B + C) > 30
  
```

Esta execução deu como resultado

6 6 2
 9 7 3
 12 8 4

15 9 5
 18 10 6

Exercício 4

```

1: função A4
2: A ← _____
3: B ← 11
4: C ← 13
5: enquanto (A > 10) ∨ (B > 10) ∨ (C > 10)
6:   se A > B
7:     C ← C - 1
8:   fim{se}
9:   se (B mod 3) = 1
10:    A ← A - 1
11:  fim{se}
12:  se C > B
13:    B ← B - 3
14:  fim{se}
15:  imprima A, B, C
16:  A ← A - 1
17:  B ← B - 1
18:  C ← C - 2
19: fim{enquanto}
  
```

Esta execução deu como resultado

14 8 12
 12 4 9
 11 0 6

Respostas

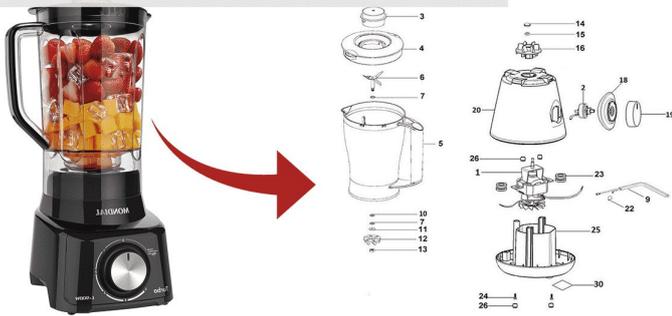
Escreva aqui os valores presumidos para as 4 variáveis "A" de cada um dos 4 algoritmos

1	2	3	4

75884

Engenharia Reversa

ENGENHARIA REVERSA



Engenharia Reversa de algoritmos

Se a programação fosse mais engenharia e menos arte, talvez não precisássemos tanto usar os dons de Sherlock Holmes para corrigir programas. Infelizmente (ou será felizmente ??) tal não ocorre, e um programador profissional gasta mais de 50% do seu tempo procurando e corrigindo erros em programas.

É tarefa difícil, sutil, complexa e delicada. s vezes um ponto fora do lugar põe a perder um imenso projeto. É tão importante a tarefa que se diz que bom programador não é aquele de programa bem, mas aquele que depura bem. A propósito, depurar é **identificar, localizar e corrigir erros, de maneira continuada e integrada para produzir software de qualidade.**

O exercício de hoje, visa estimular esta habilidade. Você vai receber alguns códigos e alguns resultados, e deverá inferir quais as inicializações originais das variáveis em cada código.

Por exemplo, no algoritmo

```

1: função EXEMPLO
2: A ← _____
3: B ← 2
4: C ← 3
5: enquanto A < 10
6:   se (B mod 3) = 0
7:     A ← A + 1
8:   senão
9:     A ← A + 2
10:  fim{se}
11:  se (C mod 4) = 0
12:    B ← B + 2
13:  senão
14:    A ← A + 1
15:  fim{se}
16:  imprima A, B, C
17: fim{enquanto}
    
```

Analisando as saídas, que são:

4 2 3
 7 2 3
 10 2 3

Percebe-se que a inicialização original do A foi 1. A resposta deste exemplo portanto, é 1.

Para você fazer

A seguir, diversos algoritmos. Em cada um deles, você deve dizer qual o valor inicial da variável A.

Exercício 1

```

1: função A1
2: A ← _____
3: B ← 6
4: C ← 12
5: enquanto (A + B) > 4
6:   se A > C
7:     A ← A - 1
8:   senão
9:     A ← A + 1
    
```

```

10:  fim{se}
11:  B ← B - 2
12:  imprima A, B, C
13: fim{enquanto}
    
```

Esta execução deu como resultado

6 4 12
 7 2 12
 8 0 12
 9 -2 12
 10 -4 12
 11 -6 12
 12 -8 12

Exercício 2

```

1: função A2
2: A ← _____
3: B ← 13
4: C ← 10
5: enquanto (C ≠ 0) ∧ (C ≠ 1)
6:   se (B mod 5) ≠ 0
7:     B ← B - 1
8:   fim{se}
9:   se (A mod 3) = 0
10:    A ← A + 1
11:  fim{se}
12:  se (A > B)
13:    A ← A + 2
14:  fim{se}
15:  C ← C - 1
16:  imprima A, B, C
17: fim{enquanto}
    
```

Esta execução deu como resultado

16 12 9
 18 11 8
 21 10 7
 24 10 6
 27 10 5
 30 10 4
 33 10 3
 36 10 2
 39 10 1

Exercício 3

```

1: função A3
2: A ← _____
3: B ← 1
4: C ← 2
5: repita
6:   A ← A + 1
7:   B ← B + 1
8:   C ← C + 1
9: enquanto C > 10
10:  C ← C - 2
11: fim{enquanto}
12: se (C ≠ 3) ∨ (B ≠ 2)
13:  A ← A + 2
14: fim{se}
15: imprima A, B, C
16: até (A + B + C) > 30
    
```

Esta execução deu como resultado

4 2 3
 7 3 4
 10 4 5

13 5 6
 16 6 7
 19 7 8

Exercício 4

```

1: função A4
2: A ← _____
3: B ← 15
4: C ← 13
5: enquanto (A > 10) ∨ (B > 10) ∨ (C > 10)
6:   se A > B
7:     C ← C - 1
8:   fim{se}
9:   se (B mod 3) = 1
10:    A ← A - 1
11:  fim{se}
12:  se C > B
13:    B ← B - 3
14:  fim{se}
15:  imprima A, B, C
16:  A ← A - 1
17:  B ← B - 1
18:  C ← C - 2
19: fim{enquanto}
    
```

Esta execução deu como resultado

13 15 13
 12 14 11
 10 13 9
 9 12 7
 8 11 5

Respostas

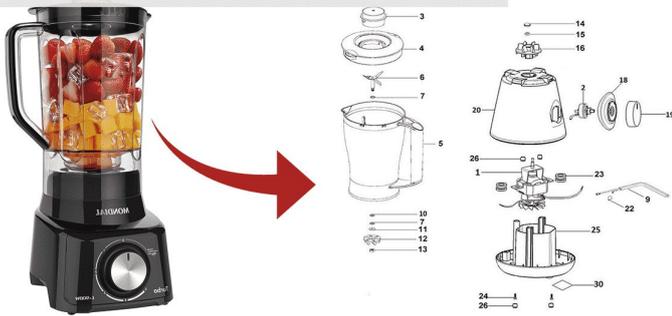
Escreva aqui os valores presumidos para as 4 variáveis "A" de cada um dos 4 algoritmos

1	2	3	4

75989

Engenharia Reversa

ENGENHARIA REVERSA



Engenharia Reversa de algoritmos

Se a programação fosse mais engenharia e menos arte, talvez não precisássemos tanto usar os dons de Sherlock Holmes para corrigir programas. Infelizmente (ou será felizmente ??) tal não ocorre, e um programador profissional gasta mais de 50% do seu tempo procurando e corrigindo erros em programas.

É tarefa difícil, sutil, complexa e delicada. s vezes um ponto fora do lugar põe a perder um imenso projeto. É tão importante a tarefa que se diz que bom programador não é aquele de programa bem, mas aquele que depura bem. A propósito, depurar é **identificar, localizar e corrigir erros, de maneira continuada e integrada para produzir software de qualidade.**

O exercício de hoje, visa estimular esta habilidade. Você vai receber alguns códigos e alguns resultados, e deverá inferir quais as inicializações originais das variáveis em cada código.

Por exemplo, no algoritmo

```

1: função EXEMPLO
2: A ← _____
3: B ← 2
4: C ← 3
5: enquanto A < 10
6:   se (B mod 3) = 0
7:     A ← A + 1
8:   senão
9:     A ← A + 2
10:  fim{se}
11:  se (C mod 4) = 0
12:    B ← B + 2
13:  senão
14:    A ← A + 1
15:  fim{se}
16:  imprima A, B, C
17: fim{enquanto}
    
```

Analisando as saídas, que são:

4 2 3
 7 2 3
 10 2 3

Percebe-se que a inicialização original do A foi 1. A resposta deste exemplo portanto, é 1.

Para você fazer

A seguir, diversos algoritmos. Em cada um deles, você deve dizer qual o valor inicial da variável A.

Exercício 1

```

1: função A1
2: A ← _____
3: B ← 9
4: C ← 10
5: enquanto (A + B) > 4
6:   se A > C
7:     A ← A - 1
8:   senão
9:     A ← A + 1
    
```

```

10:  fim{se}
11:  B ← B - 2
12:  imprima A, B, C
13: fim{enquanto}
    
```

Esta execução deu como resultado

6 7 10
 7 5 10
 8 3 10
 9 1 10
 10 -1 10
 11 -3 10
 10 -5 10
 11 -7 10

Exercício 2

```

1: função A2
2: A ← _____
3: B ← 9
4: C ← 11
5: enquanto (C ≠ 0) ∧ (C ≠ 1)
6:   se (B mod 5) ≠ 0
7:     B ← B - 1
8:   fim{se}
9:   se (A mod 3) = 0
10:    A ← A + 1
11:  fim{se}
12:  se (A > B)
13:    A ← A + 2
14:  fim{se}
15:  C ← C - 1
16:  imprima A, B, C
17: fim{enquanto}
    
```

Esta execução deu como resultado

16 8 10
 18 7 9
 21 6 8
 24 5 7
 27 5 6
 30 5 5
 33 5 4
 36 5 3
 39 5 2
 42 5 1

Exercício 3

```

1: função A3
2: A ← _____
3: B ← 2
4: C ← 5
5: repita
6:   A ← A + 1
7:   B ← B + 1
8:   C ← C + 1
9:   enquanto C > 10
10:    C ← C - 2
11:  fim{enquanto}
12:  se (C ≠ 3) ∨ (B ≠ 2)
13:    A ← A + 2
14:  fim{se}
15:  imprima A, B, C
16: até (A + B + C) > 30
    
```

Esta execução deu como resultado

8 3 6

11 4 7
 14 5 8
 17 6 9

Exercício 4

```

1: função A4
2: A ← _____
3: B ← 13
4: C ← 12
5: enquanto (A > 10) ∨ (B > 10) ∨ (C > 10)
6:   se A > B
7:     C ← C - 1
8:   fim{se}
9:   se (B mod 3) = 1
10:    A ← A - 1
11:  fim{se}
12:  se C > B
13:    B ← B - 3
14:  fim{se}
15:  imprima A, B, C
16:  A ← A - 1
17:  B ← B - 1
18:  C ← C - 2
19: fim{enquanto}
    
```

Esta execução deu como resultado

11 13 12
 10 12 10
 9 11 8

Respostas

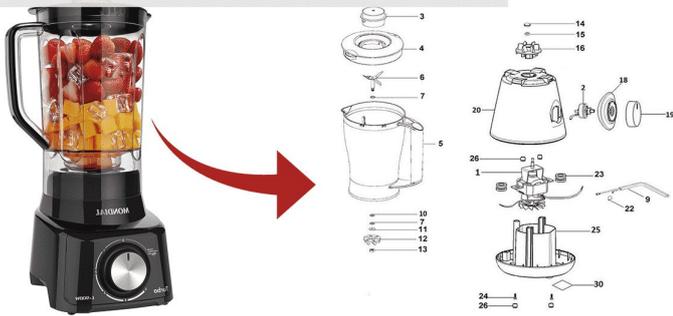
Escreva aqui os valores presumidos para as 4 variáveis "A" de cada um dos 4 algoritmos

1	2	3	4

75891

Engenharia Reversa

ENGENHARIA REVERSA



Engenharia Reversa de algoritmos

Se a programação fosse mais engenharia e menos arte, talvez não precisássemos tanto usar os dons de Sherlock Holmes para corrigir programas. Infelizmente (ou será felizmente ??) tal não ocorre, e um programador profissional gasta mais de 50% do seu tempo procurando e corrigindo erros em programas.

É tarefa difícil, sutil, complexa e delicada. s vezes um ponto fora do lugar põe a perder um imenso projeto. É tão importante a tarefa que se diz que bom programador não é aquele de programa bem, mas aquele que depura bem. A propósito, depurar é **identificar, localizar e corrigir erros, de maneira continuada e integrada para produzir software de qualidade.**

O exercício de hoje, visa estimular esta habilidade. Você vai receber alguns códigos e alguns resultados, e deverá inferir quais as inicializações originais das variáveis em cada código.

Por exemplo, no algoritmo

```
1: função EXEMPLO
2: A ← _____
3: B ← 2
4: C ← 3
5: enquanto A < 10
6:   se (B mod 3) = 0
7:     A ← A + 1
8:   senão
9:     A ← A + 2
10:  fim{se}
11:  se (C mod 4) = 0
12:    B ← B + 2
13:  senão
14:    A ← A + 1
15:  fim{se}
16:  imprima A, B, C
17: fim{enquanto}
```

Analisando as saídas, que são:

4 2 3
7 2 3
10 2 3

Percebe-se que a inicialização original do A foi 1. A resposta deste exemplo portanto, é 1.

Para você fazer

A seguir, diversos algoritmos. Em cada um deles, você deve dizer qual o valor inicial da variável A.

Exercício 1

```
1: função A1
2: A ← _____
3: B ← 7
4: C ← 12
5: enquanto (A + B) > 4
6:   se A > C
7:     A ← A - 1
8:   senão
9:     A ← A + 1
```

```
10:  fim{se}
11:  B ← B - 2
12:  imprima A, B, C
13: fim{enquanto}
```

Esta execução deu como resultado

8 5 12
9 3 12
10 1 12
11 -1 12
12 -3 12
13 -5 12
12 -7 12
13 -9 12

Exercício 2

```
1: função A2
2: A ← _____
3: B ← 14
4: C ← 11
5: enquanto (C ≠ 0) ∧ (C ≠ 1)
6:   se (B mod 5) ≠ 0
7:     B ← B - 1
8:   fim{se}
9:   se (A mod 3) = 0
10:    A ← A + 1
11:  fim{se}
12:  se (A > B)
13:    A ← A + 2
14:  fim{se}
15:  C ← C - 1
16:  imprima A, B, C
17: fim{enquanto}
```

Esta execução deu como resultado

11 13 10
11 12 9
11 11 8
13 10 7
15 10 6
18 10 5
21 10 4
24 10 3
27 10 2
30 10 1

Exercício 3

```
1: função A3
2: A ← _____
3: B ← 1
4: C ← 3
5: repita
6:   A ← A + 1
7:   B ← B + 1
8:   C ← C + 1
9: enquanto C > 10
10:  C ← C - 2
11: fim{enquanto}
12: se (C ≠ 3) ∨ (B ≠ 2)
13:  A ← A + 2
14: fim{se}
15: imprima A, B, C
16: até (A + B + C) > 30
```

Esta execução deu como resultado

5 2 4

8 3 5
11 4 6
14 5 7
17 6 8

Exercício 4

```
1: função A4
2: A ← _____
3: B ← 14
4: C ← 12
5: enquanto (A > 10) ∨ (B > 10) ∨ (C > 10)
6:   se A > B
7:     C ← C - 1
8:   fim{se}
9:   se (B mod 3) = 1
10:    A ← A - 1
11:  fim{se}
12:  se C > B
13:    B ← B - 3
14:  fim{se}
15:  imprima A, B, C
16:  A ← A - 1
17:  B ← B - 1
18:  C ← C - 2
19: fim{enquanto}
```

Esta execução deu como resultado

11 14 12
9 13 10
8 12 8
7 11 6

Respostas

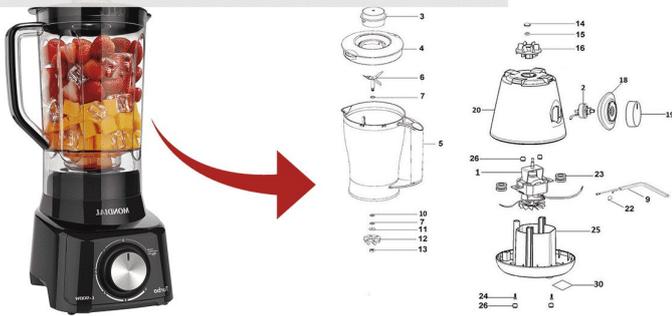
Escreva aqui os valores presumidos para as 4 variáveis "A" de cada um dos 4 algoritmos

1	2	3	4

75903

Engenharia Reversa

ENGENHARIA REVERSA



Engenharia Reversa de algoritmos

Se a programação fosse mais engenharia e menos arte, talvez não precisássemos tanto usar os dons de Sherlock Holmes para corrigir programas. Infelizmente (ou será felizmente ??) tal não ocorre, e um programador profissional gasta mais de 50% do seu tempo procurando e corrigindo erros em programas.

É tarefa difícil, sutil, complexa e delicada. s vezes um ponto fora do lugar põe a perder um imenso projeto. É tão importante a tarefa que se diz que bom programador não é aquele de programa bem, mas aquele que depura bem. A propósito, depurar é **identificar, localizar e corrigir erros, de maneira continuada e integrada para produzir software de qualidade.**

O exercício de hoje, visa estimular esta habilidade. Você vai receber alguns códigos e alguns resultados, e deverá inferir quais as inicializações originais das variáveis em cada código.

Por exemplo, no algoritmo

```
1: função EXEMPLO
2: A ← _____
3: B ← 2
4: C ← 3
5: enquanto A < 10
6:   se (B mod 3) = 0
7:     A ← A + 1
8:   senão
9:     A ← A + 2
10:  fim{se}
11:  se (C mod 4) = 0
12:    B ← B + 2
13:  senão
14:    A ← A + 1
15:  fim{se}
16:  imprima A, B, C
17: fim{enquanto}
```

Analisando as saídas, que são:

4 2 3
7 2 3
10 2 3

Percebe-se que a inicialização original do A foi 1. A resposta deste exemplo portanto, é 1.

Para você fazer

A seguir, diversos algoritmos. Em cada um deles, você deve dizer qual o valor inicial da variável A.

Exercício 1

```
1: função A1
2: A ← _____
3: B ← 6
4: C ← 14
5: enquanto (A + B) > 4
6:   se A > C
7:     A ← A - 1
8:   senão
9:     A ← A + 1
```

```
10:  fim{se}
11:  B ← B - 2
12:  imprima A, B, C
13: fim{enquanto}
```

Esta execução deu como resultado

7 4 14
8 2 14
9 0 14
10 -2 14
11 -4 14
12 -6 14
13 -8 14
14 -10 14

Exercício 2

```
1: função A2
2: A ← _____
3: B ← 15
4: C ← 10
5: enquanto (C ≠ 0) ∧ (C ≠ 1)
6:   se (B mod 5) ≠ 0
7:     B ← B - 1
8:   fim{se}
9:   se (A mod 3) = 0
10:    A ← A + 1
11:  fim{se}
12:  se (A > B)
13:    A ← A + 2
14:  fim{se}
15:  C ← C - 1
16:  imprima A, B, C
17: fim{enquanto}
```

Esta execução deu como resultado

11 15 9
11 15 8
11 15 7
11 15 6
11 15 5
11 15 4
11 15 3
11 15 2
11 15 1

Exercício 3

```
1: função A3
2: A ← _____
3: B ← 1
4: C ← 3
5: repita
6:   A ← A + 1
7:   B ← B + 1
8:   C ← C + 1
9: enquanto C > 10
10:  C ← C - 2
11: fim{enquanto}
12: se (C ≠ 3) ∨ (B ≠ 2)
13:  A ← A + 2
14: fim{se}
15: imprima A, B, C
16: até (A + B + C) > 30
```

Esta execução deu como resultado

8 2 4
11 3 5

14 4 6
17 5 7
20 6 8

Exercício 4

```
1: função A4
2: A ← _____
3: B ← 15
4: C ← 11
5: enquanto (A > 10) ∨ (B > 10) ∨ (C > 10)
6:   se A > B
7:     C ← C - 1
8:   fim{se}
9:   se (B mod 3) = 1
10:    A ← A - 1
11:  fim{se}
12:  se C > B
13:    B ← B - 3
14:  fim{se}
15:  imprima A, B, C
16:  A ← A - 1
17:  B ← B - 1
18:  C ← C - 2
19: fim{enquanto}
```

Esta execução deu como resultado

11 15 11
10 14 9
8 13 7
7 12 5
6 11 3

Respostas

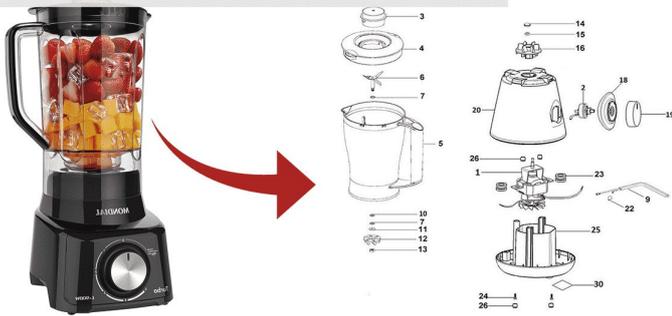
Escreva aqui os valores presumidos para as 4 variáveis "A" de cada um dos 4 algoritmos

1	2	3	4

75910

Engenharia Reversa

ENGENHARIA REVERSA



Engenharia Reversa de algoritmos

Se a programação fosse mais engenharia e menos arte, talvez não precisássemos tanto usar os dons de Sherlock Holmes para corrigir programas. Infelizmente (ou será felizmente ??) tal não ocorre, e um programador profissional gasta mais de 50% do seu tempo procurando e corrigindo erros em programas.

É tarefa difícil, sutil, complexa e delicada. s vezes um ponto fora do lugar põe a perder um imenso projeto. É tão importante a tarefa que se diz que bom programador não é aquele de programa bem, mas aquele que depura bem. A propósito, depurar é **identificar, localizar e corrigir erros, de maneira continuada e integrada para produzir software de qualidade.**

O exercício de hoje, visa estimular esta habilidade. Você vai receber alguns códigos e alguns resultados, e deverá inferir quais as inicializações originais das variáveis em cada código.

Por exemplo, no algoritmo

```

1: função EXEMPLO
2: A ← _____
3: B ← 2
4: C ← 3
5: enquanto A < 10
6:   se (B mod 3) = 0
7:     A ← A + 1
8:   senão
9:     A ← A + 2
10:  fim{se}
11:  se (C mod 4) = 0
12:    B ← B + 2
13:  senão
14:    A ← A + 1
15:  fim{se}
16:  imprima A, B, C
17: fim{enquanto}
    
```

Analisando as saídas, que são:

4 2 3
 7 2 3
 10 2 3

Percebe-se que a inicialização original do A foi 1. A resposta deste exemplo portanto, é 1.

Para você fazer

A seguir, diversos algoritmos. Em cada um deles, você deve dizer qual o valor inicial da variável A.

Exercício 1

```

1: função A1
2: A ← _____
3: B ← 8
4: C ← 11
5: enquanto (A + B) > 4
6:   se A > C
7:     A ← A - 1
8:   senão
9:     A ← A + 1
    
```

```

10:  fim{se}
11:  B ← B - 2
12:  imprima A, B, C
13: fim{enquanto}
    
```

Esta execução deu como resultado

9 6 11
 10 4 11
 11 2 11
 12 0 11
 11 -2 11
 12 -4 11
 11 -6 11
 12 -8 11

Exercício 2

```

1: função A2
2: A ← _____
3: B ← 10
4: C ← 11
5: enquanto (C ≠ 0) ∧ (C ≠ 1)
6:   se (B mod 5) ≠ 0
7:     B ← B - 1
8:   fim{se}
9:   se (A mod 3) = 0
10:    A ← A + 1
11:  fim{se}
12:  se (A > B)
13:    A ← A + 2
14:  fim{se}
15:  C ← C - 1
16:  imprima A, B, C
17: fim{enquanto}
    
```

Esta execução deu como resultado

16 10 10
 18 10 9
 21 10 8
 24 10 7
 27 10 6
 30 10 5
 33 10 4
 36 10 3
 39 10 2
 42 10 1

Exercício 3

```

1: função A3
2: A ← _____
3: B ← 2
4: C ← 2
5: repita
6:   A ← A + 1
7:   B ← B + 1
8:   C ← C + 1
9: enquanto C > 10
10:  C ← C - 2
11: fim{enquanto}
12: se (C ≠ 3) ∨ (B ≠ 2)
13:  A ← A + 2
14: fim{se}
15: imprima A, B, C
16: até (A + B + C) > 30
    
```

Esta execução deu como resultado

4 3 3

7 4 4
 10 5 5
 13 6 6
 16 7 7
 19 8 8

Exercício 4

```

1: função A4
2: A ← _____
3: B ← 13
4: C ← 11
5: enquanto (A > 10) ∨ (B > 10) ∨ (C > 10)
6:   se A > B
7:     C ← C - 1
8:   fim{se}
9:   se (B mod 3) = 1
10:    A ← A - 1
11:  fim{se}
12:  se C > B
13:    B ← B - 3
14:  fim{se}
15:  imprima A, B, C
16:  A ← A - 1
17:  B ← B - 1
18:  C ← C - 2
19: fim{enquanto}
    
```

Esta execução deu como resultado

14 13 10
 13 12 7
 12 11 4
 10 10 1

Respostas

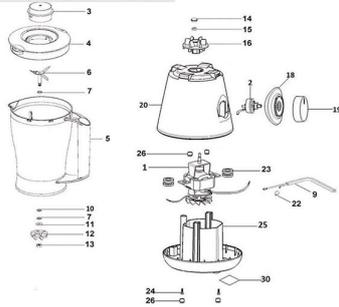
Escreva aqui os valores presumidos para as 4 variáveis "A" de cada um dos 4 algoritmos

1	2	3	4

75927

Engenharia Reversa

ENGENHARIA REVERSA



Engenharia Reversa de algoritmos

Se a programação fosse mais engenharia e menos arte, talvez não precisássemos tanto usar os dons de Sherlock Holmes para corrigir programas. Infelizmente (ou será felizmente ??) tal não ocorre, e um programador profissional gasta mais de 50% do seu tempo procurando e corrigindo erros em programas.

É tarefa difícil, sutil, complexa e delicada. Às vezes um ponto fora do lugar põe a perder um imenso projeto. É tão importante a tarefa que se diz que bom programador não é aquele de programa bem, mas aquele que depura bem. A propósito, depurar é **identificar, localizar e corrigir erros, de maneira continuada e integrada para produzir software de qualidade.**

O exercício de hoje, visa estimular esta habilidade. Você vai receber alguns códigos e alguns resultados, e deverá inferir quais as inicializações originais das variáveis em cada código.

Por exemplo, no algoritmo

```

1: função EXEMPLO
2: A ← _____
3: B ← 2
4: C ← 3
5: enquanto A < 10
6:   se (B mod 3) = 0
7:     A ← A + 1
8:   senão
9:     A ← A + 2
10:  fim{se}
11:  se (C mod 4) = 0
12:    B ← B + 2
13:  senão
14:    A ← A + 1
15:  fim{se}
16:  imprima A, B, C
17: fim{enquanto}
    
```

Analisando as saídas, que são:

4 2 3
 7 2 3
 10 2 3

Percebe-se que a inicialização original do A foi 1. A resposta deste exemplo portanto, é 1.

Para você fazer

A seguir, diversos algoritmos. Em cada um deles, você deve dizer qual o valor inicial da variável A.

Exercício 1

```

1: função A1
2: A ← _____
3: B ← 7
4: C ← 9
5: enquanto (A + B) > 4
6:   se A > C
7:     A ← A - 1
8:   senão
9:     A ← A + 1
    
```

```

10:  fim{se}
11:  B ← B - 2
12:  imprima A, B, C
13: fim{enquanto}

Esta execução deu como resultado
6 5 9
7 3 9
8 1 9
9 -1 9
10 -3 9
9 -5 9
    
```

Exercício 2

```

1: função A2
2: A ← _____
3: B ← 13
4: C ← 10
5: enquanto (C ≠ 0) ∧ (C ≠ 1)
6:   se (B mod 5) ≠ 0
7:     B ← B - 1
8:   fim{se}
9:   se (A mod 3) = 0
10:    A ← A + 1
11:  fim{se}
12:  se (A > B)
13:    A ← A + 2
14:  fim{se}
15:  C ← C - 1
16:  imprima A, B, C
17: fim{enquanto}
    
```

Esta execução deu como resultado

16 12 9
 18 11 8
 21 10 7
 24 10 6
 27 10 5
 30 10 4
 33 10 3
 36 10 2
 39 10 1

Exercício 3

```

1: função A3
2: A ← _____
3: B ← 5
4: C ← 4
5: repita
6:   A ← A + 1
7:   B ← B + 1
8:   C ← C + 1
9:   enquanto C > 10
10:    C ← C - 2
11:  fim{enquanto}
12:  se (C ≠ 3) ∨ (B ≠ 2)
13:    A ← A + 2
14:  fim{se}
15:  imprima A, B, C
16: até (A + B + C) > 30
    
```

Esta execução deu como resultado

4 6 5
 7 7 6
 10 8 7
 13 9 8

16 10 9

Exercício 4

```

1: função A4
2: A ← _____
3: B ← 14
4: C ← 14
5: enquanto (A > 10) ∨ (B > 10) ∨ (C > 10)
6:   se A > B
7:     C ← C - 1
8:   fim{se}
9:   se (B mod 3) = 1
10:    A ← A - 1
11:   fim{se}
12:   se C > B
13:     B ← B - 3
14:   fim{se}
15:   imprima A, B, C
16:   A ← A - 1
17:   B ← B - 1
18:   C ← C - 2
19: fim{enquanto}
    
```

Esta execução deu como resultado

14 14 14
 12 13 12
 11 12 10
 10 11 8

Respostas

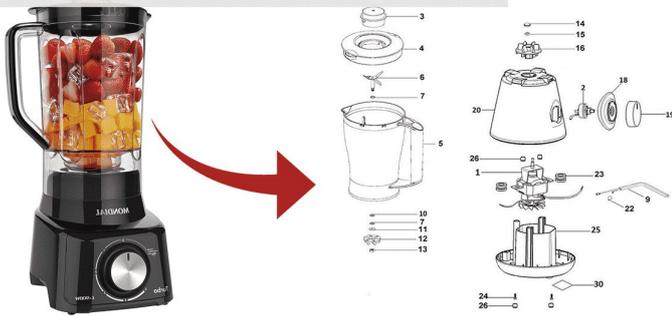
Escreva aqui os valores presumidos para as 4 variáveis "A" de cada um dos 4 algoritmos

1	2	3	4

75934

Engenharia Reversa

ENGENHARIA REVERSA



Engenharia Reversa de algoritmos

Se a programação fosse mais engenharia e menos arte, talvez não precisássemos tanto usar os dons de Sherlock Holmes para corrigir programas. Infelizmente (ou será felizmente ??) tal não ocorre, e um programador profissional gasta mais de 50% do seu tempo procurando e corrigindo erros em programas.

É tarefa difícil, sutil, complexa e delicada. s vezes um ponto fora do lugar põe a perder um imenso projeto. É tão importante a tarefa que se diz que bom programador não é aquele de programa bem, mas aquele que depura bem. A propósito, depurar é **identificar, localizar e corrigir erros, de maneira continuada e integrada para produzir software de qualidade.**

O exercício de hoje, visa estimular esta habilidade. Você vai receber alguns códigos e alguns resultados, e deverá inferir quais as inicializações originais das variáveis em cada código.

Por exemplo, no algoritmo

```

1: função EXEMPLO
2: A ← _____
3: B ← 2
4: C ← 3
5: enquanto A < 10
6:   se (B mod 3) = 0
7:     A ← A + 1
8:   senão
9:     A ← A + 2
10:  fim{se}
11:  se (C mod 4) = 0
12:    B ← B + 2
13:  senão
14:    A ← A + 1
15:  fim{se}
16:  imprima A, B, C
17: fim{enquanto}
    
```

Analisando as saídas, que são:

4 2 3
 7 2 3
 10 2 3

Percebe-se que a inicialização original do A foi 1. A resposta deste exemplo portanto, é 1.

Para você fazer

A seguir, diversos algoritmos. Em cada um deles, você deve dizer qual o valor inicial da variável A.

Exercício 1

```

1: função A1
2: A ← _____
3: B ← 7
4: C ← 9
5: enquanto (A + B) > 4
6:   se A > C
7:     A ← A - 1
8:   senão
9:     A ← A + 1
    
```

```

10:  fim{se}
11:  B ← B - 2
12:  imprima A, B, C
13: fim{enquanto}

Esta execução deu como resultado
8 5 9
9 3 9
10 1 9
9 -1 9
10 -3 9
9 -5 9
    
```

Exercício 2

```

1: função A2
2: A ← _____
3: B ← 9
4: C ← 10
5: enquanto (C ≠ 0) ∧ (C ≠ 1)
6:   se (B mod 5) ≠ 0
7:     B ← B - 1
8:   fim{se}
9:   se (A mod 3) = 0
10:    A ← A + 1
11:  fim{se}
12:  se (A > B)
13:    A ← A + 2
14:  fim{se}
15:  C ← C - 1
16:  imprima A, B, C
17: fim{enquanto}
    
```

Esta execução deu como resultado

```

13 8 9
15 7 8
18 6 7
21 5 6
24 5 5
27 5 4
30 5 3
33 5 2
36 5 1
    
```

Exercício 3

```

1: função A3
2: A ← _____
3: B ← 5
4: C ← 1
5: repita
6:   A ← A + 1
7:   B ← B + 1
8:   C ← C + 1
9: enquanto C > 10
10:  C ← C - 2
11: fim{enquanto}
12: se (C ≠ 3) ∨ (B ≠ 2)
13:  A ← A + 2
14: fim{se}
15: imprima A, B, C
16: até (A + B + C) > 30
    
```

Esta execução deu como resultado

```

8 6 2
11 7 3
14 8 4
17 9 5
    
```

Exercício 4

```

1: função A4
2: A ← _____
3: B ← 11
4: C ← 15
5: enquanto (A > 10) ∨ (B > 10) ∨ (C > 10)
6:   se A > B
7:     C ← C - 1
8:   fim{se}
9:   se (B mod 3) = 1
10:    A ← A - 1
11:  fim{se}
12:  se C > B
13:    B ← B - 3
14:  fim{se}
15:  imprima A, B, C
16:  A ← A - 1
17:  B ← B - 1
18:  C ← C - 2
19: fim{enquanto}
    
```

Esta execução deu como resultado

```

13 8 14
11 4 11
    
```

Respostas

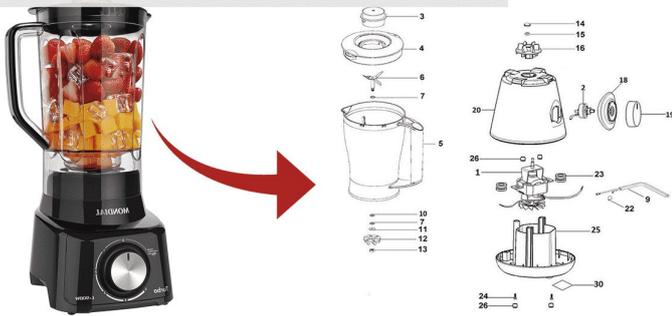
Escreva aqui os valores presumidos para as 4 variáveis "A" de cada um dos 4 algoritmos

1	2	3	4

75941

Engenharia Reversa

ENGENHARIA REVERSA



Engenharia Reversa de algoritmos

Se a programação fosse mais engenharia e menos arte, talvez não precisássemos tanto usar os dons de Sherlock Holmes para corrigir programas. Infelizmente (ou será felizmente ??) tal não ocorre, e um programador profissional gasta mais de 50% do seu tempo procurando e corrigindo erros em programas.

É tarefa difícil, sutil, complexa e delicada. s vezes um ponto fora do lugar põe a perder um imenso projeto. É tão importante a tarefa que se diz que bom programador não é aquele de programa bem, mas aquele que depura bem. A propósito, depurar é **identificar, localizar e corrigir erros, de maneira continuada e integrada para produzir software de qualidade.**

O exercício de hoje, visa estimular esta habilidade. Você vai receber alguns códigos e alguns resultados, e deverá inferir quais as inicializações originais das variáveis em cada código.

Por exemplo, no algoritmo

```

1: função EXEMPLO
2: A ← _____
3: B ← 2
4: C ← 3
5: enquanto A < 10
6:   se (B mod 3) = 0
7:     A ← A + 1
8:   senão
9:     A ← A + 2
10:  fim{se}
11:  se (C mod 4) = 0
12:    B ← B + 2
13:  senão
14:    A ← A + 1
15:  fim{se}
16:  imprima A, B, C
17: fim{enquanto}
    
```

Analisando as saídas, que são:

4 2 3
 7 2 3
 10 2 3

Percebe-se que a inicialização original do A foi 1. A resposta deste exemplo portanto, é 1.

Para você fazer

A seguir, diversos algoritmos. Em cada um deles, você deve dizer qual o valor inicial da variável A.

Exercício 1

```

1: função A1
2: A ← _____
3: B ← 8
4: C ← 10
5: enquanto (A + B) > 4
6:   se A > C
7:     A ← A - 1
8:   senão
9:     A ← A + 1
    
```

```

10:  fim{se}
11:  B ← B - 2
12:  imprima A, B, C
13: fim{enquanto}
    
```

Esta execução deu como resultado

8 6 10
 9 4 10
 10 2 10
 11 0 10
 10 -2 10
 11 -4 10
 10 -6 10

Exercício 2

```

1: função A2
2: A ← _____
3: B ← 14
4: C ← 10
5: enquanto (C ≠ 0) ∧ (C ≠ 1)
6:   se (B mod 5) ≠ 0
7:     B ← B - 1
8:   fim{se}
9:   se (A mod 3) = 0
10:    A ← A + 1
11:   fim{se}
12:   se (A > B)
13:     A ← A + 2
14:   fim{se}
15:   C ← C - 1
16:   imprima A, B, C
17: fim{enquanto}
    
```

Esta execução deu como resultado

16 13 9
 18 12 8
 21 11 7
 24 10 6
 27 10 5
 30 10 4
 33 10 3
 36 10 2
 39 10 1

Exercício 3

```

1: função A3
2: A ← _____
3: B ← 3
4: C ← 2
5: repita
6:   A ← A + 1
7:   B ← B + 1
8:   C ← C + 1
9: enquanto C > 10
10:  C ← C - 2
11: fim{enquanto}
12: se (C ≠ 3) ∨ (B ≠ 2)
13:  A ← A + 2
14: fim{se}
15: imprima A, B, C
16: até (A + B + C) > 30
    
```

Esta execução deu como resultado

4 4 3
 7 5 4
 10 6 5

13 7 6
 16 8 7

Exercício 4

```

1: função A4
2: A ← _____
3: B ← 14
4: C ← 11
5: enquanto (A > 10) ∨ (B > 10) ∨ (C > 10)
6:   se A > B
7:     C ← C - 1
8:   fim{se}
9:   se (B mod 3) = 1
10:    A ← A - 1
11:   fim{se}
12:   se C > B
13:     B ← B - 3
14:   fim{se}
15:   imprima A, B, C
16:   A ← A - 1
17:   B ← B - 1
18:   C ← C - 2
19: fim{enquanto}
    
```

Esta execução deu como resultado

14 14 11
 12 13 9
 11 12 7
 10 11 5

Respostas

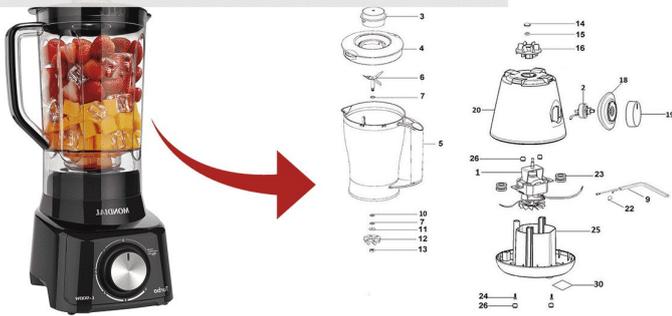
Escreva aqui os valores presumidos para as 4 variáveis "A" de cada um dos 4 algoritmos

1	2	3	4

75958

Engenharia Reversa

ENGENHARIA REVERSA



Engenharia Reversa de algoritmos

Se a programação fosse mais engenharia e menos arte, talvez não precisássemos tanto usar os dons de Sherlock Holmes para corrigir programas. Infelizmente (ou será felizmente ??) tal não ocorre, e um programador profissional gasta mais de 50% do seu tempo procurando e corrigindo erros em programas.

É tarefa difícil, sutil, complexa e delicada. s vezes um ponto fora do lugar põe a perder um imenso projeto. É tão importante a tarefa que se diz que bom programador não é aquele de programa bem, mas aquele que depura bem. A propósito, depurar é **identificar, localizar e corrigir erros, de maneira continuada e integrada para produzir software de qualidade.**

O exercício de hoje, visa estimular esta habilidade. Você vai receber alguns códigos e alguns resultados, e deverá inferir quais as inicializações originais das variáveis em cada código.

Por exemplo, no algoritmo

```

1: função EXEMPLO
2: A ← _____
3: B ← 2
4: C ← 3
5: enquanto A < 10
6:   se (B mod 3) = 0
7:     A ← A + 1
8:   senão
9:     A ← A + 2
10:  fim{se}
11:  se (C mod 4) = 0
12:    B ← B + 2
13:  senão
14:    A ← A + 1
15:  fim{se}
16:  imprima A, B, C
17: fim{enquanto}
    
```

Analisando as saídas, que são:

4 2 3
 7 2 3
 10 2 3

Percebe-se que a inicialização original do A foi 1. A resposta deste exemplo portanto, é 1.

Para você fazer

A seguir, diversos algoritmos. Em cada um deles, você deve dizer qual o valor inicial da variável A.

Exercício 1

```

1: função A1
2: A ← _____
3: B ← 7
4: C ← 13
5: enquanto (A + B) > 4
6:   se A > C
7:     A ← A - 1
8:   senão
9:     A ← A + 1
    
```

```

10:  fim{se}
11:  B ← B - 2
12:  imprima A, B, C
13: fim{enquanto}
    
```

Esta execução deu como resultado

6 5 13
 7 3 13
 8 1 13
 9 -1 13
 10 -3 13
 11 -5 13
 12 -7 13
 13 -9 13

Exercício 2

```

1: função A2
2: A ← _____
3: B ← 10
4: C ← 11
5: enquanto (C ≠ 0) ∧ (C ≠ 1)
6:   se (B mod 5) ≠ 0
7:     B ← B - 1
8:   fim{se}
9:   se (A mod 3) = 0
10:    A ← A + 1
11:  fim{se}
12:  se (A > B)
13:    A ← A + 2
14:  fim{se}
15:  C ← C - 1
16:  imprima A, B, C
17: fim{enquanto}
    
```

Esta execução deu como resultado

13 10 10
 15 10 9
 18 10 8
 21 10 7
 24 10 6
 27 10 5
 30 10 4
 33 10 3
 36 10 2
 39 10 1

Exercício 3

```

1: função A3
2: A ← _____
3: B ← 3
4: C ← 5
5: repita
6:   A ← A + 1
7:   B ← B + 1
8:   C ← C + 1
9: enquanto C > 10
10:  C ← C - 2
11: fim{enquanto}
12: se (C ≠ 3) ∨ (B ≠ 2)
13:  A ← A + 2
14: fim{se}
15: imprima A, B, C
16: até (A + B + C) > 30
    
```

Esta execução deu como resultado

8 4 6

11 5 7
 14 6 8
 17 7 9

Exercício 4

```

1: função A4
2: A ← _____
3: B ← 15
4: C ← 14
5: enquanto (A > 10) ∨ (B > 10) ∨ (C > 10)
6:   se A > B
7:     C ← C - 1
8:   fim{se}
9:   se (B mod 3) = 1
10:    A ← A - 1
11:  fim{se}
12:  se C > B
13:    B ← B - 3
14:  fim{se}
15:  imprima A, B, C
16:  A ← A - 1
17:  B ← B - 1
18:  C ← C - 2
19: fim{enquanto}
    
```

Esta execução deu como resultado

14 15 14
 13 14 12
 11 13 10
 10 12 8
 9 11 6

Respostas

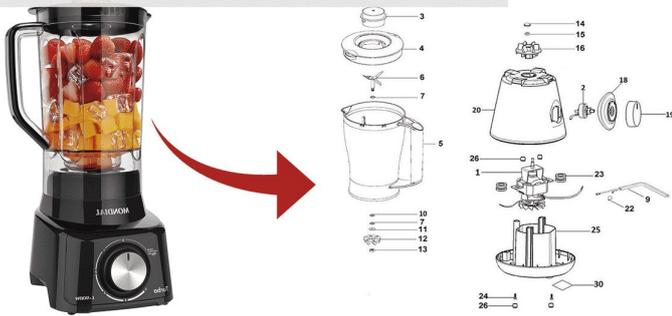
Escreva aqui os valores presumidos para as 4 variáveis "A" de cada um dos 4 algoritmos

1	2	3	4

75965

Engenharia Reversa

ENGENHARIA REVERSA



Engenharia Reversa de algoritmos

Se a programação fosse mais engenharia e menos arte, talvez não precisássemos tanto usar os dons de Sherlock Holmes para corrigir programas. Infelizmente (ou será felizmente ??) tal não ocorre, e um programador profissional gasta mais de 50% do seu tempo procurando e corrigindo erros em programas.

É tarefa difícil, sutil, complexa e delicada. s vezes um ponto fora do lugar põe a perder um imenso projeto. É tão importante a tarefa que se diz que bom programador não é aquele de programa bem, mas aquele que depura bem. A propósito, depurar é **identificar, localizar e corrigir erros, de maneira continuada e integrada para produzir software de qualidade.**

O exercício de hoje, visa estimular esta habilidade. Você vai receber alguns códigos e alguns resultados, e deverá inferir quais as inicializações originais das variáveis em cada código.

Por exemplo, no algoritmo

```

1: função EXEMPLO
2: A ← _____
3: B ← 2
4: C ← 3
5: enquanto A < 10
6:   se (B mod 3) = 0
7:     A ← A + 1
8:   senão
9:     A ← A + 2
10:  fim{se}
11:  se (C mod 4) = 0
12:    B ← B + 2
13:  senão
14:    A ← A + 1
15:  fim{se}
16:  imprima A, B, C
17: fim{enquanto}
    
```

Analisando as saídas, que são:

4 2 3
 7 2 3
 10 2 3

Percebe-se que a inicialização original do A foi 1. A resposta deste exemplo portanto, é 1.

Para você fazer

A seguir, diversos algoritmos. Em cada um deles, você deve dizer qual o valor inicial da variável A.

Exercício 1

```

1: função A1
2: A ← _____
3: B ← 6
4: C ← 12
5: enquanto (A + B) > 4
6:   se A > C
7:     A ← A - 1
8:   senão
9:     A ← A + 1
    
```

```

10:  fim{se}
11:  B ← B - 2
12:  imprima A, B, C
13: fim{enquanto}
    
```

Esta execução deu como resultado

6 4 12
 7 2 12
 8 0 12
 9 -2 12
 10 -4 12
 11 -6 12
 12 -8 12

Exercício 2

```

1: função A2
2: A ← _____
3: B ← 12
4: C ← 11
5: enquanto (C ≠ 0) ∧ (C ≠ 1)
6:   se (B mod 5) ≠ 0
7:     B ← B - 1
8:   fim{se}
9:   se (A mod 3) = 0
10:    A ← A + 1
11:  fim{se}
12:  se (A > B)
13:    A ← A + 2
14:  fim{se}
15:  C ← C - 1
16:  imprima A, B, C
17: fim{enquanto}
    
```

Esta execução deu como resultado

16 11 10
 18 10 9
 21 10 8
 24 10 7
 27 10 6
 30 10 5
 33 10 4
 36 10 3
 39 10 2
 42 10 1

Exercício 3

```

1: função A3
2: A ← _____
3: B ← 2
4: C ← 2
5: repita
6:   A ← A + 1
7:   B ← B + 1
8:   C ← C + 1
9: enquanto C > 10
10:  C ← C - 2
11: fim{enquanto}
12: se (C ≠ 3) ∨ (B ≠ 2)
13:  A ← A + 2
14: fim{se}
15: imprima A, B, C
16: até (A + B + C) > 30
    
```

Esta execução deu como resultado

8 3 3
 11 4 4

14 5 5
 17 6 6
 20 7 7

Exercício 4

```

1: função A4
2: A ← _____
3: B ← 14
4: C ← 12
5: enquanto (A > 10) ∨ (B > 10) ∨ (C > 10)
6:   se A > B
7:     C ← C - 1
8:   fim{se}
9:   se (B mod 3) = 1
10:    A ← A - 1
11:  fim{se}
12:  se C > B
13:    B ← B - 3
14:  fim{se}
15:  imprima A, B, C
16:  A ← A - 1
17:  B ← B - 1
18:  C ← C - 2
19: fim{enquanto}
    
```

Esta execução deu como resultado

14 14 12
 12 13 10
 11 12 8
 10 11 6

Respostas

Escreva aqui os valores presumidos para as 4 variáveis "A" de cada um dos 4 algoritmos

1	2	3	4

75972