1. Algoritmo é uma sequência finita de passos/comandos que ao fim da execução alcança um resultado.
2. São referencias a alocações de memoria onde informações/dados ficam armazenados, é utilizado para armazenar valores que podem ser utilizados durante o algoritmo.
3. São o tipo da informação que a variável está referenciando, tipos estes que podem ser inteiros, textos, caracteres, decimais etc. São uteis para a organização e interpretação do código pelo compilador e pelo programador.
4. Início

Embarcam 2 canibais no lado A;

Os 2 canibais remam para o lado B;

Desembarca 1 canibal;

Volta 1 canibal remando para o lado A;

Embarca 1 canibal;

Os 2 canibais remam para o lado B

Desembarca 1 canibal;

Volta 1 canibal remando para o lado A;

Desembarca 1 canibal;

Embarca 2 Jesuítas;

Os 2 jesuítas remam para o lado B;

Desembarca 1 jesuíta;

Embarca 1 canibal;

Volta 1 canibal e 1 jesuíta para o lado A;

Desembarca 1 canibal;

Embarca 1 jesuíta;

Os 2 jesuítas remam para o lado B;

Desembarcam 2 jesuítas;

Embarca 1 canibal;

O canibal rema para o lado A;

Embarca 1 canibal;

Os 2 canibais remam para o lado B;

Desembarca 1 canibal;

O canibal rema para o lado A;

Embarca 1 canibal;

Os 2 canibais remam para o lado B;

Os 2 canibais desembarcam;

Fim

1. **a)** -6

**b)** 5

**c)** 44,333...

**d)** 81

**e)** 12

1. **a)** y = 2 + a /b – 3 – 2\* x + pow(x, 3);

**b)** x = b / a + c + 4 \* a / 2 – 2 \* a / 3 + c;

**c)** x = pow(-b, 15) + sqrt(pow(b, 2) – 4 \* a \* c) / 2 \* pow(a, 2) \* sqrt(7 \* c);

**d)** x = pow(b + 1, 2) + c + a / b + 1 / 2 + 2 \* a / pow(b, 2);

1. **a)** y = pow(2 \* sqrt(x + y \* pow(pow(x, 3) / 2, (1/25))), (1/13));

**b)** x = 1 / sqrt(x) + 1 / pow(x, (1/3)) + 1 / pow(x, (1/5));

**c)** mq = sqrt(pow(a, 2) + pow(b, 2) + pow(c, 2) + pow(d, 2) / 4)

**d)** mg = pow(a \* b \* c \* d, (1/4));

1. Início

inteiro q = 10, w = -1, r // q armazena 10, w armazena -1 e r armazena 0

real e // e armazena 0.0

q = 10 + 30 // q armazena 40

w = -1 // w armazena -1

w = w + q // w armazena -39

q = q % w // q armazena 1

q = w / (q + 40) // q armazena 0

e = 2 \* q / w // e armazena 0.0

r = 0 // r armazena 0

r = r + 1 // r armazena 1

r = r + 1 // r armazena 2

Fim

1. **a)** FALSE

**b)** FALSE

**c)** FALSE

**d)** TRUE

**e)** TRUE

**f)** TRUE

**g)** FALSE

**h)** FALSE

**i)** TRUE