

Roteiro

- Acesse o website <https://tokiwen.veigo.dev/> por meio de um navegador gráfico atualizado e aguarde o carregamento da página. Um editor de texto familiar deve aparecer à esquerda.
- Na área para edição de texto à esquerda, insira o texto a seguir, que representa um programa para a determinação da quantidade de etapas necessárias para que um número atinja 1 por meio de aplicações repetidas da função de Collatz https://en.wikipedia.org/wiki/Collatz_conjecture:

```
int number;
read number;

int count = 0;

while(number != 1) {
    if (number % 2 == 0) {
        number /= 2;
    } else {
        number = number * 3 + 1;
    }

    count += 1;
}

write count;
```

- Clique no botão “Compilar”. Se estiver acessando por um dispositivo móvel, ele estará fixo na parte inferior da tela. Caso contrário, ele se encontrará logo abaixo da região para edição de texto.
- Utilizando os botões “Executar statement”, “Executar instrução”, “Desfazer statement” e “Desfazer instrução”, execute interativamente o programa.
- Quando o sistema alertar que o programa está aguardando input, insira um número inteiro no campo “Input” posicionado abaixo da área de edição de texto e clique em “Enviar”.
- Observe, pelos painéis que, em dispositivos móveis, se encontram na parte inferior da página e, nos demais, se encontram na parte direita, como variam os valores na memória e em cada variável. O texto na parte superior indica a linha da próxima instrução que será executada. Caso tenha o conhecimento, observe também as instruções e os valores nos registradores. É possível ler o que cada instrução faz: Em dispositivos móveis, clique na respectiva instrução. Em outros dispositivos, simplesmente repouse o cursor em cima dela.
- Quando estiver satisfeito(a), clique em “Configurações”. Ali é possível substituir as palavras-chave pelo que preferir (porém os termos devem começar com uma letra ou _ e no geral conter somente letras, números e _. Acentos e letras não romanas não são válidos infelizmente).
- Então, apague o código fonte inserido anteriormente e construa o seu próprio. As funcionalidades disponíveis são:
 - `int nome;` Declara uma variável de nome `nome` sem valor inicial. Somente ints (números inteiros) são adequadamente suportados no momento. O termo `int` não é considerado uma palavra-chave e portanto infelizmente não é customizável.
 - `int nome = 10;` Declara uma variável de nome `nome` e define o seu valor inicial como 10.
 - `nome = 10;` Define o valor atual da variável `nome`, que deve ter sido declarada anteriormente.
 - `read nome;` Faz com que o programa aguarde por input e armazene o resultado na variável de nome `nome`.
 - `write 10;` Escreve o valor 10 no campo Output, localizado abaixo da área para edição de texto.

- `if (nome < 10) write x;` Executa `write x`; somente se a condição `nome < 10` for verdadeira.
- `if (nome < 10) write x; else nome += 1;` Faz o mesmo que o anterior, mas, além disso, caso a condição não seja verdadeira, é executado `nome += 1`.
- `while (nome < 10) write x;` Executa `write x`; enquanto a condição `nome < 10` for verdadeira. Note que o exemplo fornecido é um loop infinito.
- `{ int nome = 10; }` Encapsula o bloco em um escopo. Variáveis declaradas dentro de um escopo não são válidas fora dele. Isto serve também para executar múltiplas coisas no corpo de um `if` ou `while`.
- `fim: write x;` Associa a label (ou o rótulo) `fim` ao statement `write x`;
- `goto fim;` Altera o fluxo de execução do programa e faz com que ele passe a executar código a partir da label `fim`.
- São suportadas as operações aritméticas de soma (+), subtração (-), multiplicação (*), divisão (/), resto (%), negação (- unário) e identidade (+ unário).
- São suportadas as comparações menor que (<), maior que (>), menor ou igual a (<=), maior ou igual a (>=), igual a (==) e diferente de (!=).
- São suportados os operadores lógicos e (&&), ou (||) e não (! unário).
- É possível combinar sinais de operações aritméticas binárias e de assignment (=) para modificar o valor de uma variável. Por exemplo `number /= 2;` é o mesmo que `number = number / 2;` e `nome += 1;` é o mesmo que `nome = nome + 1;`.

Sugestões de programas:

Primeiramente, evite loops infinitos.

- Leia dois números que representam os lados de um retângulo. Escreva 1 caso o retângulo seja um quadrado e 0 caso não seja.
- Leia três números. Escreva 1 caso eles possam formar os lados de um triângulo e 0 caso não possam.
- Leia três números e os escreva em ordem crescente. Um substituirá o outro, portanto execute interativamente.
- Leia dois números e escreva a soma dos números inteiros entre eles.
- Leia um número e escreva o respectivo valor na sequência de Fibonacci.
- Leia um número e escreva 1 caso ele seja primo e 0 caso não seja.

Questionário

1. Achas que tu ou teus alunos teriam aprendido a programar melhor se esta ferramenta fosse utilizada de início? Avalie com um número inteiro de -5 (seria muito pior) a +5 (seria muito melhor).
2. O quão útil acreditas que a capacidade de customizar as palavras-chave seria no aprendizado seu ou de seus alunos? Avalie de 0 (irrelevante) a +5 (muito útil).
3. Reconsidere a pergunta anterior imaginando que tu ou teus alunos começaram a aprender a programar enquanto no ensino fundamental.
4. O quão útil acreditas que é a capacidade de avançar e retornar na execução do código? Avalie de 0 (irrelevante) a +5 (muito útil).
5. O quão útil acreditas que é a capacidade de observar os valores na memória e em cada variável em tempo real? Avalie de 0 (irrelevante) a +5 (muito útil).
6. Dirias que, no geral, o sistema funcionou de forma intuitiva, isto é, de acordo com as tuas expectativas? Avalie com um número inteiro de -5 (não) a +5 (sim).