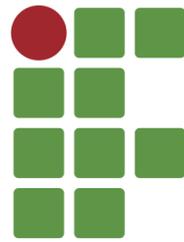


Programação de Computadores 1
Projeto 01: Sudoku
Tecnologia em Automação Industrial

Prof. Daniel Saad Nogueira Nunes



**INSTITUTO
FEDERAL**
Brasília

1 Contexto

O Sudoku é um quebra-cabeça muito comum, presente em diversas revistas e jornais pelo mundo afora. O quebra-cabeça é disposto em um formato de um grid com diversas regiões e é completado quando todas as regiões, linhas e colunas possuem símbolos distintos. O formato de grid mais comum do quebra-cabeça é o 9×9 , mas existem variações de tamanho, como o 16×16 e o 25×25 . Na variante 9×9 , as regiões são subgrids de tamanho 3×3 , e em cada região, linha ou coluna, os números de 1 a 9 só podem ocorrer uma única vez. A Figura 1 ilustra uma solução válida do quebra-cabeça Sudoku, nota-se que esse grid possui 9 regiões disjuntas, de tamanho 3×3 .

Este projeto consiste em determinar se um grid de Sudoku de tamanho 9×9 constitui uma solução válida.

2 Especificação

O programa a ser construído receberá um grid preenchido de tamanho 9×9 e deverá determinar se ele corresponde a uma solução de Sudoku ou não. O projeto deverá estar bem modularizado, dividindo tarefas complexas em tarefas (funções) menores.

2.1 Entrada e Saída

Os dados de entrada deverão ser lidos do teclado `stdin` e os dados de saída deverão ser escritos na tela (`stdout`). É muito importante que o programa siga rigorosamente o formato de entrada e saída. Nada além do que está especificado deve ser impresso em tela. Não é necessário validar a entrada, é garantido que as entradas utilizadas para testar o programa seguem a especificação.

2.2 Entrada

A entrada possui 9 linhas, cada uma com 9 inteiros, que descrevem o grid do quebra-cabeça. O valor que cada inteiro pode assumir está na faixa $[1, 9]$.

2.3 Saída

Imprima uma linha com a mensagem “Sim” se o grid configura uma solução ou “Nao” caso contrário.

2.4 Modularização

O sistema deverá ser dividido em módulos, cada um para cumprir uma tarefa. Estes módulos podem ser organizados internamente através de várias funções e eles correspondem aos seguintes:

- Módulo de leitura: efetua a leitura dos dados de entrada.
- Módulo de saída: efetua a impressão das respostas para cada caso de teste.

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

Figura 1: Solução de um quebra-cabeça 9×9 . Fonte: Wikipedia

- Módulo de processamento: realiza o processamento da entrada para composição da saída.
- Módulo principal: contém a função `main` e as chamadas das funções exportadas pelos outros módulos. O ideal é que este módulo possua uma quantidade muito pequena de código, já que ele vai utilizar funções que estão presentes nos outros módulos.

Os módulos devem ser organizados em arquivos separados, com seus respectivos arquivos de cabeçalho e implementação.

2.5 Construção do sistema

Um `Makefile` deverá ser produzido para a compilação dos códigos-fontes no executável e deverá ser distribuído junto ao código.

2.6 Documentação

O código deve ser bem documentado, com presença de comentários explicando os trechos mais complexos do código. Além disso, um arquivo `README` deve ser providenciado com a devida identificação do autor descrevendo o projeto e instruindo como o código deve ser compilado através da ferramenta `make`.

3 Critérios de correção

Deve ser utilizada a linguagem de programação `C` para a implementação do sistema.

Para validação da correção do algoritmo, testes automatizados serão realizados, então é **crucial** que a saída esteja conforme o especificado.

Serão descontados pontos dos códigos que não possuem indentação.

3.1 Ambiente de Correção

Para a correção dos projetos, será utilizada uma máquina de 64-bits com sistema operacional `GNU/LINUX` e compilador `GCC 10.2.0`, logo é imprescindível que o sistema seja capaz de ser compilado e executado nesta configuração.

4 Considerações

- Este projeto deve ser executado **individualmente**.
- A incidência de plágio acarretará automaticamente nota 0 (**zero**) para os envolvidos. Medidas disciplinares também serão tomadas.
- Trabalhos que não compilarem não serão avaliados.
- O código deve ser entregue em uma pasta zipada com a devida identificação do aluno através da sala de aula virtual da disciplina.