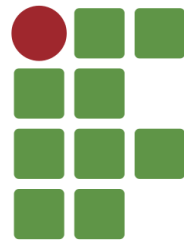


Programação de Computadores 1
Sudoku
Automação Industrial

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 25×25 , as regiões são subgrids de tamanho 5×5 , e em cada região, linha ou coluna, os números de 1 a 25 só podem ocorrer uma única vez. A Figura 1 ilustra uma solução válida do quebra-cabeça Sudoku, nota-se que esse grid possui 25 regiões disjuntas, de tamanho 5×5 .

Este projeto consiste em determinar se um grid de Sudoku de tamanho 25×25 constitui uma solução válida.

2 Especificação

O programa a ser construído receberá um grid preenchido de tamanho 25×25 e deverá determinar se ele corresponde a uma solução de Sudoku ou não. O projeto deverá estar bem modularizado.

2.1 Modularização

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

- Módulo de leitura: efetua a leitura do programa de entrada.
- Módulo de saída: efetua a impressão dos dados.
- Módulo de processamento: responsável pelo processamento das regras de negócio.
- Módulo principal: contém a função main e realiza a chamada aos outros módulos.

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

2.2 Construção do sistema

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

2.3 Documentação

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

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
11	12	13	14	15	1	2	3	4	5	16	17	18	19	20	22	23	21	24	25	6	7	8	9	10
6	7	8	9	10	24	25	22	21	23	1	2	3	4	5	11	12	13	14	15	16	17	18	19	20
23	22	21	25	24	16	17	18	19	20	6	7	8	9	10	1	2	3	4	5	11	12	13	14	15
16	17	18	19	20	11	12	13	14	15	24	21	23	22	25	6	7	8	9	10	1	2	3	4	5
3	1	5	2	14	19	16	20	7	11	18	15	21	24	23	8	10	25	22	12	4	9	17	13	6
4	8	10	6	18	3	1	23	24	2	13	22	25	12	14	9	16	19	11	17	5	15	20	21	7
24	15	25	7	12	4	5	9	13	6	3	1	16	17	2	18	20	14	21	23	8	10	22	11	19
9	19	11	20	21	8	22	25	17	14	4	5	10	7	6	3	1	15	13	2	18	24	16	12	23
13	23	17	22	16	10	15	21	18	12	8	9	20	11	19	4	5	24	7	6	3	1	14	25	2
2	5	1	15	3	14	13	17	25	8	23	16	19	10	18	12	24	11	20	9	7	4	21	6	22
7	16	6	11	4	2	18	1	5	3	12	24	17	8	22	14	21	23	10	13	20	25	19	15	9
12	25	23	21	8	7	4	6	11	9	2	20	1	5	3	17	15	22	18	19	14	13	24	10	16
14	20	19	24	9	22	23	10	16	21	7	4	6	15	13	2	25	1	5	3	12	8	11	17	18
17	10	22	18	13	12	24	15	20	19	14	25	11	21	9	7	4	6	8	16	2	23	1	5	3
19	6	4	1	2	25	21	11	12	24	20	18	9	23	8	15	14	16	17	22	10	5	7	3	13
5	13	9	3	7	17	8	2	1	4	22	19	24	25	11	10	18	20	23	21	15	6	12	16	14
15	18	20	23	17	5	6	7	3	16	10	14	2	1	4	19	13	9	12	24	22	21	25	8	11
22	14	12	16	11	18	10	19	15	13	5	6	7	3	21	25	8	2	1	4	17	20	9	23	24
10	21	24	8	25	20	9	14	23	22	15	13	12	16	17	5	6	7	3	11	19	18	2	1	4
18	3	2	5	1	15	20	4	8	25	17	10	22	13	24	23	11	12	16	14	9	19	6	7	21
25	4	7	13	6	9	3	16	2	1	19	23	14	20	12	21	22	10	15	8	24	11	5	18	17
20	11	15	12	19	23	14	5	6	7	21	3	4	2	1	24	9	17	25	18	13	16	10	22	8
8	9	16	17	22	21	19	24	10	18	25	11	5	6	7	13	3	4	2	1	23	14	15	20	12
21	24	14	10	23	13	11	12	22	17	9	8	15	18	16	20	19	5	6	7	25	3	4	2	1

Figura 1: Solução de um quebra-cabeça 25×25 .

2.4 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.

2.5 Entrada

A entrada possui 25 linhas, cada uma com 25 inteiros, que descrevem o grid do quebra-cabeça. O valor que cada inteiro pode assumir está na faixa $[1, 25]$.

2.6 Saída

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

3 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.