

Trabalho Prático Final

1. Objetivo

O objetivo deste trabalho é implementar um programa em C para jogar Sudoku numa versão simplificada (4x4). O programa deve armazenar diversos problemas de Sudoku e apresentá-los ao utilizador para que este introduza a sua solução.

3	1		
		4	1

2. Requisitos

O programa deve possuir as seguintes funcionalidades:

- Possuir diversos problemas de Sudoku.** Os problemas devem estar armazenados em um ficheiro e devem ser lidos automaticamente para a memória (por exemplo: vetores) ao executar o programa.
- Permitir inserir novos problemas que serão acrescentados ao ficheiro de problemas. Cada problema deve ter um nome (*string*) atribuído pelo utilizador. Não deve ser possível introduzir um nome igual ao de outro problema já existente.
- Permitir alterar o nome de um problema, desde que não exista outro problema com o mesmo nome.
- Apresentar ao utilizador problemas **aleatórios** dentre os problemas existentes. Não deve ser apresentado o mesmo problema por duas vezes seguidas na mesma execução do programa.
- Permitir o preenchimento do jogo de Sudoku.** Caso a jogada seja inválida (ver regras no item 3), deve ser apresentada mensagem de erro ao utilizador. A cada jogada válida introduzida pelo utilizador deve ser apresentada a nova situação do jogo. O jogo é considerado perdido se forem realizadas duas jogadas inválidas.
- Permitir remover um número do Sudoku, desde que não seja um dos números iniciais do problema.
- Permitir abortar um jogo e iniciar um novo problema.
- Efetuar as seguintes consultas:
 - Consulta dos problemas existentes e de quantas vezes cada um foi jogado na execução atual do programa.
 - Consulta do número total de jogos realizados, do número de jogos concluídos com sucesso e da percentagem de sucesso (com uma casa decimal).
- Sair do programa.

Especificações adicionais:

- O programa deve operar com base em **menus de opções**.
- Devem ser apresentadas mensagens apropriadas a cada operação bem ou mal sucedida (exemplos: problema inserido, problema alterado, opção inválida, número inválido, jogada inválida, etc.).
- Durante um jogo somente deve ser permitido inserir/remover números e abortar o jogo.
- Após finalizado um jogo deve ser apresentada opção ao utilizador para iniciar novo jogo ou voltar ao menu principal.
- O programa deve ser capaz de gerir até 100 problemas.

3. Regras e referências:

As seguintes regras devem ser verificadas na validação de uma jogada:

- a) Os números de 1 a 4 devem ocorrer apenas uma vez em cada coluna.
- b) Os números de 1 a 4 devem ocorrer apenas uma vez em cada linha.
- c) Os números de 1 a 4 devem ocorrer apenas uma vez em cada quadrado 2x2.

Referências:

https://pt.wikibooks.org/wiki/Matem%C3%A1tica_divertida/Mini-sudoku
<https://www.sudokuonline.io/pt/criancas/numeros-4-4>

4. Implementação

Recomenda-se que utilize funções e múltiplos ficheiros em C, de forma a melhorar a organização e o entendimento do programa. Inserir comentários ao início de em cada ficheiro (.c e .h) com o nome do autor e a descrição do conteúdo do ficheiro.

O programa deve estar corretamente indentado.

5. Entrega, documentação e apresentação

O trabalho é **individual**. A cópia de programas é considerada fraude e implicará na reprovação à unidade curricular, de acordo com o disposto no item 12 do artigo 25º do Regulamento Académico do 1º Ciclo da ESTGOH. Nestes casos, a penalização é aplicada a todos os alunos envolvidos com a cópia, independentemente de terem copiado ou disponibilizado os programas. Além disto, é também proibida e considerada como fraude a utilização de qualquer programa similar obtido da Internet ou de qualquer outra fonte, inclusive os gerados por sistemas de inteligência artificial.

Devem ser submetidos todos os programas e ficheiros necessários à compilação e execução do programa, com exceção do executável.

Além disto, deve ser submetido um relatório em pdf, contendo, no mínimo, a seguinte informação:

- a) Descrição da estrutura do programa (funções, ficheiros .c, e .h).
- b) Descrição dos ficheiros de armazenamento dos dados e dos formatos utilizados.
- c) Manual de utilizador, com imagens da interface com o utilizador (screenshots).
- d) Descrição das funcionalidades não implementadas, ou parcialmente implementadas, e erros de execução que não tenham sido eliminados.

O trabalho está sujeito a uma apresentação do aluno (defesa) para demonstração do programa e resposta a questões técnicas, em data e hora marcada individualmente para cada aluno. A avaliação final do aluno neste trabalho levará em conta, além do programa desenvolvido, o relatório e a apresentação.