

## Trabalho 02 - N Rainhas com Pilha

stco01 2024s2

Data de entrega: 30/10/2024

Importante:

- **Não** olhem códigos de outros grupos ou da internet. Exceto os que são fornecidos.
- Os trabalhos DEVEM ser feitos com o mesmo grupo do trabalho 01.
- TODOS os membros do grupo devem participar e compreender completamente a implementação.
- Em caso de plágio, fraude ou tentativa de burlar o sistema será aplicado nota 0 na disciplina aos envolvidos.
- Alguns alunos serão solicitados para explicar com detalhes a implementação.
- Passar em todos os testes não é garantia de tirar a nota máxima. Sua nota ainda depende do cumprimento das especificações do trabalho, qualidade do código, clareza dos comentários, boas práticas de programação e entendimento da matéria demonstrada em possível reunião.
- O líder do grupo deverá submeter, até a data de entrega, o seu código na plataforma [runcodes.hokama.com.br](https://runcodes.hokama.com.br).
- É esperado e faz parte do aprendizado vocês terem algumas dificuldades como Problemas com Falha de Segmentação, Loops Infinitos, etc. Então não deixe para o último dia!

Neste trabalho, o seu grupo deverá desenvolver um programa para resolver o problema das  $n$  rainhas. O objetivo é posicionar  $n$  rainhas em um tabuleiro de xadrez de dimensão  $n \times n$  de modo que nenhuma rainha ataque outra. Isso implica que nenhuma delas pode compartilhar a mesma linha, coluna ou diagonal.

O programa deve ser implementado em linguagem C e **obrigatoriamente** deverá usar a estrutura **pilha** (pilha.c e pilha.h fornecida). Ou seja, seu programa não poderá usar a estratégia recursiva.

Seu programa deve ler da entrada padrão do sistema apenas o inteiro  $n$ , *por exemplo*:

```
5
```

seu programa deverá imprimir todas as soluções em ordem lexicográfica, no exemplo de 5, o número de soluções é 10 e você deve imprimir todas. As casas com rainha devem ser representadas por "R" e as casas vazias por "\_", entre cada solução deve haver uma quebra de linha.

```
R _ _ _ _  
_ _ R _ _
```

- - - R  
- R - -  
- - - R -

R - - -  
- - - R -  
- R - - -  
- - - R  
- - R - -

- R - - -  
- - - R -  
R - - - -  
- - R - - -  
- - - - R

- R - - -  
- - - R  
- - R - -  
R - - - -  
- - - R -

- - R - -  
R - - - -  
- - - R -  
- R - - -  
- - - - R

- - R - -  
- - - - R  
- R - - -  
- - - R -  
R - - - -

- - - R -  
R - - - -  
- - R - -  
- - - - R  
- R - - -

- - - R -  
- R - - -  
- - - - R  
- - R - -  
R - - - -

- - - - R  
- R - - -  
- - - R -  
R - - - -

```
-- R --  
-- -- R  
-- R --  
R -- --  
-- -- R --  
- R - - -
```

- Seu programa deve executar no run.codes em menos de 1 segundo.
- Você deve liberar adequadamente a memória alocada.
- Se você não tiver certeza se alguma coisa é permitida ou não no trabalho, não hesite em perguntar ao professor!