Homework 2: Contracts
Two alternatives

1. Code Contracts
   - Project web site
   - Online interface
     - https://riseforfun.com/CodeContracts

2. Viper
   - Project web site
     - https://www.pm.inf.ethz.ch/research/viper.html
   - Use plugin for VS Code or web interface
     - http://viper.ethz.ch/examples/blank-example.html
Task 1

- Implement data structure in C# or Viper
  - Elements: integer type, duplicates allowed
  - Access: using element index or actual value
  - Operations
    - void Add(int val)
    - int Get(int index)
    - int GetHigher(int val)
      - It should return the least element greater than val
    - void Remove(int index)
    - void RemoveAll(int val)
    - void Sort()
    - int FindMin()
    - bool Contains(int val)
    - void Clear()
    - int Size()
Define contracts for all operations provided by your data structure

- Contracts should capture the expected behavior
  - All typical usage patterns supported by the operations
- Try to cover also some important corner cases
  - Example: index out of bounds
Task 3

- Write small test client for the data structure
  - It should exercise typical usage patterns and some important corners cases

- Note for tasks 1+3
  - We will not judge the quality of your code
    - Some prefer and use other languages (Java, C, C++, ...)

Task 4

- Use static checker to verify the implementation of your data structure against the contracts

- Use the runtime checker on your test client [optional, just for Code Contracts]
Task 5

• Document your solution
  ▪ Informally describe what non-trivial properties you specified using Code Contracts or Viper
    • “why you did what you did”
  ▪ Positive experience: what contracts (properties) you were able to successfully verify
  ▪ Negative experience: what are the major observed limitations of Code Contracts or Viper
  ▪ For each reported spurious error (if you get some), try to explain why the particular checker reported the error in your opinion
  ▪ Also discuss missed errors (and possible reasons)
Notes about Viper (alternative 2)

- No runtime checker

- Special task [optional]
  - Compare the verification abilities of VC generator and symbolic execution

- Read the tutorial
  - [http://viper.ethz.ch/tutorial/](http://viper.ethz.ch/tutorial/)
Organization

- Deadline: **17.5.2021**
- Submission
  - E-mail: parizek@d3s.mff.cuni.cz