

ASSIGNMENT #2 AND LAB #2

How to solve problems, understand data flow, and reuse code

Primary Lessons

- Understand the data flowing through a program
- Identify reusable and testable patterns in algorithms
- Practice writing algorithms

Secondary Lessons

- Putting code into shared libraries
- Thinking about testing
- Get comfortable with solutions and projects
- Familiarize with the technique of test driven development
- Learn how to make useful console applications
- How to reduce repetition
- ***How to deal with open-ended problem statements!***

General Advice

- Assess facts
- Formulate questions
- Formulate hypotheses
- Validate
- Observe
- Start simple

How to make all this easier?

- Practice
- Have fun
- Celebrate small wins
- Patience
- Kindness
- Keep it simple

Lab #2 Review

- Create a solution
- 3 Console applications
- 1 Class library
- 1 Unit test project
- See: <https://github.com/cdiggins/cs321/tree/main/code-examples/cs321/Lab2>

Assignment #2 Review

- Fill out the implementation to several algorithms
- See: <https://referencesource.microsoft.com/#mscorlib/system/array.cs>
- And:
<https://referencesource.microsoft.com/#mscorlib/system/collections/generic/list.cs>
- Warning: not very modern implementations!

Foreshadowing

- The final project (and maybe future assignments) will look similar to this!
- One or more applications
- One or more test projects
- One or more shared code libraries

Lab 2: What are the questions?

- How to tackle it:
- What are your questions?
 - Is it a misunderstanding or ambiguity of what is requested?
 - Is it an unclear of how to best tackle the problem?
 - Are you unclear on how to best represent data?

How to answer questions

- Really well worded questions are easily answered or googled

“It doesn’t work”

- What doesn’t work?
- How does it not work?
- What is the compiler saying?

Learn to be specific ... like overly specific

- Computers are unlike people
- They don't understand context
- Compiler messages are very helpful

Kinds of Errors

- Type system errors
- Syntax errors
- Semantic errors
- Typographical errors
- Arithmetic errors
- Misspelling
- Forgotten braces

How to make coding easy

- Practice
- ... then ...
- practice some more

Finished Product

- Three executables
- They should allow you to do
 - Piping
 - Redirection
 - Help

Advice

- Get small success
- What do you know how to do?

Question to ask yourself

- What are the requirements?
- What is unclear?
- What is ambiguous?
- What are the possible interpretations?
- What is the data flow?
- What data structures do I need?
- What algorithms do I need?
- What can I test?

Data Flow

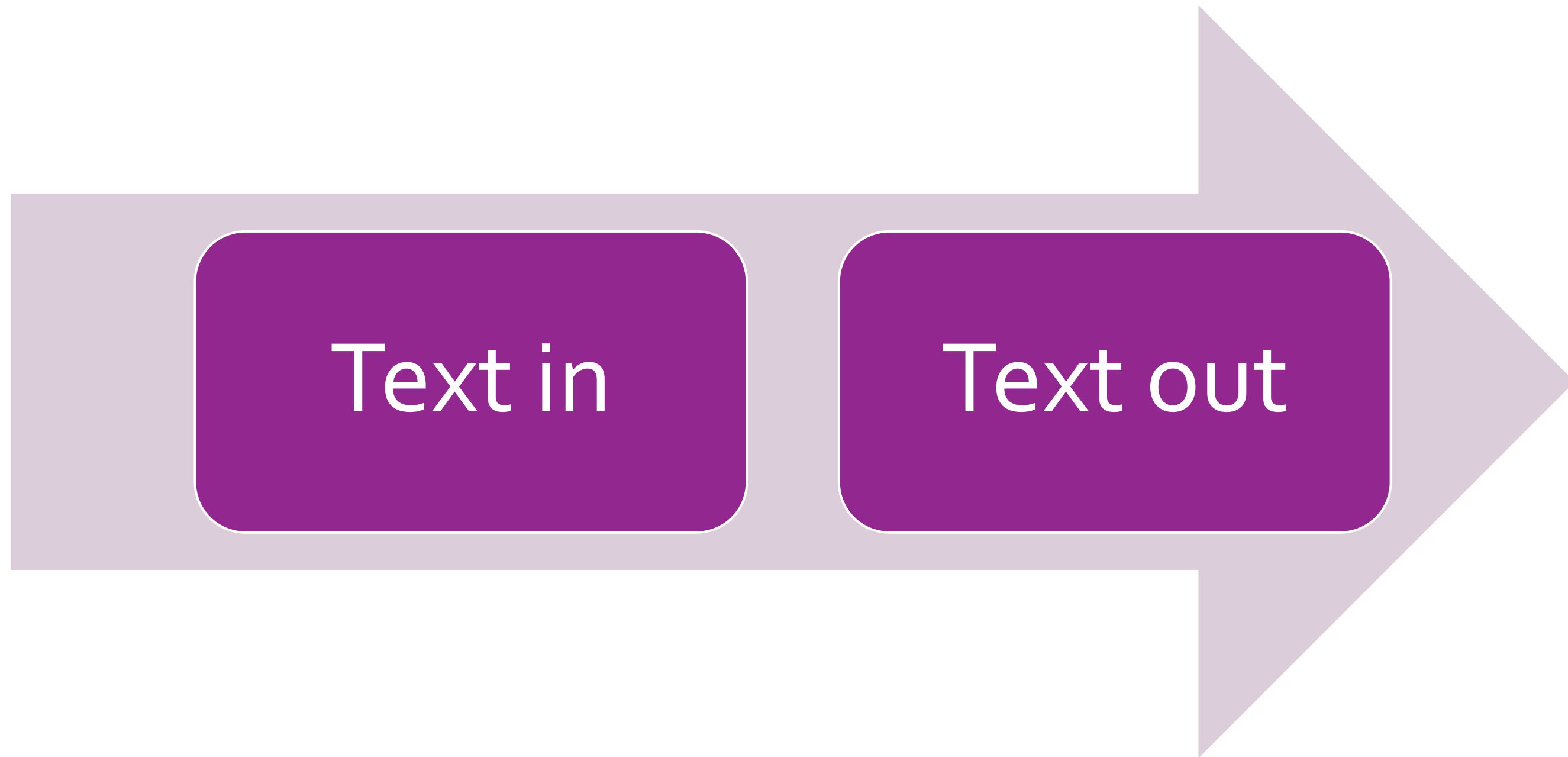
- What data is going into each part of a program?
- How is the data is transformed, filtered, or aggregated?

Transform

- Data is mapped from one domain into another using a function

In the context of the lab

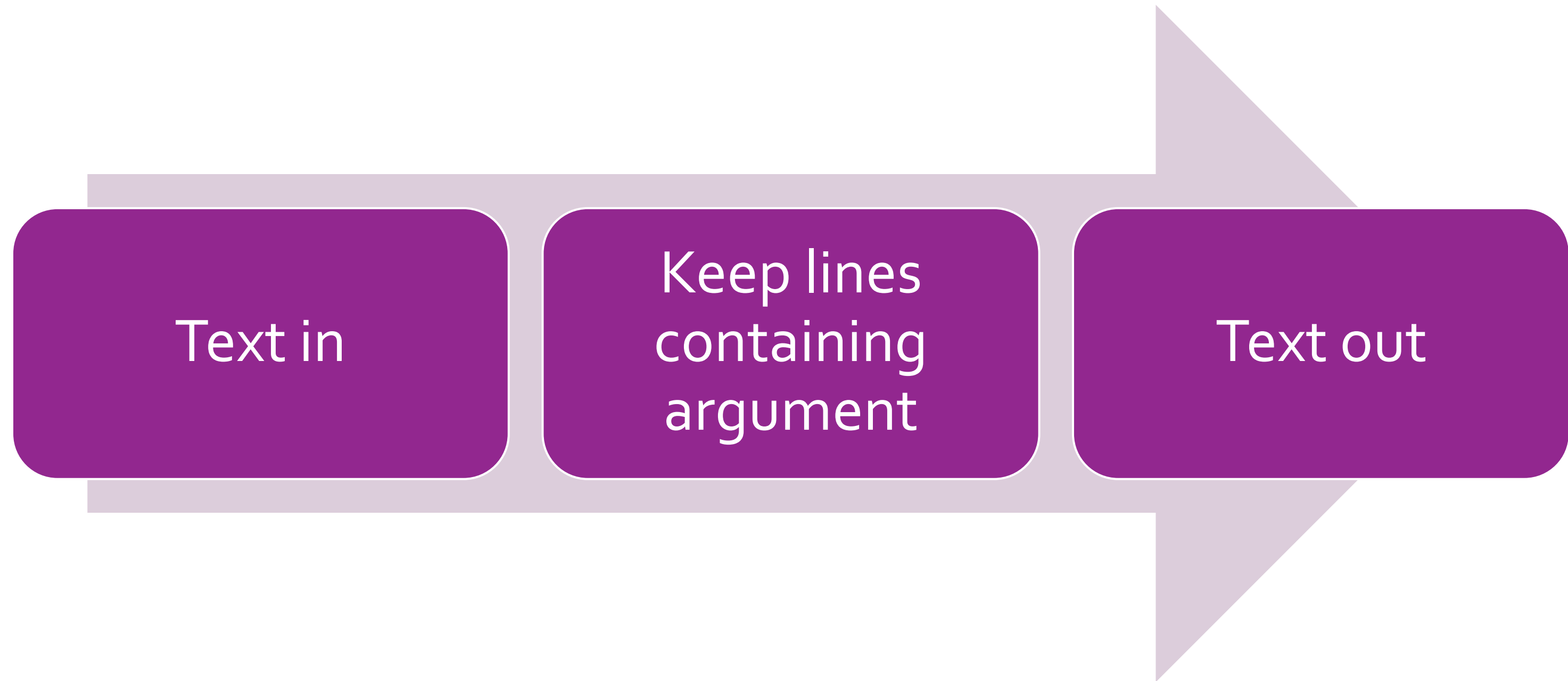
Type – via Standard Input



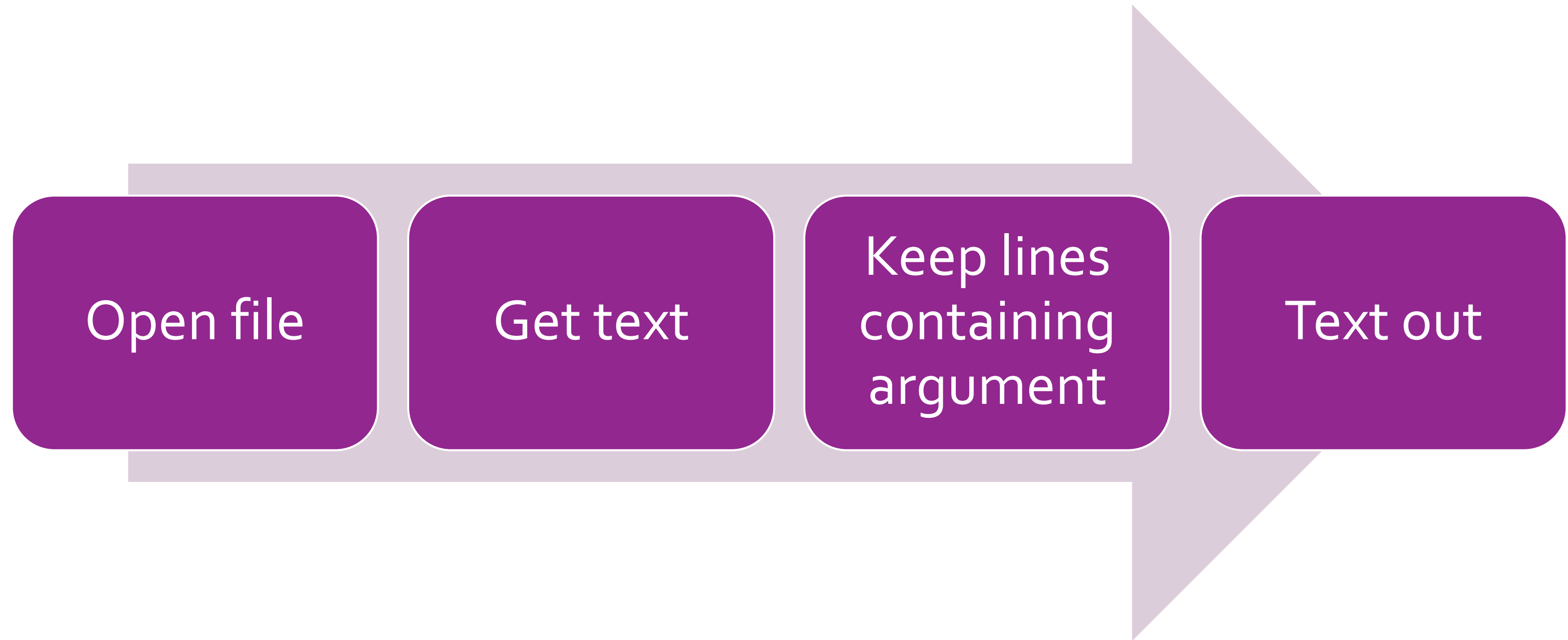
Type – via Command Line Argument



Find – via Standard Input



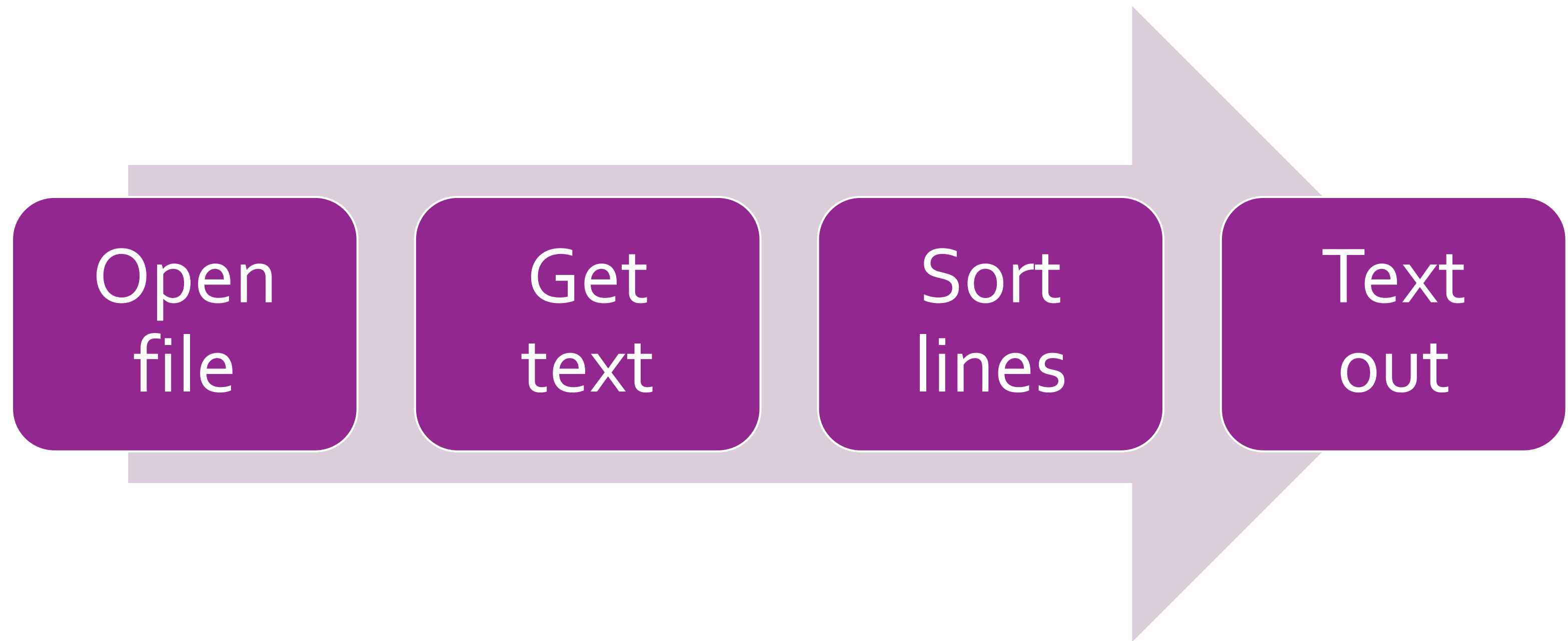
Find – via Command Line Argument



Sort – via Standard Input



Sort – via Command Line Argument



A blue pen is positioned diagonally across the frame, resting on a document. The document features a bar chart with several vertical bars of varying heights. The background is a light blue grid. The entire image is framed by a thick purple border.

WHAT IS THE DATA TYPE?

Describe the data

- Look for keywords in description:
- Text
- Lines

How do we represent it?

- Before we answer ... we need to know what we are doing with it.
- How we process data affects the representation

What do we do to the data?

- Input
- Output
- Filter – “That contains”
- Sort – “In sorted order”

How do we represent it?

- `Array<byte>`
- `List<byte>`
- `IEnumerable<byte>`
- `string`
- `Array<char>`
- `List<char>`
- `IEnumerable<char>`
- `string[]`
- `List<string>`
- `IEnumerable<string>`
- Other ideas?

Patterns

- Data comes from either standard input or a file
- All programs de