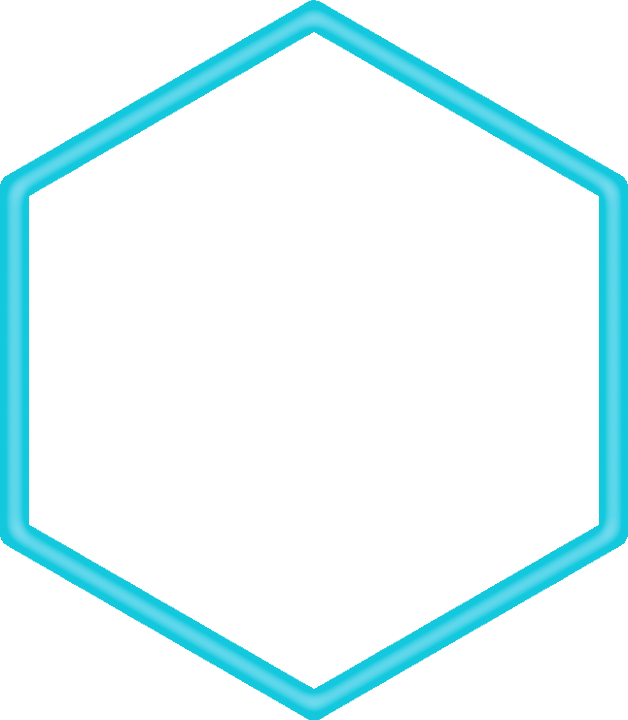
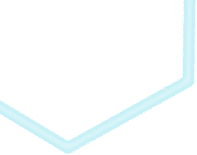
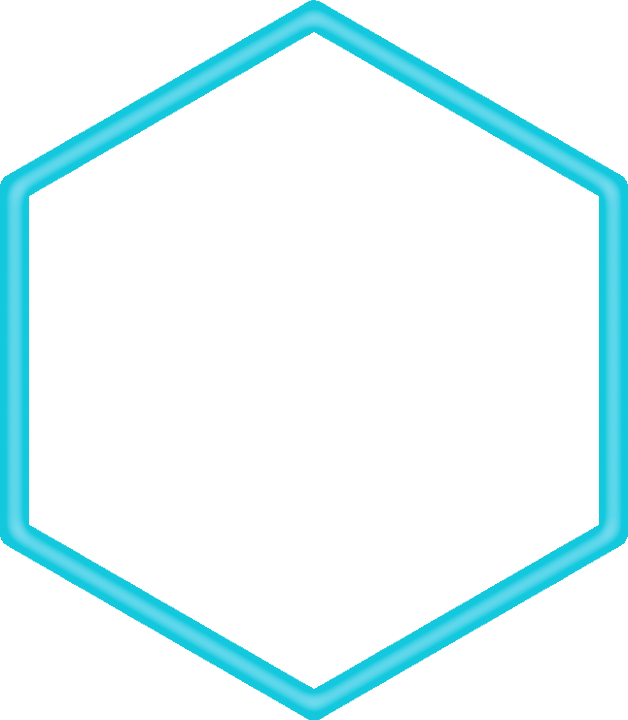
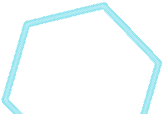
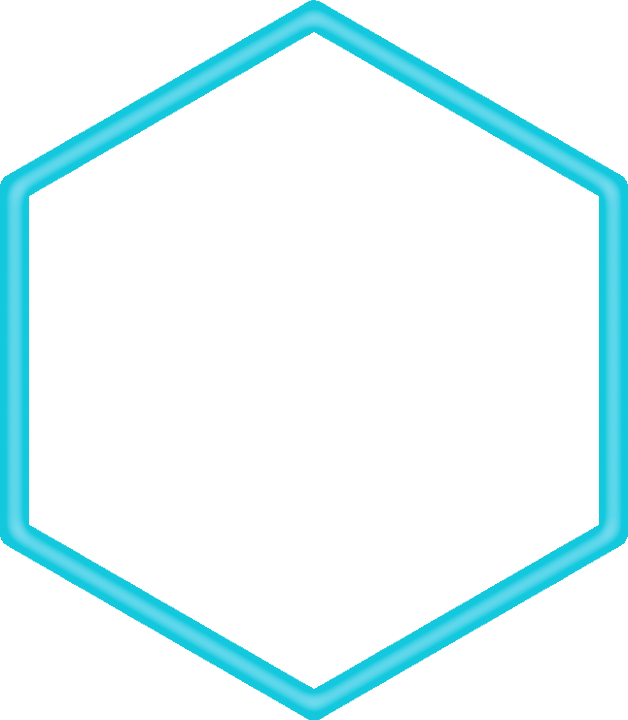
**ONLINE TEST FOR APPLICANTS**

**FactFinder**



For the implementation you can use any mainstream language. If you know any functional-first languages (Haskell, OCaml/F#), you can use those also. You can use whatever IDEs and tools you like to, of course.

Supplied code should be compilable.

Given a string, return the middle character of the string. If the string's length is odd, return the middle character. If the string's length is even, return the middle 2 characters.

**Specification**

**Question 1/3**

**getMiddle("test") should return "es"**

**getMiddle("testing") should return "t"**

**getMiddle("middle") should return "dd"**

**getMiddle("A") should return "A"**

**Examples**

The middle character(s) of the string, represented as another string

**Output**

A string of length 0 < str < 1000.

**Input**

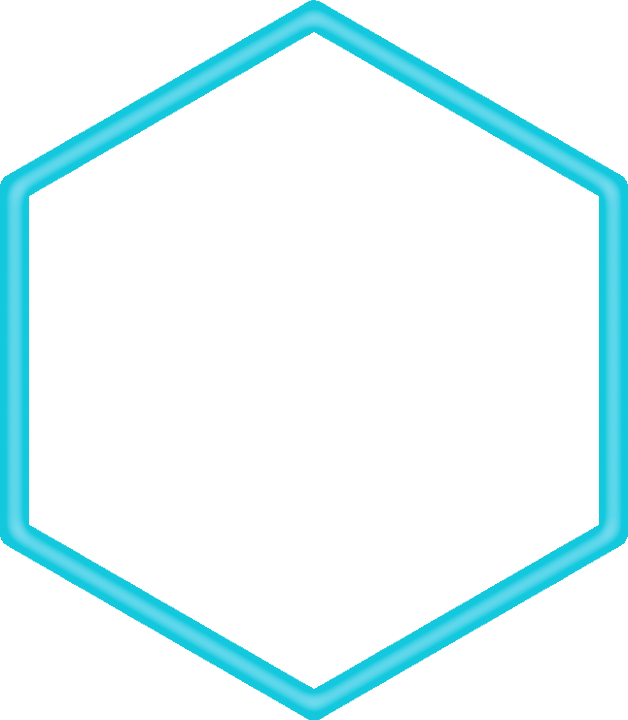
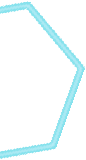
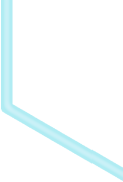
**Question 2/3**

# Specification

Write a simple markdown parser function that will take in a single line of markdown and translate it into the appropriate HTML.

To keep it simple, support only one feature of markdown in atx syntax: headers. Headers are designated by 1-6 hash characters followed by a space, followed by text. The number of hashes determines the header level of the HTML output.

Header content should only come after the initial hashtag(s) plus a space character. Invalid headers should just be returned as the markdown that was received, no translation necessary. Spaces before the hashes and after the header should be kept in the output, but between the hashes and the text only one space is allowed.



# Examples

**"# Header" should return "<h1>Header</h1>"**

**"## Header" should return "<h2>Header</h2>"**

**"###### Header" should return "<h6>Header</h6>"**

**"####### Header" should return "####### Header" (too many hashes)**

**"### Header" should return "### Header" (too many spaces between)**

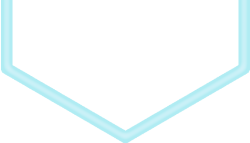
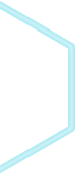
**"Header" should return "Header" (no hashes)**

# Input Output

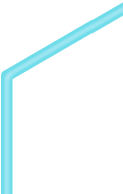
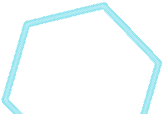
A string that contains html formatted text, if the input is valid markdown.

Otherwise, it just returns the input.

A string that contains markdown text.



**Specification**



You are given the following code, to be merged into "production"-running app.

Do a review (as thorough as you'd like) and provide suggestions on how to fix the issues (if there are any).

**Question 3/3**



using System;

public class Class1 {

public static bool IsFormat(string str, string f)  
 {  
 var allowedDict = new Dictionary<string, bool>()  
 {  
 { "number",true},  
 { "date",true},  
 { "timespan",true}  
 };  
  
 int isNotAllowed = 0;  
 for (var i = 1; i < allowedDict.Count; i++)  
 {  
 if (f == allowedDict.Keys.ToArray()[i - 1])  
 {  
 isNotAllowed |= 1 << i;  
 }  
 }  
  
 if (isNotAllowed > 0)  
 throw new Exception("Format not allowed.");  
  
 if (f.ToLower() == "number")  
 return Int32.TryParse(str, out var \_);  
 else if (f.ToLower() == "date")  
 return DateTime.TryParse(str, out var \_);  
 else if (f.ToLower() == "timespan")  
 return DateTime.TryParse(str, out var \_);  
  
 throw new Exception("Unknown format.");  
 }

}