Exercise 8

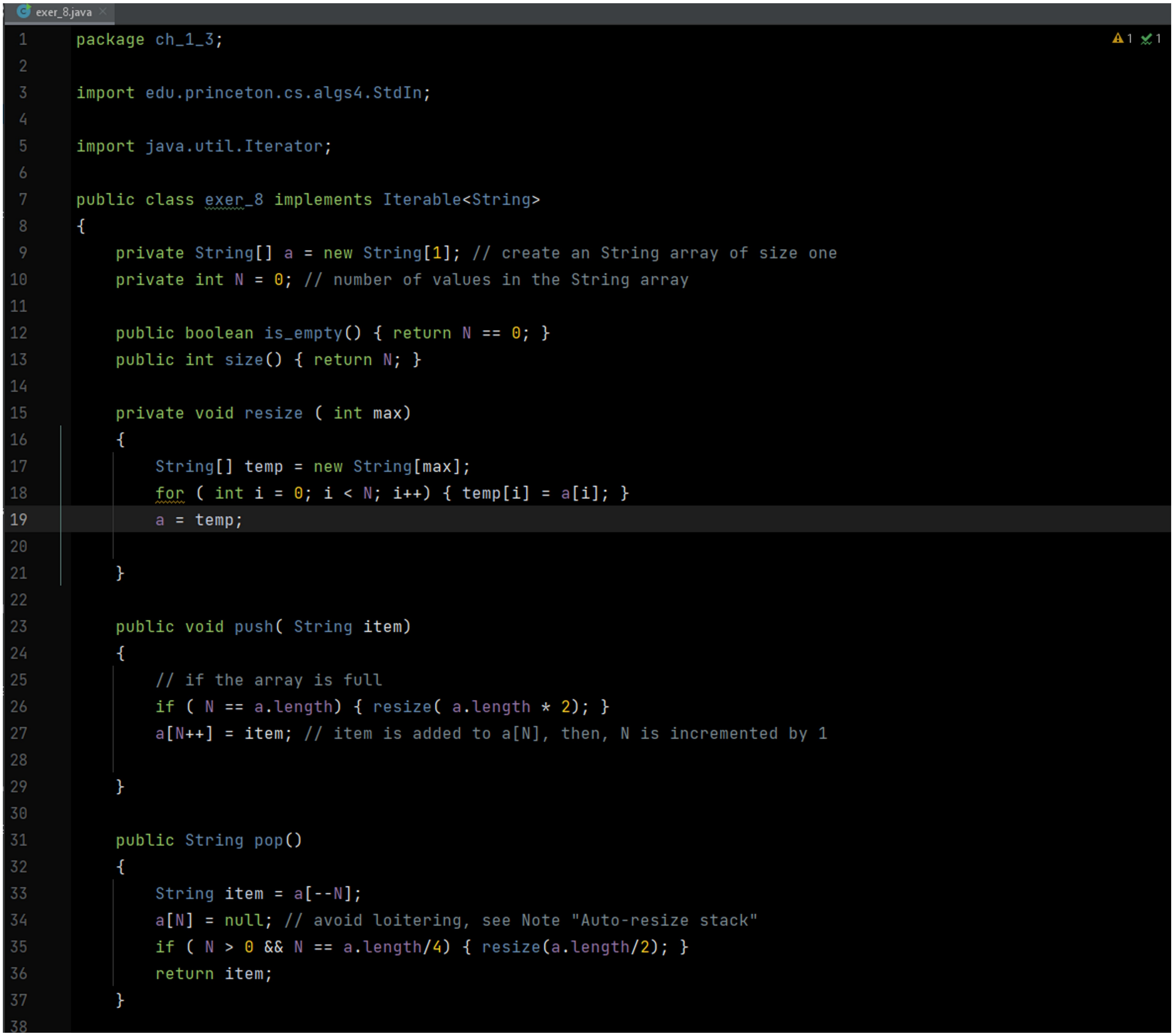
Wednesday, September 14, 2022 1:55 PM

1.3.8 Give the contents and size of the array for `DoublingStackOfStrings` with the input

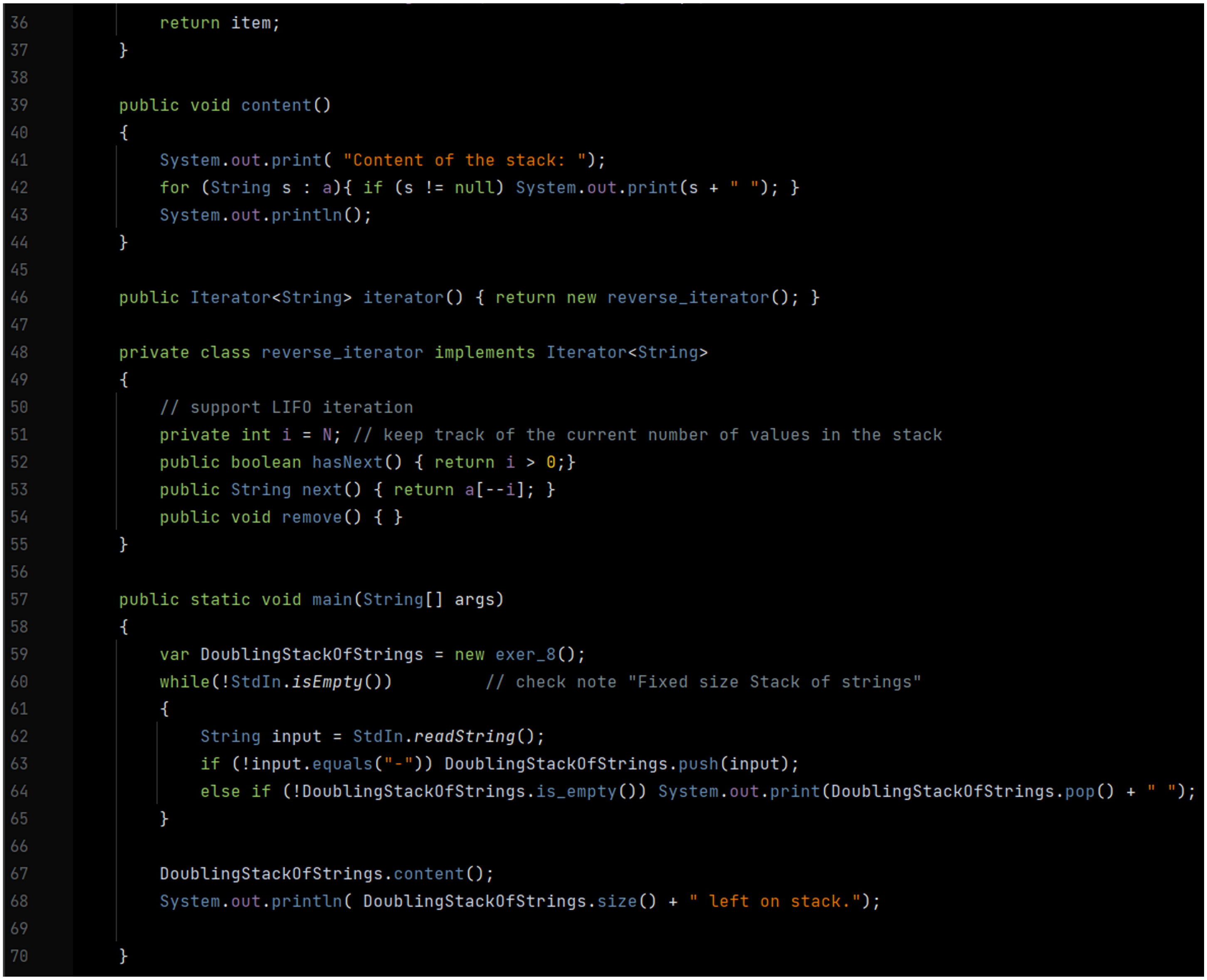
```
it was - the best - of times - - - it was - the - -
```

The question is asking you to create a class that implement stack using array and this class can automatically resize the array while the client push items into the stack or pop items from the stack. The type of your stack class should only accept String values and the stack should resize with a constant factor of 2.

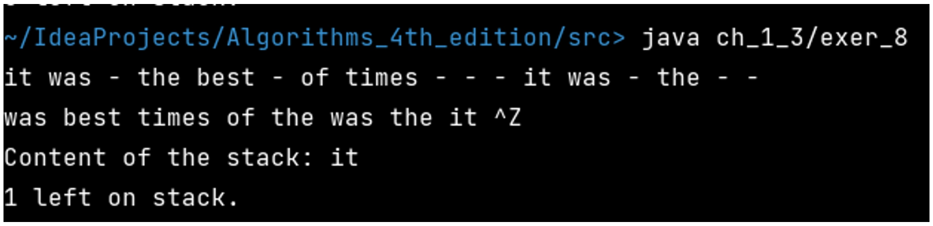
Please use **ResizingArrayStack<Item>** as a reference for this problem. (check Auto-resize stack note)



```java
package ch_1_3;

import edu.princeton.cs.algs4.StdIn;

import java.util.Iterator;

public class exer_8 implements Iterable<String>
{
    private String[] a = new String[1]; // create an String array of size one
    private int N = 0; // number of values in the String array

    public boolean is_empty() { return N == 0; }
    public int size() { return N; }

    private void resize ( int max)
    {
        String[] temp = new String[max];
        for ( int i = 0; i < N; i++) { temp[i] = a[i]; }
        a = temp;

    }

    public void push( String item)
    {
        // if the array is full
        if ( N == a.length) { resize( a.length * 2); }
        a[N++] = item; // item is added to a[N], then, N is incremented by 1

    }

    public String pop()
    {
        String item = a[--N];
        a[N] = null; // avoid loitering, see Note "Auto-resize stack"
        if ( N > 0 && N == a.length/4) { resize(a.length/2); }
        return item;
    }

```



```java
        return item;
    }

    public void content()
    {
        System.out.print( "Content of the stack: ");
        for (String s : a){ if (s != null) System.out.print(s + " "); }
        System.out.println();
    }

    public Iterator<String> iterator() { return new reverse_iterator(); }

    private class reverse_iterator implements Iterator<String>
    {
        // support LIFO iteration
        private int i = N; // keep track of the current number of values in the stack
        public boolean hasNext() { return i > 0;}
        public String next() { return a[--i]; }
        public void remove() { }
    }

    public static void main(String[] args)
    {
        var DoublingStackOfStrings = new exer_8();
        while(!StdIn.isEmpty())         // check note "Fixed size Stack of strings"
        {
            String input = StdIn.readString();
            if (!input.equals("-")) DoublingStackOfStrings.push(input);
            else if (!DoublingStackOfStrings.is_empty()) System.out.print(DoublingStackOfStrings.pop() + " ");
        }

        DoublingStackOfStrings.content();
        System.out.println( DoublingStackOfStrings.size() + " left on stack.");

    }
```



```
~/IdeaProjects/Algorithms_4th_edition/src> java ch_1_3/exer_8
it was - the best - of times - - - it was - the - -
was best times of the was the it ^Z
Content of the stack: it
1 left on stack.
```