Auto Indent

Mr. Panda has a piece of code that he needs to submit for his CS2040 programming assignment. As part of the grading criteria, he knows that a significant portion of his marks will go towards programming style. However, being a competitive programmer, he does not bother to indent his code properly while he codes, thus he needs to redo all the indentation after he finishes coding. As a fellow CS2040 student, you've kindly volunteered to help Mr. Panda solve his indentation problem, and hopefully, earn 3% yourself.

Mr. Panda codes very consistently. In particular, it is guaranteed that:

1. Every open curly bracket will match with a close curly bracket. Additionally, a close curly bracket will only appear after its corresponding open curly bracket.
2. Mr. Panda adopts the Allman coding style. In other words, every curly bracket is on a new line with no other trailing characters. Refer to the sample input for an illustration. However, note that the bracket **can still be poorly indented**, i.e. there can be spaces before a bracket on its own line.
3. Mr. Panda's source code only contains alphanumeric characters and printable ASCII symbols (ASCII values 32-126 inclusive). The space character (ASCII value 32) is used for indentation. Similarly, you **should not use the tab character** to indent in your output.
4. There will be no empty lines, or lines containing only spaces.
5. There will not be trailing spaces in any line.

You only need to format Mr. Panda's code according to the simplified rules of indentation as follows:

1. The first line should be at indentation level 0, i.e. it has no spaces on its left.
2. When you encounter an open curly bracket,
    a. **Every subsequent line** of code nested within needs to be indented by an additional **2 spaces**.
    b. This indentation should continue until a matching close curly bracket is encountered. The close curly bracket should be indented back **at the same level as the open curly bracket**. In other words, it should **not** be indented by the extra 2 spaces.
3. The indentation stacks, i.e. code that are nested within **K** curly brackets need to be indented by **2K** spaces.
4. Note that the original source code can be already indented with any number of spaces; you need to indent/unindent as necessary.
5. Otherwise, you should **not** modify any other part of the source code, including but not limited to: adding unnecessary spaces/indentation, capitalisation, or adding extra newlines.

Input

The input will contain Mr. Panda's source code. You are to process the entire source code until the end of file is encountered.

Output

The source code indented as specified as per the rules above.

Limits

- There will be at most 2500 lines and 50000 characters in Mr. Panda's source code.
- There will be no more than 500 **pairs** of **{** and **}**.

Sample Input (**autoindent1.in**)	Sample Output (**autoindent1.out**)

`public class Autoindent`
`{`
`public void run()`
`{`
`return;`
`}`
`}`
	`public class Autoindent`
	`{`
	`  public void run()`
	`  {`
	`  return;`
	`  }`
	`}`

Sample Input (**autoindent2.in**)	Sample Output (**autoindent2.out**)

`public class GgEqualsG`
` {`
`//the most important method`
`  public void run()`
`  {`
` print("who  needs gg=G");`
`  }`
`}`
	`public class GgEqualsG`
	`{`
	`  //the most important method`
	`  public void run()`
	`  {`
	`  print("who  needs gg=G");`
	`  }`
	`}`

On line 6 of **autoindent2.in**, notice that spaces within the line are preserved.

Notes:

1. You should develop your program in the subdirectory **ex2** and use the skeleton java file provided. You should not create a new file or rename the file provided.
2. You are free to define your own helper methods and classes (or remove existing ones).
3. Please be reminded that the marking scheme is:
   a. Public Test Cases (1%) - 1% for passing **all** test cases, 0% otherwise
   b. Hidden Test Cases (1%) - Partial scoring depending on test cases passed
   c. Manual Grading (1%)
      i. Overall Correctness (correctness of algorithm, severity of bugs)
      ii. Coding Style (meaningful comments, modularity, proper indentation, meaningful method and variable names)
4. Your program will be tested with a time limit of not less than **2 sec** on Codecrunch.

Skeleton File – Autoindent.java

You are given the below skeleton file `Autoindent.java`. You should see a non-empty file when you open the skeleton file. Otherwise, you might be in the wrong working directory.

```
/**
 * Name          :
 * Matric. No    :
 * PLab Acct.    :
 */
import java.util.*;

public class Autoindent {
    private void run() {
        //implement your "main" method here
    }

    public static void main(String[] args) {
        Autoindent newAutoindent = new Autoindent();
        newAutoindent.run();
    }
}
```