```java
package xyz.amtstl.soup;

import java.util.List;

public class Parser {
    private static int inx;

    /**
     * Parser that gets the numbers
     * @param i
     * @param cache
     * @return
     * @throws NumberFormatException
     * @throws SoupVariableException
     * @deprecated use the new Parser (it's integrated with InterVar)
     */
    public String[] ~~parseNumbers~~(int i, String cache) throws NumberFormatException, SoupVariableException {
        String whole = "";

        int index = 0;

        for (int e = i; e < cache.length(); e++) {
            if (cache.charAt(e) == '}'){
                index = e;
                break;
            }
            else if(cache.charAt(e) != '}') {
                whole+=cache.charAt(e);
            }
        }

        inx = index;

        whole = whole.substring(2, whole.length());

        String[] numbers = whole.split(",");
        return numbers;
    }

    /**
     * General parser for soup
     * @param i
     * @param cache
     * @return
     * @throws NumberFormatException
     * @throws SoupVariableException
     * @throws SoupSyntaxException
     */
    public static List<String> parse(int i, String cache) throws NumberFormatException, SoupVariableException, SoupSyntaxException {
        String whole = "";

        int index = 0;

        for (int e = i; e < cache.length(); e++) {
            if (cache.charAt(e) == '}'){
                index = e;
                break;
            }
            else if(cache.charAt(e) != '}') {
                whole+=cache.charAt(e);
            }
        }

        inx = index;

        whole = whole.substring(2, whole.length());

        InterVar.parseInternalVar(whole.split(","));
        return InterVar.getParsedNumbers();
    }

    /**
     * Parser for special functions that don't use the bracket delimiter
     * @param i
     * @param cache
     * @return
     * @throws NumberFormatException
     * @throws SoupVariableException
     * @throws SoupSyntaxException
     */
    public static List<String> parseInternalFunctions(int i, String cache) throws
NumberFormatException, SoupVariableException, SoupSyntaxException {
        String whole = "";

        int index = 0;

        for (int e = i; e < cache.length(); e++) {
            if (cache.charAt(e) == ')'){
                index = e;
                break;
            }
            else if(cache.charAt(e) != ')') {
                whole+=cache.charAt(e);
            }
        }

        inx = index;

        whole = whole.substring(2, whole.length());

        InterVar.parseInternalVar(whole.split("!"));
        return InterVar.getParsedNumbers();
    }

    /**
     * Parses a single number
     * @deprecated
     * @param i
     * @param cache
     * @return
     */
    public static String ~~parseSingle~~(int i, String cache) {
```

```java
        String whole = "";

        int index = 0;

        for (int e = i; e < cache.length(); e++) {
            if (cache.charAt(e) == '}'){
                index = e;
                break;
            }
            else if(cache.charAt(e) != '}') {
                whole+=cache.charAt(e);
            }
        }

        inx = index;
        System.out.println(String.valueOf(whole.charAt(2)));
        whole = whole.substring(2, whole.length());
        return whole;
    }

    public int getIndex() {
        return inx;
    }

    public void setIndex(int newIndex) {
        inx = newIndex;
    }
}
```