**9. JavaScript Object Notation**

# Program Name: JSON.java Input File: json.dat

JavaScript Object Notation (JSON) is a way of representing data in plaintext files that makes it easier to read than the dat files we usually use at contests- at least it appears to be easier. To find out if it really is easier for you, the input for this problem is in JSON notation.

**Input**

JSON stores objects in key value pairs. Keys will always be a string, but values can be a number, string, array, or another JSON object. To access some value, you can use its key. Since values can be objects, it is possible to nest these keys together. For example, with input JSON

{

"data": {

"number": 6,

"word": "hello",

"todo": [

"wash dishes",

"homework",

"212"

]

},

"something": "else"

}

data would return the nested JSON object within data, data number would return 6, and data todo 1 would return homework. Note: It is possible for two keys to have the same name if they are not into the same scope. For example:

{

"data": {

"something": "cool"

},

"something": "uncool"

}

uncool would be accessed through something, and cool would be accessed through data something.

**Input**

There will first be an unknown number of lines of input in JSON format, which is delimited by the first { at the beginning of JSON input and the last } at the end. (There may be more than one set of curly braces; those signify the starts and ends of values that are JSON objects themselves). After the JSON input, there will be an unknown number of lines which each contain keys and/or array positions. Array positions will always be an integer >= 0, and a valid array index. You may assume all keys in the input file exist in the json, and there will not be any erroneous data. Nested keys will be separated by a space, so to access a key n inside an object that is the value of a key m, the input would be in the format n m.

**Output**

For each of these lines, print out the value for the key that is being requested. You can assume that no key will point to an entire object or array, only to strings and integers. Nested keys will be located inside other keys that point to an object or array, but your final innermost value will always either be an integer or a string.

**Example Input File**

{

"data": {

"number": 6,

"word": "hello",

"todo": [

"wash dishes",

"homework",

"212"

]

},

"something": "else"

}

something

data todo 1

data number

data word

**Example Output to Screen**

else

homework

6

hello