Extra Credit: Compiler 7 - Strings, Printing, and Dynamic Memory

Add string variables and string literals to your compiler so that the following line of code adds the string data to the .data section of the executable:

```
let message: str = "Hello, World!"
```

Add built-in functions `print(s: str)`, `print_int(n: int)`, and `print_nl()` to your compiler. These should print the given string, the given integer, or a new line to stdout, respectively. The following program:

```
fn main() -> int
{
    call print("The answer is ")
    call print_int(42)
    call print_nl()
    call print("What is the question?")
    call print_nl()
    return 0
}
```

should print the following to the console:

```
The answer is 42
What is the question?
```

Also, add dynamic memory to your compiler. To do this, you must extend the type system to handle arrays. From now on a type is a basic type along with an indirection count. The basic types are `int`, `str`, and `bool`. Any basic type can be turned into an array by adding square brackets around them, or an array of arrays by adding two pairs of brackets, and so on. So `int` is a single integer and has an indirection count of 0, `[bool]` is an array of bools with an indirection count of 1, `[[[str]]]` is an array of arrays of arrays of strings and has indirection count 3.

Create a built-in function `alloc(item_count: int) -> [any]` that allocates 8*item_count bytes and returns a pointer to the beginning of the allocation. We can read and write elements of an array using indexing operations like the following:

```
let array: [int] = alloc(2)
set array[0] = 23
set array[1] = 19
return array[0] + array[1]   // Should return 42
```

Each indexing operation removes one level of indirection.

As a final test of your compiler, here is an implementation of the game of life in jive: gol.jive . It should print a sequence of iterations to the console of a glider moving diagonally downwards and to the right, starting as so and continuing through iteration 12:

```
Iteration: 0
+----------------+
|                |
|    []          |
|      []        |
|  [][][]        |
|                |
|                |
|                |
|                |
+----------------+

Iteration: 1
+----------------+
|                |
|                |
|  []  []        |
|    [][]        |
|    []          |
|                |
|                |
|                |
+----------------+
```

Good luck, and have fun! This is an extra credit assignment worth up to 100 points

需要在新的浏览器窗口中加载此工具

在新窗口中加载 Extra Credit: Compiler 7 - Strings, Printing, and Dynamic Memory