Homework: Extended Interpreter

In this assignment, you will augment the interpreter you wrote earlier with with mutable arrays.

1 Restrictions

You cannot use arrays or any other mutable data structures in this assignment. That would defeat the purpose of building arrays from scratch.

2 Requirements

Write a program that takes one argument:

`./xinterp.d.byte` *program*

The argument *program* should be the name of a file that contains a program written using the grammar defined in fig. 8.1, which extends the grammar in fig. 6.1 from the last assignment. Your interpreter may output the result of the program in any format you like. However, **if the result of an operation is not defined, the interpreter must exit with a non-zero exit code.** (In OCaml, if your program terminates with an unhandled exception or a false assertion, it wil exit with code 2.)

You should implement a call-by-value semantics. We discussed the paper *The Essence of JavaScript*, which presents the semantics of mutable references. The principles behind that approach apply to arrays too.

3 Support Code

In the support code for the class, the `Xinterp_util` module defines an abstract syntax for the language and a parser for the concrete syntax shown in fig. 8.1. You are not strictly required to use the support code. In particular, you are free to copy it and modify it if you wish to extend the language in any way. However, any extensions you make must be compatible with the base language.

4 Template and Hand In

A template file for the assignment is provided on the course web page. Solve the assignment in this file and submit only this file using Moodle.

Expressions

e	::=	$\cdots$		
	|	`array(`e_1`,`e_2`)`	`MkArray (`e_1`, `e_2`)`	Allocates an array of length e_1 with e_2 at every index
	|	e_1`[`e_2`]`	`GetArray (`e_1`, `e_2`)`	Produces the value at index e_2 in the array e_1
	|	e_1`[`e_2`]` $= e_3$	`SetArray (`e_1`, `e_2`, `e_3`)`	In the array e_1, set the value at index e_2 to the value of e_3 and returns that value

Figure 8.1: The concrete syntax and abstract syntax of the language.