This is *ginseng 1.1.0*, our very first compiler for the *Ginseng* programming language, developed by us.

**Ginseng programming language**

*Ginseng* is a small, compact and strongly typed programming language. It has the following characteristics:

* **Set-oriented paradigm**: *Ginseng* provides a native *set* type, that reflects the properties of its mathematical counterpart (more on this in the type system section).
* **Embedded operators**: provides the developers with a collection of built-in operators to efficiently manipulate data.
* **Simple and intuitive notation**: *Ginseng* operations follow the most common notational conventions, making it an intuitive and easy to use programming language for those who have even just basic knowledge of math.

**Ginseng type system**

Ginseng is a strongly typed language: *type inference* is used during compilation to infer variable types and enforcing operations on such. The reason behind this is to keep the syntax as light as possible, making the task of writing a ginseng source file as fast and less verbose as possible, while maintaining a good level of expressiveness in the meantime.

Ginseng offers the following \_ native data types:

* **number**: this is the basic data type. It is used to represent numbers of any kind, i.e. integers and floating-points. Number variable ids can contain alphanumeric characters only and must start with a lowercase letter.
* **array (sequences)**: this data type is the equivalent to the array data structure native to most programming languages. It fits good in abstracting mathematical sequences, as they have the same ordering and the same indexing. It is possible to declare an array variable, populate it and remove elements from it, either via functions or via the classical index notation ([]).  
  Arrays are declared with variables identifiers within the so-called “diamond”, i.e. <>
* **set**: data type which abstracts the mathematical concept of a set. Basically, it represents an unordered collection of unique numbers.  
  Sets support the common set operations, such as membership tests, intersection with another set, union and difference. It is also possible to add single number variables to a set, and even to use them as operands in one of the previously mentioned set operations! A number used as operand along with a set is interpreted as a *singleton*, which is a special type of set containing just a single element.  
  Following the most widespread mathematical convention, sets are declared with identifiers starting with an uppercase letter.