Experimenting with C extensions in Ruby

Array things in C

```c
#include "ruby.h"

static VALUE
t_init(VALUE self)
{
  return self;
}

static VALUE new_ary(VALUE self){
  VALUE ary;
  ary = rb_ary_new();
  return ary;
}

static VALUE ary_push(VALUE self, VALUE in_){
  VALUE ary = rb_ary_new();
  ary = rb_ary_push(ary, in_);
  return ary;
}

static VALUE ary_size(VALUE self, VALUE in_ary){
  VALUE len = RARRAY_LEN(in_ary);
  return INT2NUM(len);
}

VALUE cArrays;
void Init_arrays()
{
  cArrays = rb_define_class("Arrays", rb_cObject);
  rb_define_method(cArrays, "initialize", t_init, 0);
  rb_define_method(cArrays, "ary", new_ary, 0);
  rb_define_method(cArrays, "push", ary_push, 1);
  rb_define_method(cArrays, "size", ary_size, 1);
}
```

extconf.rb

```ruby
require "mkmf"
```

```
create_makefile("arrays")
```

test_c_ext.rb

```
require_relative 'arrays'
require 'minitest/autorun'
require 'benchmark'
#
class TestCExt < MiniTest::Test
  def test_time
    n = 1_000_000
    Benchmark.bm(7) do |x|
      x.report('arrays_c:') { n.times { ; t1 = Arrays.new.ary; } }
      x.report('array:') { n.times { ; t2 = []; } }
    end
  end

  def test_c_ext
    the_extension = Arrays.new
    assert_equal Object, Arrays.superclass
    assert_equal Arrays, the_extension.class
  end

  def test_ary
    array = Arrays.new
    assert_equal [], array.ary
    new_array = array.push('a string')
    assert_equal ['a string'], new_array
    assert_equal 1, array.size(new_array)
    p array.push(10)
    new_array.push
    p new_array
  end
end
```

```
Run options: --seed 17968

# Running:

[10]
["a string"]
..              user     system      total        real
arrays_c:  0.438000   0.000000   0.438000 (  0.494209)
array:     0.219000   0.000000   0.219000 (  0.206909)
```

```
.

Finished in 0.705872s, 4.2501 runs/s, 7.0834 assertions/s.

3 runs, 5 assertions, 0 failures, 0 errors, 0 skips
```

More to learn....

How can you play the game if you don't know the rules?