Ruby Basics:

While loop:

repeats a section of code as long as a specific condition is true

generally requires the condition to change as the loop executes
in order to avoid infinite loops

while loop: syntax

```
while condition do
    ...
end
```

Helpful math

my_integer += 1

while loop example

```
my_var = 10

while my_var < 30 do
    puts "my_var = #{my_var}"
    my_var += 1
end
```

Ruby Basics:

loops and arrays: the perfect match

     incremental indexes in the array
     incrementing value of test condition variable

eg

```ruby
pets = ["cats", "dogs", "ducks", "chickens", "fish" ]
pet_index = 0
while pet_index < pets.count do
    if pets[pet_index] == "ducks" ||
              pets[pet_index] == "chickens"
        pets[pet_index] = nil
    end
    pet_index += 1
end
```

until loop: the inverse of while

```
my_var = 10

until my_var > 30 do
    puts "my_var = #{my_var}"
    my_var += 1
end
```

Try it yourself:

      Write a program that asks the user for 10 integers. Then sort the integers from least to greatest. Output the sorted list to the user.

Ruby Basics:

```ruby
integers = []
current_integer = 0
while current_integer < 10 do
    puts "Type an integer"
    integers[ current_integer] = gets.to_i
     current_integer+=1
end

integers.sort.each do |this_int|
    puts this_int
end
```