Ruby trunk - Bug #15103

undecipherable nil error for `+`

09/11/2018 04:02 PM - jzakiya (Jabari Zakiya)

Status:	Rejected		
Priority:	Normal		
Assignee:			
Target version:			
ruby -v:		**Backport:**	2.3: UNKNOWN, 2.4: UNKNOWN, 2.5: UNKNOWN

Description

I am experiencing the following error in a method I rewrote that makes no sense whatsover.

Here's the original method.

```
      def method1(num, residues, flag)
        modpg, rescnt = (residues[-1] - 1), residues.size
        num = 2 if num < 2
        num -= 1; flag ? (num |= 1; k = num/modpg) : (k = (num - 1)/modpg)
        modk = k * modpg; r = 0
        while num >= modk + residues[r]; r += 1 end
        [k * rescnt + r, r, modk]
      end
```

Here's the modified method, where I put a debugging line to see variable values.

```
      def method2(num, residues, flag)
        modpg, rescnt = (residues[-1] - 1), residues.size
        num = 2 if num < 2
        flag ? (k = ((num - 1) | 1)/modpg) : (k = (num - 2)/modpg)
        modk = k * modpg; r = 0

        puts "k = #{k}, modk = #{modk}, r = #{r}, residues[r] = #{residues[r]}"

        while num >= modk + residues[r]; r += 1 end
        [k * rescnt + r, r, modk]
      end
```

They both produce correct results for all inputs when flag = true.

The problem exists in method2 when flag = false AND the input num
is an additive multiple of the last element in the array residues.

Examples:
Using residues = [7, 11, 13, 17, 19, 23, 29, 31]
errors occur in method2 when flag = false and num = 31 + n*30 => 31, 61, 91....

```
2.6.0-preview2 :179 >       def method2(num, residues, lte)
2.6.0-preview2 :180?>           modpg, rescnt = (residues[-1] - 1), residues.size
2.6.0-preview2 :181?>           num = 2 if num < 2
2.6.0-preview2 :182?>           lte ? (k = ((num - 1) | 1)/modpg) : (k = (num - 2)/modpg)
2.6.0-preview2 :183?>           modk = k * modpg; r = 0
2.6.0-preview2 :184?>           puts "k = #{k}, modk = #{modk}, r = #{r}, residues[r] = #{residues
[r]}"
2.6.0-preview2 :185?>           while num >= modk + residues[r]; r += 1 end
2.6.0-preview2 :186?>           [k * rescnt + r, r, modk]
2.6.0-preview2 :187?>         end
 => :method2

2.6.0-preview2 :190 > method2  30, residues, false
k = 0, modk = 0, r = 0, residues[r] = 7
 => [7, 7, 0]
```

```
2.6.0-preview2 :191 > method2  32, residues, false
k = 1, modk = 30, r = 0, residues[r] = 7
 => [8, 0, 30]
2.6.0-preview2 :192 > method2  31, residues, false
k = 0, modk = 0, r = 0, residues[r] = 7
Traceback (most recent call last):
        4: from /home/jzakiya/.rvm/rubies/ruby-2.6.0-preview2/bin/irb:11:in `<main>'
        3: from (irb):192
        2: from (irb):185:in `method2'
        1: from (irb):185:in `+'
TypeError (nil can't be coerced into Integer)


2.6.0-preview2 :193 >
2.6.0-preview2 :194 > method2  60, residues, false
k = 1, modk = 30, r = 0, residues[r] = 7
 => [15, 7, 30]
2.6.0-preview2 :195 > method2  62, residues, false
k = 2, modk = 60, r = 0, residues[r] = 7
 => [16, 0, 60]
2.6.0-preview2 :196 > method2  61, residues, false
k = 1, modk = 30, r = 0, residues[r] = 7
Traceback (most recent call last):
        4: from /home/jzakiya/.rvm/rubies/ruby-2.6.0-preview2/bin/irb:11:in `<main>'
        3: from (irb):196
        2: from (irb):185:in `method2'
        1: from (irb):185:in `+'
TypeError (nil can't be coerced into Integer)
2.6.0-preview2 :197 >
```

The outputs for num = 31, 61.. should be the same for num = 30, 60..

The line: 1: from (irb):185:in `+' refers to while num >= modk + residues[r]; r += 1 end

The only thing I can think of is, for some reason in modk + residues[r] somehow r
is set past the last index of residues causing a nil, but I don't see how or why that can occur.

This behavior is consistent in Ruby 2.5.1, 2.6.0-preview2, Jruby-9.2.0.0 and Truffleruby-1.0.0-rc6.

This same code runs with no errors in Crystal.

History

#1 - 09/11/2018 04:16 PM - jzakiya (Jabari Zakiya)

Never mind, I found the error. Please close this.

They were not operationally exact.
This while num >= modk + residues[r]; r += 1 end
had to be this while (num-1) >= modk + residues[r]; r += 1 end

Now I remember why I did that in the first place.

#2 - 09/12/2018 02:22 AM - nobu (Nobuyoshi Nakada)

- Status changed from Open to Rejected