23. Find the sum of all the positive integers which cannot be written as the sum of two abundant numbers.

According to wikipedia, all numbers greater than 20161 can be written as the sum of two abundant numbers.

```
Make an object with the numbers 12 - 20161 as keys
Make an array of the keys of that object
Loop through the numbers n from 12 - 20161
    Loop through the abundant (a) numbers < n
        If n - a is a key in the object, go to next n
        If n - a is not a key in the object for all a < n, add n to list
Return the sum of the list
```

```
def divisors(num)
  div = []
  1.upto(num / 2) do |n|
    div.push(n) if (num % n).zero?
  end
  div
end

def abundant?(num)
 divisors(num).sum > num
end

def generate_abundant(num)
  list = {}
  12.upto(num) do |n|
    list[n] = true if abundant?(n)
  end
  list
end
```

```
def non_abundant_sum()
  stop = 20161
  abundant = generate_abundant(stop)
  abundant_arr = abundant.keys
  to_sum = []
  1.upto(stop) do |n|
    i = 0
    not_found = true
    while abundant[i] < n - 1
      if abundant.has_key?(
                    n - abundant_arr[i])
        not_found = false
        break
      end
      i += 1
    end
    to_sum.push(n) if not_found
  end
  to_sum.sum
end
```