

【atcoder】練習10問

問0:absとs

```
a = gets.to_i
b,c = gets.chomp.split.map(&:to_i)
s = gets.chomp
puts "#{a+b+c} #{s}"
```

*gets.chomp.split.map(&:to_i)でスペースを空けて入力した際に複数の数値をとることができる。

問1:責は奇数か偶数

```
a,b = gets.chomp.split.map(&:to_i)
puts (a*b).even? ? 'Even' : 'Odd'

=begin
if (a*b).even? == true
  puts "Even"
else
  puts "Odd"
end
=end
```

*xx ? yy : zz .. .if xx then yy else zz end と同じ意味。

問2:3つの1or0

```
nums = gets.chomp.split("").map(&:to_i)
puts nums.sum
#num.inject(0){ |cnt,n| cnt += n }
```

*split("")...数字を分ける 例)22 =>[2,2]

*sum...要素の合計を返す。例えば [e1, e2, e3].sum は init + e1 + e2 + e3 を返す。

ブロックが与えられた場合、加算する前に各要素にブロックが適用される。

配列が空の場合、initを返します。

*inject...たたみこみ演算を行う。

問3:N個の偶数何回割れるか

```
num = gets.to_i
nums = gets.chomp.split.map(&:to_i)
sum = 0
```

```

while nums.all?(&:even?) do
  sum += 1
  #sum = sum.succ
  nums = nums.map{ |n| n/2 }
end
puts sum

```

*all? ...すべての要素が真である場合に true を返す。偽である要素があれば、ただちに false を返す。ブロックを伴う場合は、各要素に対してブロックを評価し、すべての結果が真である場合に true を返す。ブロックが偽を返した時点で、ただちに false を返す。

*succ...self(object) の「次の」文字列を返す

問4:硬貨の枚数とX円が何通りか

```

a,b,c,x = 4.times.map {gets.to_i}
sum = 0
(0..a).each do |an|
  (0..b).each do |bn|
    (0..c).each do |cn|
      sum += 1 if an*500 + bn*100 + cn*50 == x
    end
  end
end

=begin
for num1 in 0..a do
  for num2 in 0..b do
    for num3 in 0..c do
      sum += 1 if num1*500 + num2*100 + num3*50 == x
    end
  end
end
=end

puts sum

```

*times...self 回だけ繰り返す。self が正の整数でない場合は何もしない。

ブロックパラメータには 0 から self - 1 までの数値が渡される。

*複数の値を与えたいとき 例)5.times.map{ gets.to_i }

問5:1以上n以下の整数の桁がa以上b以下であるものの総和

```

n,a,b = gets.chomp.split.map(&:to_i)

def n_sum(n)
  sum = 0
  while n > 0 do
    sum += n % 10
    n /= 10
  end
  sum
end

ns = (1..n).map{ |i| [i,n_sum(i)] }.select{ |i| a <= i[1] && i[1] <= b }
  .inject(0){ |sum, i| sum + i[0] }
puts ns

```

*桁の計算方法...

while n > 0 do(0=0で終了)

sum += n % 10 (sum(=0)にnを足し、10の余りを取り、1の桁を取り出している)

n /= 10(nを10で割り、10の位を取り出している、2桁だと2週目で0になる)

end

sum (1と10の桁の値の和)

*select..各要素に対してブロックを評価した値が真であった要素を全て含む配列を返す。真になる要素がひとつもなかった場合は空の配列を返す。

問6:カードゲーム

```

n = gets.to_i
as = gets.strip.split.map(&:to_i)
alice = 0
bob = 0
until as.empty? do
  as.delete(as.max) if alice += as.max
  break if as.empty?
  as.delete(as.max) if bob += as.max
end
puts (alice - bob)

```

*until as.empty? do で存在しているという条件でループすることができる。

*max...最大値をとる。 ⇔ min

*delete...delete()で対象の値を消去できる。

問7:鏡餅

```
n = gets.to_i
ns = n.times.map { gets.to_i }
puts ns.uniq.length
```

*uniq...配列から重複した要素を取り除き、新しい配列を返す。

*length,size,count

問8:お年玉

```
n,y = gets.chomp.split.map(&:to_i)
a,b,c = -1, -1, -1
(0..n).each do |x|
  (0..(n-x)).each do |z|
    v = n - x - z
    if x * 10000 + z * 5000 + v * 1000 == y
      a,b,c = x,z,v
    end
  end
end
end

print a,b,c
```

- ・ each文の繰り返し記法
- ・ 初めに値を入れておいて、条件に応じて値を変えさせることを学んだ。