

# コンピュータサイエンス第一

## 課題 2 「循環小数の循環を止めよ！」

学籍番号\*\*\*\*\* \*\*\*\*

### 概要

入力した自然数で 1 を割ったときの値を小数点以下の位ごとに出力するプログラムが与えられた。しかし、そのプログラムは答えが循環小数になる場合に循環したまま終了できない。課題として、この問題の解決に取り組んだ。

## 1 実行方法と出力例

ターミナルを開き、junkan.rb が存在するディレクトリ内で、以下のコマンドを実行することにより、本プログラムを実行できる。

```
$ ruby junkan.rb
```

1/2、1/6、1/7 を計算させたときの出力例はそれぞれ以下の通りである。

分母 d を下さい

2

1 / 2 を求めます

1:5

Answer: 0.5

分母 d を下さい

6

1 / 6 を求めます

1:1

2:6

3:6

Answer: 0.1(6)

分母 d を下さい

7

1 / 7 を求めます

1:1

2:4

3:2

4:8

5:5

6:7

7:1

Answer: 0.(142857)

## 2 循環小数検知の仕組み

整数どうしの割り算であるから、筆算を考えたときに、下の余りが同じならば同じ商の位が立つ。すなわち、 $x * 10 / d$  (本プログラムでは、これを  $r$  とおいた。) の値が複数回登場したことを検知することで、循環小数であることを特定できる。

本プログラムでは、この機構を配列を用いて実装した。まず、配列 `rList` を長さ  $d$  で初期化した。ループのたびに、 $r$  がすでに `rList` に含まれているものであればプログラムを終了した。そうでなければ  $r$  の値をこの配列に格納した。

## 3 工夫した点

循環小数であることを検知するだけでなく、循環節の表示も実装した。循環節の始まりでは、 $r$  が同じだけでなく、 $q(=x * 10 \% d)$  も同じである必要があった。(たとえば、 $1/6$  を求めるときなど。) そこで、 $r$  と同様に `qList` という配列を作り、同じ組み合わせが出現したとき、循環節の開始と終了の位置を特定するようにした。

## 4 ソースコード

以下に `junkan.rb` のソースコードを示す。

```
# junkan.rb
# 配列の使い方の練習（循環小数を循環するまで求める）
# 入力: d
# 出力: 1/d の各桁を循環するまで求める

puts("分母_d_を下さい")
d = gets().to_i
print("1_/_", d, "_を求めます\n")
```

```

leng = 0;  x = 1
rList = Array.new(d) # 余りの整数を格納する配列
qList = Array.new(d) # 除の位の値を格納する配列

# 1の例外処理
if d == 1
  print('1')
  exit
end

while true
  x = x * 10
  q = x / d
  r = x % d

  # 出力
  print(leng + 1, ': ', q, "\n")
  sleep(0.5)
  qList[leng] = q
  rList[leng] = r

  # 割り切れるなら終了
  if r == 0
    print('Answer: 0.', qList[leng], "\n")
    break
  end

  for i in 0..(leng - 1)
    # 余りと除の位の値それぞれが等しい組み合わせがあったとき
    if r == rList[i] and q == qList[i]
      repeatBeginKey = i
      # 循環していない部分を表示
      print('Answer: 0. ')
      for j in 0..(repeatBeginKey - 1)
        print(qList[j])
      end
      # 循環節を表示

```

```

repeatEndKey = leng - 1
print(' ( ')
for j in repeatBeginKey .. repeatEndKey
    print(qList[j])
end
print(' ) ')
print("\n")
exit
end
end

# 次のループの準備
leng = leng + 1
x = r
end

```