

演習ガイド: Windows

1. 講義のウェブページから day4.zip をダウンロードする.
 - Downloads(ダウンロード)フォルダに day4.zip を置かれる
 - ファイルを開く → すべてを展開
 - 展開先として, Documents¥CS1 を指定して, 展開
2. コマンドプロンプトを実行
3. day4 フォルダに移動
4. 内容を確認

OneDriveを使用している場合, zipファイルの展開先が
OneDrive¥ドキュメント¥CS1
になるので下の最初のコマンドを変更

```
C:¥Users¥minamide> cd Documents¥CS1
C:¥Users¥minamide¥Documents¥CS1> cd day4
C:¥Users¥minamide¥Documents¥CS1¥day2> dir

...
                hindo.py
...
```

演習ガイド: Mac

1. 講義のウェブページから day4.zip をダウンロードする.
 - Downloads(ダウンロード)フォルダに day4.zip が展開される
2. Terminal を実行
3. Terminal でフォルダ day4 をCS1 に移動
4. day4 フォルダに移動
5. 内容を確認

```
$ cd Documents/CS1
$ mv ~/Downloads/day4 ./
$ cd day4
$ ls
hindo.py  junkan-hint.py  junkan.py
```

前回の課題について

関係演算：Python では、3個以上の値を比較できる

```
$ python  
...  
>>> 1 < 2 < 3  
True  
>>> 2 < 2 < 3  
False  
>>> 2 <= 2 < 3  
True
```

この授業では非推奨

- ほとんどのプログラミング言語ではできない

推奨する書き方

```
>>> 1 < 2 and 2 < 3  
True
```

配列：初期化（大きさが固定でない場合）

```
$ python
...
>>> [3, 1, 4, 1, 5, 9, 2]
[3, 1, 4, 1, 5, 9, 2]
>>> [0] * 10
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
>>> x = 5
>>> [3] * x
[3, 3, 3, 3, 3]
>>> exit()
```

Mac の場合は `python3`

各要素が 0 で, 長さ 10 の配列

各要素が 3 で, 長さ x の配列

配列を用いたプログラム: 頻度

入力:

- 非負整数 n
- 整数(0以上, n 未満)の配列

出力

- 配列に 0 から $n-1$ が, それぞれいくつ含まれるか

```
$ python hindo.py
非負整数を入力して下さい
5
0 から n-1 の整数配列を入力して下さい
0 3 0 1 2 0 1
0: 3 個
1: 2 個
2: 1 個
3: 1 個
4: 0 個
```

Mac の場合は `python3`

配列を用いたプログラム: 頻度

```
print("非負整数を入力して下さい")  
m = int(input())
```

```
print("0 から n-1 の整数配列を入力して下さい")  
a = list(map(int, input().split()))  
n = len(a)
```

```
hindo = [0] * m # 整数 0 で初期化された大きさ m の配列
```

```
for i in range(n):  
    hindo[a[i]] = hindo[a[i]] + 1 // 頻度の計算
```

```
for j in range(m):  
    print(j, ":", hindo[j], "個") // 頻度の出力
```

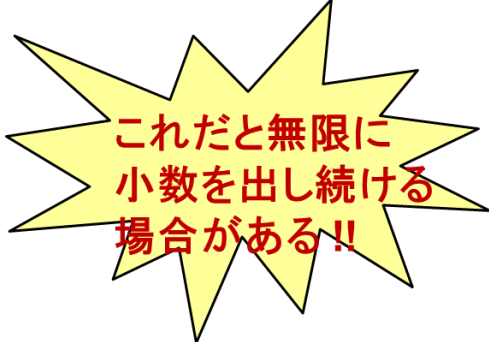
CS第1 レポート課題2

課題 循環小数の循環を止めよ！

```
print("分母 d を下さい")
d = int(input())
print("1 / ", d, " を求めます")

leng = 0
x = 1
while x != 0:
    x = x * 10
    q = x // d
    leng = leng + 1
    print(leng, ":", q)
    time.sleep(0.5)
    x = x % d
```

junkan.py



これだと無限に
小数を出し続ける
場合がある!!

プログラムの要求仕様

- ・ 循環小数になっても計算が止まる
- ・ 繰り返しが出る直前, 直後くらいまでに止める

実行例

```
$ python junkan.py
分母 d を下さい
4
1 / 4 を求めます
1 : 2
2 : 5
```

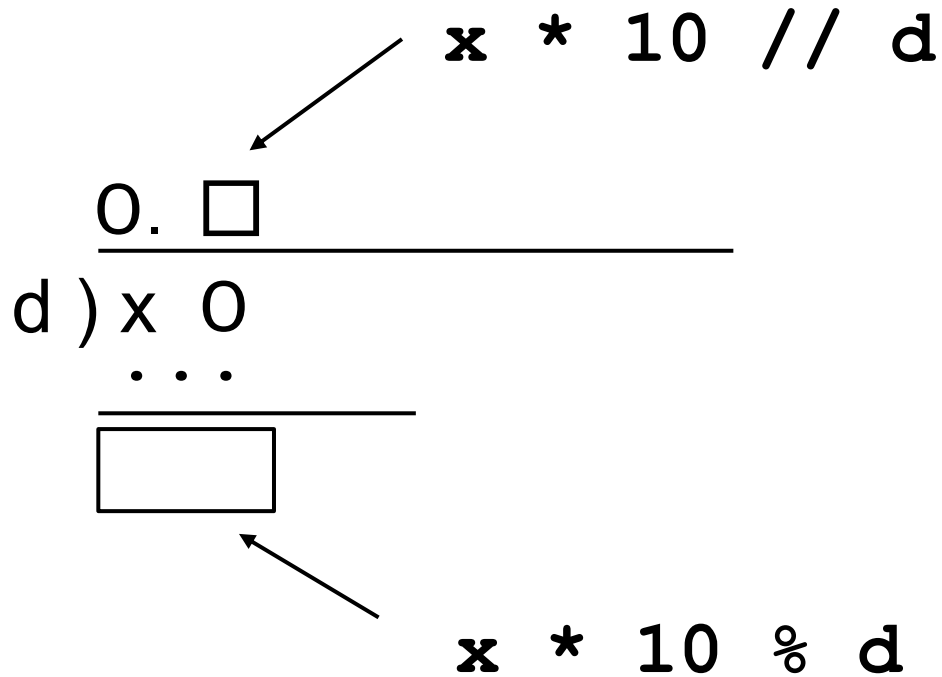
```
$ python junkan.py
分母 d を下さい
7
1 / 7 を求めます
1 : 1
2 : 4
3 : 2
4 : 8
5 : 5
6 : 7
7 : 1
8 : 4
...
```

強制終了の方法

- Windows:
control + break(pause)
または Fn + control + break
- Mac: control + c

プログラムの考え方

- x / d を小数として表示する. ただし, $x < d$



注意

n桁目とm桁目の数が同じでもそこで循環していない場合もある

```
$ python junkan.py
```

分母 d を下さい

17

1 / 17 を求めます

1 : 0

2 : 5

3 : 8

4 : 8

5 : 2

6 : 3

7 : 5

8 : 2

9 : 9

10 : 4

11 : 1

12 : 1

13 : 7

14 : 6

15 : 4

16 : 7

17 : 0

18 : 5

19 : 8

20 : 8

CS第1 レポート課題2

提出方法

- ・ OCW-i から提出: 11月4日 レポート課題2
- ・ 提出期限: 11月11日 午前10:40
 - 11月11日 22:00までのレポートは, 採点しますが, 大きく減点します.
- 提出物
 - レポート(PDFファイル)
 - ファイル名 **kadai2.pdf**
 - ファイル形式がPDFでない場合は減点します.
 - プログラム: **junkan.py**

CS第1 レポート課題2

提出すべきものと採点基準(満点 20)

1. レポート

- どうやって循環を見つけるのかの説明. A4 で 半ページ～1 ページ程度. 読み手は, そのやり方を知らないと思って書くこと(配点 5)
- 工夫した点(配点 5)
 - 循環する部分がわかるようにしよう

2. プログラムのファイル(配点 10)

- 循環を止められているか
- 循環する部分がわかるか

Pythonの配列

【配列】	配列の作成	[0, 0, -5, 4]	
		[0] * 4	各要素の初期値が 0, 要素数が4の配列
	要素の参照	aa[i]	aa の i 番目 (添え字 i は 0 から)
	大きさ	len(aa)	配列 aa の長さ (=要素数)
※「添え字」は「インデックス」(index) ともいう.			