

⚡ Type-C Talk ⚡

N Code Labo LT 大会 2025 年 3 月

自己紹介

名前：鎌田 一輝（かまた かずき）

所属：N Code Labo 横浜教室

とくいなこと：うえぶあぷりのかいはつ

よく使う技術：Angular(TypeScript), Django(Python)

大学入試センター試験の受験回数：2 回

今、一番アツいプログラミング言語って？🤔

Python? Java? TypeScript? Go? ...

DNCL

DNCL って何？

- 共通テスト手順記述標準言語
- **Daigaku Nyushi Center Language**
- 共通テスト「情報Ⅰ」のプログラミング系の問題で使用する
- 1系と2系がある
 - 1系：VBAなどに似ている。おそらく滅びた。
 - 2系：Pythonなどに似ている。今後こちらが使われそう。

参考: DNCL (1 系)

大学入試センターから [仕様書](#) が公開されている

↓ Fizz Buzz を書いてみるとこんな感じ

```
x を 1 から 30 まで 1 ずつ増やしながら,  
  もし  $x \% 15 = 0$  ならば  
    "FizzBuzz" を表示する  
  を実行し, そうでなくもし  $x \% 3 = 0$  ならば  
    "Fizz" を表示する  
  を実行し, そうでなくもし  $x \% 5 = 0$  ならば  
    "Buzz" を表示する  
  を実行し, そうでなければ  
    x を表示する  
  を実行する  
を繰り返す
```

今年の本試験の問題を見てみよう 🤝

第 3 問から抜粋

```
(01) Nissu = [4, 1, 3, 1, 3, 4, 2, 4, 3]
(02) kougaihin = 9
(03) Akibi = [1, 1, 1]
(04) buinsu = 3
(05) ケ を 1 から コ まで 1 ずつ増やしながら繰り返す:
(06) | tantou = 1
(07) | buin を 2 から buinsu まで 1 ずつ増やしながら繰り返す:
(08) | | もし キ ならば:
(09) | | | tantou = buin
(10) | 表示する("工芸品", kougaihin, "...",
      |       "部員", tantou, ":",
      |       Akibi[tantou], "日目～",
      |       Akibi[tantou] + サ, "日目")
(11) | Akibi[tantou] = Akibi[tantou] + シ
```

図 5 各工芸品の担当と期間の一覧を表示するプログラム

ケ・コ の解答群

① kougaihin ② tantou

③ buinsu ④ kougaihin

⑤ Nissu[kougaihin]

サ・シ の解答群

① Nissu[tantou]

② Nissu[kougaihin] - 1 ③ Nissu[tantou] - 1

④ Nissu[kougaihin - 1] ⑤ Nissu[tantou - 1]

穴埋めしてみる

```
Nissu = [4, 1, 3, 1, 3, 4, 2, 4, 3]
kougaihin = 9
Akibi = [1, 1, 1]
buinsu = 3
kougaihin を 1 から kougaihin まで 1 ずつ増やしながら繰り返す:
    tantou = 1
    buin を 2 から buinsu まで 1 ずつ増やしながら繰り返す:
        もし Akibi[buin] < Akibi[tantou] ならば:
            tantou = buin
    表示する(
        "工芸品", kougaihin, , "...",
        "部員", tantou, ":",
        Akibi[tantou], "日目～",
        Akibi[tantou] + Nissu[kougaihin] - 1, "日目"
    )
    Akibi[tantou] = Akibi[tantou] + Nissu[kougaihin]
```

Python で書き換えてみる

```
Nissu = [4, 1, 3, 1, 3, 4, 2, 4, 3]
kougeihinsu = 9
Akibi = [1, 1, 1]
buinsu = 3
for kougeihin in range(1, kougeihinsu + 1):
    tantou = 1
    for buin in range(2, buinsu + 1):
        if Akibi[buin - 1] < Akibi[tantou - 1]:
            tantou = buin
    print(
        "工芸品", kougeihin, , "...",
        "部員", tantou, ":",
        Akibi[tantou - 1], "日目～",
        Akibi[tantou - 1] + Nissu[kougeihin - 1] - 1, "日目"
    )
    Akibi[tantou - 1] = Akibi[tantou - 1] + Nissu[kougeihin - 1]
```

DNCL2 vs Python🔪

DNCL2

```
kougeihin を 1 から kougeihinsu まで 1 ずつ増やしながら繰り返す:  
    tantou = 1  
    buin を 2 から buinsu まで 1 ずつ増やしながら繰り返す:  
        もし Akibi[buin] < Akibi[tantou] ならば:  
            tantou = buin
```

Python

```
for kougeihin in range(1, kougeihinsu + 1):  
    tantou = 1  
    for buin in range(2, buinsu + 1):  
        if Akibi[buin - 1] < Akibi[tantou - 1]:  
            tantou = buin
```

DNCL2 の言語仕様は問題によって変わるかも 🤔

zero-based indexing (試作問題)

配列 **Kouka** に硬貨の額を低い順に設定している。なお、配列の添字は 0 から始まるものとする。最低額の硬貨が 1 円玉なので **Kouka[0]** の値は 1 となる。

one-based indexing (令和 7 年度本試験)

最も早く空きになる部員の番号を求めるために、各部員が空きになる日付を管理する配列 **Akibi** を用意する。この配列の添字(1 から始まる。)は部員の番号であり、要素はその部員が空きになる日付である。

大丈夫！怖くない！ 🙏

考え方は普段と同じ！

普段 Python を書いている人が C# を書くくらいの違い。
頭の使い方は変わらない！

問題をよく読もう！

どんな条件でどんなことをする問題なのか、しっかり読み取ろう！
数学でも国語でも大事！

まとめ

- DNCL は共通テストで 사용되는言語
- Python Like な 2 系が使われそう
- 配列の indexing が問題によって異なるかも

※鎌田の感想を含みます ⚠

第一志望は、ゆずれない。