

- ファイル名のXXXXXXXXXのところは学籍番号に変更してください
- プログラム中にstudent_numberに自分の学籍番号を入れてください

```
student_number = XXXXXXXXX
```

演習1：カレンダー計算

3つの入力、y（年）、m（月）、d（日）を取り、曜日を計算するプログラムを作ります。曜日は0を日曜日、1を月曜日、...とします。計算には、以下のグレゴリオ暦の公式を用いてください。なお、÷記号は整数除算です。

$$\begin{aligned}y_0 &= y - (14 - m) \div 12 \\x &= y_0 + y_0 \div 4 - y_0 \div 100 + y_0 \div 400 \\m_0 &= m + 12 \times ((14 - m) \div 12) - 2 \\d_0 &= (d + x + 31 \times m_0 \div 12) \bmod 7\end{aligned}$$

例えば、2000/2/14は以下より月曜日とわかります。

```
y0 = 2000 - 1 = 1999
x = 1999 + 1999/4 - 1999/100 + 1999/400 = 2483
m0 = 2 + 12*1 - 2 = 12
d0 = (14 + 2483 + (31*12) / 12) mod 7 = 2528 mod 7 = 1 （月曜日）
```

テストケース：

```
y: 2022
m: 4
d: 19
2
```

解答例

```
y = int(input("y: "))
m = int(input("m: "))
d = int(input("d: "))

y0 = ?
x = ?
m0 = ?
d0 = ?
```

```
print(d0)
```

```
y: 2022  
m: 4  
d: 19  
2
```

演習 2 : 3 項目ソート

3つの整数を入力にとり、昇順に並び替えて出力するプログラムを作ってください。ビルトイン関数の`max()`及び`min()`を用いてもよいです（例えば、`x = max(2,4,7)`のとき、`x`には7が代入されます）。

解答例

```
a = int(input(""))  
b = int(input(""))  
c = int(input(""))  
sorted = None # Do not use sorted  
  
### 以下で、max_val, med_val, min_valを求めてください  
max_val = ...  
  
print(min_val, med_val, max_val)  
  
4  
8  
3  
3 4 8
```

演習 3 : 半加算器

半加算器の出力sとcを計算するプログラムを書いてください。

解答例

```
a = True  
b = False  
  
s = ? # a XOR b  
c = ? # a AND b  
print (s, c)
```

演習4（オプション演習）：夢の住宅購入

慶應義塾を卒業後、とても良い仕事に就けたあなたは、住宅購入を考えています。周知の通り、東京23区のマンション価格は年々高騰しています。そこであなたは、住宅ローンの頭金を支払うために、数年間働き貯金をしなければならないことに気づきました。この課題では、以下を想定し、頭金を貯めるためにどれだけの時間がかかるか計算してみます。

1. 住宅価格を、`total_cost` と定義します。
2. 頭金の割合を、`portion_down_payment` と定義し、今回は0.25(25%)とします。
3. 現在の貯蓄額を `current_saving` と定義します。初期値は0円です。
4. あなたは貯蓄を賢く運用することができ、年換算利回り r に相当する運用益を毎月受け取ることができます。言い換えると、月末に `current_saving * r / 12` 分貯蓄を増やすことができます（ r が年利なので12で割っています）。今回は、 $r = 0.04$ (4%) とします。
5. 手取り年収を `annual_salary` と定義します。
6. あなたは、手取り月収のうち、`portion_saved` を毎月貯蓄に回します。この値は小数(10%の場合0.1)です。
7. 月末、あなたの貯蓄額は、手取り月収(つまり、手取り年収/12)と運用益の分増えます。
8. 12ヶ月ごとに、あなたの手取り年収は `annual_raise` 倍になります。今回は、1.1倍とします。

何ヶ月で頭金が貯蓄できるか計算するプログラムを書いてください。主な変数はfloat型なので、適宜入力をfloat型に変換してください。

実行時、プログラムは以下の入力を求め、()内の変数に代入します。

1. 手取り年収(`annual_salary`)
2. 貯蓄割合(`portion_saved`)
3. 住宅価格(`total_cost`)

ヒント：

- 入力は `input()` で取得してください。この問題では、ユーザーは正しい型の入力をするものと仮定してよいです。（int型が期待される変数に、'abc'などstring型の入力がされるなどといったことは想定しなくてOK）
- いくつか状態を記憶する変数を定義し、適切に初期化する必要があります。年単位のものと月単位のものに注意してください。
- `while` で繰り返し処理（ループ）します。
- 昇給のタイミングに注意してください。昇給されるのは、12ヶ月目、24ヶ月目、...の「次の」月からです。

テストケース：

```
Enter your starting annual salary: 5000000
Enter the percent of your salary to save, as a decimal: 0.1
Enter the cost of your dream home: 120000000
Number of months: 216
```

```
Enter your starting annual salary: 8000000
Enter the percent of your salary to save, as a decimal: 0.15
Enter the cost of your dream home: 50000000
Number of months: 83
```

解答例

```
## 入力を求めて型変換
?!=? input('Enter your starting annual salary: ')
?!=? input('Enter the percent of your salary to save, as a decimal: ')
?!=? input('Enter the cost of your dream home: ')

## ここで計算

print('Number of months: {}'.format(???)
```

```
Enter your starting annual salary: 8000000
Enter the percent of your salary to save, as a decimal: 0.15
Enter the cost of your dream home: 50000000
Number of months: 83
```

時間つぶし用の演習：ドラゴン曲線

この課題は採点しません。提出しないでください。解けた人には秘密の果物が見えます。

問題：6次の[ドラゴン曲線](#)の書き方を示すプログラムを書いてください。前進を"F"、左回転を"L"、右回転を"R"で表します。

例：0次から3次までのドラゴン曲線は以下のとおりです。

```
F
FLF
FLFLFRF
FLFLFRFLFLFRFRF
```

解答例と検証用コード

```
def check_answer(x):
    import hashlib
    cypher = 137520883295039644377926921405345508733
    key = hashlib.md5(bytes(x, 'utf-8')).hexdigest()
    decrypted_hex = cypher ^ int(key, 16)
    try:
        return bytes.fromhex(hex(decrypted_hex)[2:]).decode()
    except:
        return "Decoding error"

### ここで文字列を生成
...
dragon6 = ???

print(check_answer(dragon6))
```