

レポート用紙

講義名 : 数値解析 1	年月日 : 2024 年 4 月 29 日(月)
学籍番号 : 99999999	氏名 : 幸谷 智紀

本日の課題

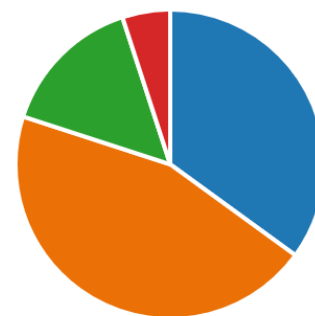
1. テキスト P.15 問題 2.4
2. テキスト P.16 問題 2.5
3. テキスト P.17 問題 2.6

感想

2. 課題の難易度はどうでしたか？

[詳細](#)

● 難しかった	7
● 少し難しかった	9
● ちょうど良かった	3
● 簡単だった	1
● とても簡単だった	0



1. テキスト P.15 問題 2.4

スクリプト例 : output_complex.py

```
# output_complex.py: 問題 2.4 複素数の書式付き出力
str_a = input('Real(c) = ? ')
str_b = input('Imag(c) = ? ')

# 文字列→複素数型に変換
a = float(str_a)
b = float(str_b)

# c := a + b * i
c = complex(a, b)

# c の表示
```

レポート用紙

```

print('c = ', c, type(c))
# %書式指定演算子
print('c = (%25.17e) + (%25.17e) * i' % (c.real, c.imag))
print('c = (%25.17f) + (%25.17f) * i' % (c.real, c.imag))
print('c = (%25.17g) + (%25.17g) * i' % (c.real, c.imag))
# format 文
print('c = ({:25.17e}) + ({:25.17e}) * i'.format(c.real, c.imag))
print('c = ({:25.17f}) + ({:25.17f}) * i'.format(c.real, c.imag))
print('c = ({:25.17g}) + ({:25.17g}) * i'.format(c.real, c.imag))
# f 文字列
print(f'c = ({c.real:25.17e}) + ({c.imag:25.17e}) * i')
print(f'c = ({c.real:25.17f}) + ({c.imag:25.17f}) * i')
print(f'c = ({c.real:25.17g}) + ({c.imag:25.17g}) * i')

```

実行例：

```

PS C:\Users\tkouy\OneDrive - 静岡理科大学\講義用ドキュメント\数値解析1\2024samples> python output_complex.py
Real(c) = ? -3
Imag(c) = ? 4
c = (-3+4j) <class 'complex'>
c = ( -3.0000000000000000e+00) + ( 4.0000000000000000e+00) * i
c = ( -3.0000000000000000) + ( 4.0000000000000000) * i
c = ( -3) + ( 4) * i
c = ( -3.0000000000000000e+00) + ( 4.0000000000000000e+00) * i
c = ( -3.0000000000000000) + ( 4.0000000000000000) * i
c = ( -3) + ( 4) * i
c = ( -3.0000000000000000e+00) + ( 4.0000000000000000e+00) * i
c = ( -3.0000000000000000) + ( 4.0000000000000000) * i
c = ( -3) + ( 4) * i

```

2. テキスト P.16 問題 2.5

スクリプト例: calc_formu.apy

```

# calc_formula.py: 問題 2.5
from cmath import * # sqrt,exp,log # 使用する関数

# 1. exp((-4/3)^3)
print('exp((-4/3) ** 3) = ', exp((-4/3) ** 3))
# 2. sqrt(-5)
print('sqrt(-5) = ', sqrt(-5))
# 3. exp(log(-10))
print('exp(log(-10)) = ', exp(log(-10)))
# 4. i^i
print('(1j) ** (1j) = ', (1j) ** (1j))

```

レポート用紙

実行例：

```
exp((-4/3) ** 3) = (0.0934461101976254+0j)
sqrt(-5) = 2.23606797749979j
exp(log(-10)) = (-10.0000000000000002+1.2246467991473535e-15j)
(1j) ** (1j) = (0.20787957635076193+0j)
```

3. テキスト P.17 問題 2.6

quadratic_eq_complete.py: 2次方程式を解く(複素解対応版)

import cmath # 複素数用 sqrt 関数

複素係数入力

re_a = input('Re a =? ')

im_a = input('Im a =? ')

re_b = input('Re b =? ')

im_b = input('Im b =? ')

re_c = input('Re c =? ')

im_c = input('Im c =? ')

a = complex(float(re_a), float(im_a))

b = complex(float(re_b), float(im_b))

c = complex(float(re_c), float(im_c))

print(f' {a:25.17e} * x^2')

print(f'+ {b:25.17e} * x')

print(f'+ {c:25.17e} = 0')

a == 0?

if (a.real == 0) and (a.imag == 0):

b == 0?

if (b.real == 0) and (b.imag == 0):

if (c.real == 0) and (c.imag == 0):

print('xは任意。')

else:

print('c != 0 -> 方程式になっていません。')

b != 0 -> b * x + c = 0 -> x = -c / b

else:

x = -c / b

print('1次方程式 -> x = ', x)

print('検算: b * x + c = 0 ? -> ', b * x + c)

レポート用紙

2次方程式の解の計算

else:

```
d = b ** 2 - 4.0 * a * c
```

```
print('complex solutions: %n')
```

```
x1 = (-b + cmath.sqrt(d)) / (2 * a)
```

```
x2 = (-b - cmath.sqrt(d)) / (2 * a)
```

出力

```
print(f'x1 = {x1:25.17e}')
```

```
print(f'x2 = {x2:25.17e}')
```

検算

```
print('a * x1^2 + b * x1 + c = 0? -> ', a * x1**2 + b * x1 + c)
```

```
print('a * x2^2 + b * x2 + c = 0? -> ', a * x2**2 + b * x2 + c)
```

実行例 :

```
Re a =? -3
Im a =? 1
Re b =? 1
Im b =? 4
Re c =? 4
Im c =? 1
-3.0000000000000000e+00+1.0000000000000000e+00j * x^2
+ 1.0000000000000000e+00+4.0000000000000000e+00j * x
+ 4.0000000000000000e+00+1.0000000000000000e+00j = 0
complex solutions:
x1 = -9.47326502254452252e-01+2.96171154866918918e-01j
x2 = 8.47326502254452385e-01+1.00382884513308102e+00j
a * x1^2 + b * x1 + c = 0? -> (1.3322676295501878e-15+4.440892098500626e-16j)
a * x2^2 + b * x2 + c = 0? -> (4.440892098500626e-16+8.881784197001252e-16j)
```

以上