

## PS0001 AY2324 S1 Coding CA2 Questions

### Qn 1

Write a Python program that can compute the following sequence:

$$F(0) = 5$$

$$F(1) = 18$$

$$F(n) = F(n-1) - 2 \cdot F(n-2)$$

- Submit on NTU Learn the value of  $F(30)$

You can check that your program is working well, by verifying that  $F(6) = 100$

### Qn 2

- Write in the answer box the value of  $F(60)$

### Qn 3

One can approximate the value of  $\pi$  using the following formula:

$$\pi = \sum_{k=0}^{\infty} \frac{1}{16^k} \left( \frac{4}{8k+1} - \frac{2}{8k+4} - \frac{1}{8k+5} - \frac{1}{8k+6} \right)$$

Using this formula, deduce (and submit on NTU Learn) the smallest value  $k$  that is sufficient to obtain an approximation of  $\pi$  of at most  $0.00000000000001$  (i.e.  $10^{-14}$ ). You can obtain the real value of  $\pi$  using the Python `math` library.

### Qn 4

Given the following  $(2 \times 2)$  square matrix  $M = \begin{pmatrix} 2 & -1 \\ 3 & 1 \end{pmatrix}$ , compute the value of  $M^{29}$ . Submit on NTU Learn the top-left value of the matrix  $M^{29}$ .

We recall that  $\begin{pmatrix} a & b \\ c & d \end{pmatrix} * \begin{pmatrix} x & y \\ z & w \end{pmatrix} = \begin{pmatrix} ax + bz & ay + bw \\ cx + dz & cy + dw \end{pmatrix}$

You can check that your program is working well, by verifying that  $M^2 = \begin{pmatrix} 1 & -3 \\ 9 & -2 \end{pmatrix}$ .

## Solutions:

```
# Qn 1
def func(n):
    if n == 0:
        return 5
    elif n == 1:
        return 18
    return func(n-1) - 2*func(n-2)

print(func(30)) # Ans: -354140
```

```
# Qn 2
def func_iter(n):
    i = 2
    a, b = 5, 18
    while i < n+1:
        c = b - 2*a
        a, b = b, c
        i += 1
    return c

print(func_iter(60)) # Ans: -8941627580
```

```
# Qn 3
import math
def series():
    diff = 10**-14
    k = 0
    sum = 0
    while True:
        sum += (1/16**k) * (4/(8*k+1) - 2/(8*k+4) - 1/(8*k+5) - 1/(8*k+6))
        if (math.pi - sum) < diff:
            break
        k += 1
    return k

print(series()) # Ans: 9
```

```
# Qn 4
def matrix():
    b11, b12, b21, b22 = 2, -1, 3, 1
    nb11, nb12, nb21, nb22 = 2, -1, 3, 1

    # Perform matrix multiplication
    for _ in range(28):
        c11 = nb11 * b11 + nb12 * b21
        c12 = nb11 * b12 + nb12 * b22
        c21 = nb21 * b11 + nb22 * b21
        c22 = nb21 * b12 + nb22 * b22

        # Update new matrix values
        nb11, nb12, nb21, nb22 = c11, c12, c21, c22

    return nb11

print(matrix()) # Ans: 5210330357
```