**Lab 02**

|  |  |
| --- | --- |
| Name: | 劉邦彬 |
| Student ID: | R11631033 |
| Total Score: |  |

**Note:**

Most of the explanations in this lab is optional (except for question C2). However, giving reasonable explanations to your answer or programs will earn you partial credits when your answer is incorrect.

1. **Multiple Choice (24 points, 6 points each question)**

|  |  |  |  |
| --- | --- | --- | --- |
| # | Answer | Explanation (Optional) | Score |
| 1 | (d) |  |  |
| 2 | (a) |  |  |
| 3 | (c) |  |  |
| 4 | (b) |  |  |

1. **Reading and Programming (24 points, 6 points each question)**

|  |  |  |
| --- | --- | --- |
| # | Explanation (Optional) | Score |
| 1 |  |  |
| 2 |  |  |
| 3 |  |  |
| 4 | Output: |  |

1. **Spot the Bug (26 points, 10 points + 16 points)**

|  |  |  |
| --- | --- | --- |
| # | Explanation (Mandatory for question 2) | Score |
| 1 |  |  |
| 2 | Output: |  |

1. **MyEnumerate (26 points)**

|  |  |  |
| --- | --- | --- |
| # | Explanation (Optional) | Score |
| 1 | Output: |  |

**LAB02\_B**

#Lab02\_B.py

from datetime import datetime,timedelta

def n\_times\_day(bday1, bday2, n):

    for i in range(100\*365):

        c\_date = bday2 + timedelta(days=i)

        d1 = (c\_date - bday1).days

        d2 = (c\_date - bday2).days

        if d1 == n\*d2:

            result=(c\_date.strftime('%Y-%m-%d %H:%M:%S'))

    return result

def main():

    now = datetime.now()

    print("Lab03 B-1")

    print(now)

    day\_of\_the\_week = now.weekday()

    todays\_date = [

        "Monday",

        "Tuesday",

        "Wednesday",

        "Thursday",

        "Friday",

        "Saturday",

        "Sunday",

    ]

    print("Today's date and the day of the week:", todays\_date[day\_of\_the\_week])

    # TODO\_B1

    # Your output should be like:

    # 2020-08-03 20:19:19.806211

    # Monday

    print("Lab03 B-2")

    s = input("Enter your birthday in mm/dd/yyyy format: ")  #'1/15/1997'

    format\_string = "%m/%d/%Y "

    ur\_birthday = datetime.strptime(s, "%m/%d/%Y")

    if now.month >= ur\_birthday.month and now.day >= ur\_birthday.day:

        next\_birthday = datetime(now.year + 1, ur\_birthday.month, ur\_birthday.day)

        until\_nextbirth = next\_birthday - now

        print(until\_nextbirth)

        print(now.year - ur\_birthday.year)

    else:

        next\_birthday = datetime(now.year, ur\_birthday.month, ur\_birthday.day)

        until\_nextbirth = next\_birthday - now

        print(until\_nextbirth)

        print(now.year - ur\_birthday.year - 1)

    # TODO\_B2

    # Your output should be like:

    # 280 days, 3:40:40.193789

    # 22

    print("Lab03 B-3")

    print("For people born on these dates:")

    bday1 = datetime(day=15, month=1, year=1997)

    bday2 = datetime(day=11, month=10, year=2003)

    print("Double Day is",n\_times\_day(bday1, bday2, n=2))

    # TODO\_B3

    # Your output should be like:

    # 2020-01-01 00:00:00 (this is not the correct answer!)

    print("Lab03 B-4")

    print("Triple Day is ",n\_times\_day(bday1, bday2, n=3))

if \_\_name\_\_ == "\_\_main\_\_":

    main()

**Lab02\_C**

# Lab02\_C.py

class Array:

    """A Array is a self-defined sequential data type which contains a name."""

    def \_\_init\_\_(self, name, contents=None):

        """Initialize the contents.

        name: string

        contents: initial contents.

        """

        self.name = name

        if contents == None:

            contents = []

        self.contents = contents

    def \_\_str\_\_(self):

        """Return a string representaion of this Array."""

        t = [self.name + " has contents:"]

        for obj in self.contents:

            s = "    " + object.\_\_str\_\_(obj)

            t.append(s)

        return "\n".join(t)

    def \_\_getitem\_\_(self, key):

        return self.contents[key]

    def \_\_setitem\_\_(self, key, value):

        self.contents[key] = value

    # TODO\_C1

    # Hint: two special functions here

    def add\_item(self, item):

        """Adds a new item to the contents.

        item: any object to be added

        """

        self.contents.append(item)

    def \_\_repr\_\_(self):

        return f"Array('{self.name}', {self.contents})"

def main():

    arr1 = Array("MyFirstArray")

    arr1.add\_item("wallet")

    arr1.add\_item("car keys")

    print(arr1)

    # you shouldsee the output:

    # MyFirstArray has contents:

    #   'wallet'

    #   'car keys'

    # The following 3 statements should work

    # after you finish TODO\_C1

    print(arr1[0])  # 'wallet'

    arr1[0] = "toys"

    print(arr1[0])  # 'toys'

    arr2 = Array("MySecondArray")

    arr1.add\_item(arr2)

    print(arr2)

    # If you run this program as is, it seems to work.

    # To see the problem, trying printing arr2.

if \_\_name\_\_ == "\_\_main\_\_":

    main()

**Lab02\_D**

# Lab02\_D.py

class MyEnumerate:

    def \_\_init\_\_(self, data, label):

        pass  # TODO\_D

        self.data = data

        self.label = label

        self.index = 0

    def \_\_iter\_\_(self):

        pass  # TODO\_D

        return self

    def \_\_next\_\_(self):

        pass  # TODO\_D

        try:

            index = self.index

            info = self.data[index]

            target = self.label[index]

            self.index += 1

            return index, info, target

        except IndexError:

            raise StopIteration

def main():

    data = [[174, 63], [165, 45], [168, 61], [180, 85], [163, 52]]

    label = ["male", "female", "male", "male", "female"]

    for index, info, target in MyEnumerate(data, label):

        print(f"id:{index} | height:{info[0]} weight:{info[1]} -> {target}")

    # the output should look like the following:

    # id:0 | height:174 weight:63 -> male

    # id:1 | height:165 weight:45 -> female

    # id:2 | height:168 weight:61 -> male

    # id:3 | height:180 weight:85 -> male

    # id:4 | height:163 weight:52 -> female

if \_\_name\_\_ == "\_\_main\_\_":

    main()