

# 2017년 2학기 공업수학 중간고사

이름:

학번:

점수:

/ 100점 만점

문제 1	문제 2	문제 3	문제 4	문제 5	문제 6	문제 7	문제 8	문제 9	문제 10	합 계

## 시험에서 사용하는 모듈 임포트 하기

```
import matplotlib.pyplot as plt  
import numpy as np
```

## 파이썬 기본 자료형

### 문제 1

실수(부동소수점)를 하나 입력받아, 그 숫자를 반지름으로 하는 원의 면적과 둘레의 길이를 튜플로 리턴하는 함수 `circle_radius`를 구현하는 코드를 작성하라,

전본강안 :

```
import math.  
def circle_radius(r):  
    area = math.pi * (r ** 2)  
    perimeter = 2 * math.pi * r  
    return (area, perimeter)
```

### 문자열 자료형

주의: `math.pi`  
대신에 3.14  
사용 가능

아래 사이트는 커피 콩의 현재 시세를 보여준다.

<http://beans-r-us.appspot.com/prices.html>

위 사이트의 내용을 html 소스코드로 보면 다음과 같으며, 검색된 시간의 커피콩의 가격은 Current price of coffee beans 문장이 담겨 있는 줄에 명시되어 있다.

```
<html><head><title>Welcome to the Beans'R'Us Pricing Page</title>  
<link rel="stylesheet" type="text/css" href="beansrus.css" />  
</head><body>  
<h2>Welcome to the Beans'R'Us Pricing Page</h2>  
<p>Current price of coffee beans = <strong>$5.94</strong></p>  
<p>Price valid for 15 minutes from Sun Sep 10 12:21:58 2017.</p>  
</body></html>
```

위 소스코드를 웹페이지의 주소만 알고 있을 때 아래와 같이 읽어들일 수 있다.

```
page = urllib2.urlopen("http://beans-r-us.appspot.com/prices.html")
text = page.read().decode("utf8")
```

위 코드를 실행하면 `text` 변수에 웹페이지의 전체 내용이 하나의 문자열로 저장된다.

## 문제 2

`text`에 저장된 문자열에서 가격 정보를 확인하는 방법을 코드를 이용하여 설명하라.

전통답안 :       $\text{price\_index} = \text{text}.find(>\$) + 2$   
 $\text{price} = \text{text}[\text{price\_index}, \text{price\_index} + 4]$

## 사전

`record_list.txt` 파일은 여덟 명의 수영 선수의 50m 기록을 담고 있다.

```
player1 21.09
player2 20.32
player3 21.81
player4 22.97
player5 23.29
player6 22.09
player7 21.20
player8 22.16
```

## 문제 3

아래코드가 하는 일을 설명하라.

```
record_f = open("record_list.txt", 'r')
record = record_f.read().decode('utf8').split('\n')

record_dict = {}

for line in record:
    (player, p_record) = line.split()
    record_dict[p_record] = player

record_f.close()

record_list = record_dict.keys()
record_list.sort()
```

전통답안 :

한글원 파일로부터 각각의 줄에 포함되어 있는 선수이름과 기록을 추출하여, 기록을 키로, 선수 이름을 카드으로하여 `record_dict` 사전 변수에 저장한다. 이후 기록 순서를 확인하기 위해 키들만의 리스트를 정렬한다.

## 튜플

### 문제 4

아래 설명 중에서 튜플 자료형의 성질에 해당하는 항목을 모두 골라라.

1. 가변 자료형이다.
2. 불변 자료형이다.
3. 인덱스와 슬라이싱을 활용하여 항목의 내용을 확인하고 활용할 수 있다.
4. 항목들이 임의의 자료형을 가질 수 있다.
5. 튜플 길이에 제한이 있다.
6. 진성정보 등 중요한 데이터를 보관할 때 사용한다.

## 리스트 조건제시법

아래 코드는 0부터 1000 사이의 홀수들의 제곱의 리스트를 조건제시법으로 생성한다

```
odd_1000 = [x**2 for x in range(0, 1000) if x % 2 == 1]
```

### 문제 5

0부터 1000까지의 숫자들 중에서 홀수이면서 7의 배수인 숫자들의 리스트를 조건제시법으로 생성하는 코드를 작성하라.

결론답안 :

$\text{odd\_1000\_7} = [x \text{ for } x \text{ in range}(0, 1000) \text{ if } x \% 2 == 1 \text{ and } x \% 7 == 0]$

설명

$\text{odd\_1000\_7} = [7 * x \text{ for } x \text{ in range}(0, 142) \text{ if } x \% 2 == 1]$

주의!

### 문제 6

0부터 1000까지의 숫자들 중에서 홀수이면서 7의 배수인 숫자들을 제곱하여 1을 더한 값들의 리스트를 조건제시법으로 생성하는 코드를 작성하라. 힌트: 아래와 같이 정의된 함수를 활용한다.

정의 :

$$f(x) = x^2 + 1$$

def f(x):

return  $x^2 + 1$

$\text{odd\_1000\_7\_f} = [f(x) \text{ for } x \text{ in odd\_1000\_7}]$

## csv 파일 읽어들이기

'Seoul\_pop2.csv' 파일에는 아래 내용이 저장되어 있다.

```
### 1949년부터 2010년 사이의 서울과 수도권 인구 증가율(%)  
# 구간,서울,수도권  
  
1949-1955,9.12,-5.83  
1955-1960,55.88,32.22  
1960-1966,55.12,32.76  
1966-1970,45.66,28.76  
1970-1975,24.51,22.93  
1975-1980,21.38,21.69  
1980-1985,15.27,18.99  
1985-1990,10.15,17.53  
1990-1995,-3.64,8.54  
1995-2000,-3.55,5.45  
2000-2005,-0.93,6.41  
2005-2010,-1.34,3.71
```

학장자가 csv인 파일은 데이터를 저장하기 위해 주로 사용한다. csv는 Comma-Separated Values의 줄임말로 데이터가 쉼표(콤마)로 구분되어 정리되어 있는 파일을 의미한다. csv 파일을 읽어드리는 방법은 csv 모듈의 `reader()` 함수를 활용하면 매우 쉽다. `reader()` 함수의 리턴값은 csv 파일에 저장된 내용을 줄 단위로, 쉼표 단위로 끊어서 2차원 리스트이다. 예를 들어, 아래 코드는 언급된 파일에 저장된 내용의 각 줄을 출력해준다.

```
import csv  
with open('Seoul_pop2.csv', 'rb') as f:  
    reader = csv.reader(f)  
    for row in reader:  
        if len(row) == 0 or row[0][0] == '#':  
            continue  
        else:  
            print(row)
```

### 문제 7

위 코드에서 5번 째 줄을 아래와 같이 하면 오류 발생한다.

```
if row[0][0] == '#' or len(row) == 0:
```

이유를 간단하게 설명하라.

전체당연: A or B를 실행할 때,  
A에서 오류가 발생하면 바로  
멈춘다. 그런데, 자료 파일의  
셋째 줄이 비었으나.  
따라서 row[0] 부분에서  
오류가 발생된다.  
따라서 반복문을 빠지는  
`len(row)==0` 을 먼저 실행  
하도록 해야 한다.

## 넘파이 활용 기초 1

넘파이 어레이를 생성하는 방법은 몇 개의 기본적인 함수를 이용하면 된다.

- np.arange()
- np.zeros()
- np.ones()
- np.diag()

### 문제 8

아래 모양의 어레이를 생성하는 코드를 작성하라. 단, 언급된 네 개의 함수들만 사용해야 하며, 수동으로 생성된 리스트나 어레이에는 허용되지 않는다.

$$\begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 2 \end{bmatrix}$$

진짜방법:

np.ones(3) \* 2

## 넘파이 활용 기초 2

population.txt 파일은 1900년부터 1920년까지 캐나다 북부지역에서 서식한 산토끼(hare)와 스라소니(lynx)의 숫자, 그리고 채소인 당근(carrot)의 재배숫자를 아래 내용으로 순수 텍스트 데이터로 담고 있다.

#	year	hare	lynx	carrot
1900		30e3	4e3	48300
1901		47.2e3	6.1e3	48200
1902		70.2e3	9.8e3	41500
1903		77.4e3	35.2e3	38200
1904		36.3e3	59.4e3	40600
1905		20.6e3	41.7e3	39800
1906		18.1e3	19e3	38600
1907		21.4e3	13e3	42300
1908		22e3	8.3e3	44500
1909		25.4e3	9.1e3	42100
1910		27.1e3	7.4e3	46000
1911		40.3e3	8e3	46800
1912		57e3	12.3e3	43800
1913		76.6e3	19.5e3	40900
1914		52.3e3	45.7e3	39400
1915		19.5e3	51.1e3	39000
1916		11.2e3	29.7e3	36700
1917		7.6e3	15.8e3	41800
1918		14.6e3	9.7e3	43300
1919		16.2e3	10.1e3	41300
1920		24.7e3	8.6e3	47300

아래 코드는 연도, 토끼 개체수, 스라소리 개체수, 당근 개체수를 따로따로 떼어 내어 각각 어레이로 변환하여 year, hares, lynxes, carrots 변수에 저장하는 코드이다.

```
data = np.loadtxt('populations.txt')
year, hares, lynxes, carrots = data.T
```

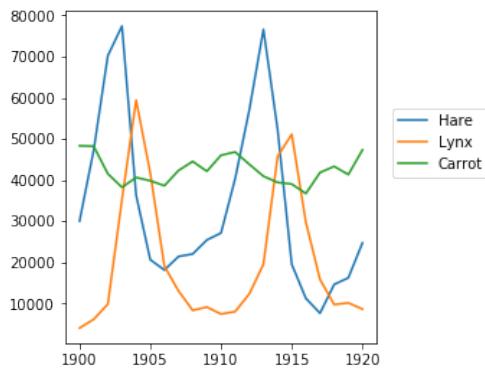
### 문제 9

위 코드에서 `data.T`에 대해 간단하게 설명하라.

전달방법 : `data.T`는 `data` 행렬의 전치행렬을 나타낸다. 따라서 `axis=0` 모양이며, 각 줄은 차례대로 연도, 토끼 개체수, 스라소리 개체수, 당근 개체수를 연도별로 가지며, 액축이 `year, hares, lynxes, carrots` 순서에 저장된다.

아래 코드는 토끼, 스라소니, 당근 각각의 개체수의 연도별 변화를 선그래프로 보여주도록 하는 코드이다.

```
plt.axes([0.2, 0.1, 0.5, 0.8])
plt.plot(year, hares, year, lynxes, year, carrots)
plt.legend(['Hare', 'Lynx', 'Carrot'], loc=(1.05, 0.5))
```



### 문제 10

산토끼, 스라소니, 당근의 예제에서 1900년부터 1921년 사이에 개체별 개체수의 변화에 대해 어떤 분석을 할 수 있는지 그레프를 이용하여 간단하게 설명하라.

전달방법 : 산토끼, 스라소니, 당근의 개체수가 서로에게 억제하게 영향을 준다. 예를 들어, 산토끼 수가 늘면 스라소니 개체수가 느는 만연해지며 당근 개체수는 줄어든다. 그러나 당근개체수가 많이 줄어지거나 스라소니 개체수가 많아지면 토끼 개체수가 확 줄어들며, 따라서 스라소니 개체수도 확 줄어든다. 반면에 당근 개체수는 다시 늘어나고 일정 시간이 지나 산토끼 개체수가 따라서 늘어난다.