

2차 프로그래밍 과제

과목: 인공지능프로그래밍언어기초

학번: 201821498

이름: 신재현

학과: e-비즈니스학과

문제 1

- 다음 조건을 만족하는 넌센스 퀴즈 프로그램을 작성하시오.<조건>

1. 넌센스 퀴즈 3개를 랜덤하게 출제
2. 딕셔너리 자료구조 사용 (key: 문제번호)
3. 문제당 1점 처리하여 최종 점수를 출력
4. 총 2번 실행한 결과 제출

백가지 과일이 죽기 직전을 다른 말로?

정답(또는 quit): 백과사전

정답

깨뜨리고 칭찬 받는 것은?

정답(또는 quit): 접시

오답

못 사온다고 해놓고 사온 것은?

정답(또는 quit): 못

정답

최종 점수는 2

```
In [2]: # 랜덤 함수  
import random
```

```
In [3]: # 딕셔너리로 넌센스 퀴즈 생성  
prob1 = {"key": 0, "prob": "백가지 과일이 죽기 직전을 다른 말로?", "ans": "백과사전"}
```

```

prob2 = {"key": 1, "prob": "깨뜨리고 칭찬 받는 것은?", "ans": "신기록"}

prob3 = {"key": 2, "prob": "못 사온다고 해놓고 사온 것은?", "ans": "못"}

prob4 = {"key": 3, "prob": "병아리가 제일 잘 먹는 약은?", "ans": "뼈약"}

prob5 = {"key": 4, "prob": "개 중에 가장 아름다운 개는?", "ans": "무지개"}

prob6 = {"key": 5, "prob": "걱정이 많은 사람이 오르는 산은?", "ans": "태산"}

prob7 = {"key": 6, "prob": "다리 중 아무도 보지 못한 다리는?", "ans": "헛다리"}

problem = [prob1, prob2, prob3, prob4, prob5, prob6, prob7]

```

In [4]:

```

# 넌센스 문제 출력 함수
# (문제번호)

def nonsense():
    global score
    key = random.randrange(0,7)
    prob = problem[key]
    print(prob["prob"])
    ans = input("정답(또는 quit): ")
    if ans == "quit":
        print("quit")
    else:
        if ans == prob["ans"]:
            print("정답")
            score += 1
        else:
            print("오답, 정답은 {}입니다.".format(prob["ans"]))

```

In [5]:

```

score = 0

for _ in range(3):
    nonsense()

print(f"최종점수는: {score}")

```

깨뜨리고 칭찬 받는 것은?
 정답
 깨뜨리고 칭찬 받는 것은?
 오답, 정답은 신기록입니다.
 개 중에 가장 아름다운 개는?
 정답
 최종점수는: 2

In [6]:

```

# 랜덤함수
import random
# 딕셔너리로 넌센스 퀴즈 생성
prob1 = {"key": 0, "prob": "백가지 과일이 죽기 직전을 다른 말로?", "ans": "백과사전"}

prob2 = {"key": 1, "prob": "깨뜨리고 칭찬 받는 것은?", "ans": "신기록"}

prob3 = {"key": 2, "prob": "못 사온다고 해놓고 사온 것은?", "ans": "못"}

prob4 = {"key": 3, "prob": "병아리가 제일 잘 먹는 약은?", "ans": "뼈약"}

prob5 = {"key": 4, "prob": "개 중에 가장 아름다운 개는?", "ans": "무지개"}

prob6 = {"key": 5, "prob": "걱정이 많은 사람이 오르는 산은?", "ans": "태산"}

```

```

prob7 = {"key":6, "prob":"다리 중 아무도 보지 못한 다리는?", "ans":"헛다리"}

problem = [prob1, prob2, prob3, prob4, prob5, prob6, prob7]
# 넌센스 문제 출력 함수
# (문제번호)

def nonsense():
    global score
    key = random.randrange(0,7)
    prob = problem[key]
    print(prob["prob"])
    ans = input("정답(또는 quit): ")
    if ans == "quit":
        print("quit")
    else:
        if ans == prob["ans"]:
            print("정답")
            score += 1
        else:
            print("오답, 정답은 {}입니다.".format(prob["ans"]))

score = 0

for _ in range(3):
    nonsense()

print(f"최종점수는: {score}")

```

다리 중 아무도 보지 못한 다리는?
 정답
 걱정이 많은 사람이 오르는 산은?
 정답
 못 사온다고 해놓고 사온 것은?
 정답
 최종점수는: 3

문제2

- 다음 조건을 만족하는 프로그램을 작성하시오.<조건>

1. 파일(wlist.txt)로부터 줄 단위로 저장된 단어 읽기

- 단어는 영문자 소문자로만 구성되어야 함
- 정규식을 이용하여 대문자는 소문자로 변경, 특수문자[, @, \$, !]나 숫자는 제거
- 새 파일(newwlist.txt)에 줄 단위로 저장

2. 새 파일에 저장된 단어들 중에 단어의 개수가 7이하인 단어의 수를 출력

```

In [7]: import re
import sys

```

```

In [8]: # 파일 불러읽기
with open("dataWWWwlist.txt", "r") as f:
    example = f.readlines()
    lines = []
    for line in example:

```

```

ch_line = line.strip()
ch_line = ch_line.lower()
ch_line = re.sub('[0-9]+', '', ch_line)
ch_line = re.sub('_', '', ch_line)
ch_line = re.sub('@', '', ch_line)
ch_line = re.sub('W$', '', ch_line)
ch_line = re.sub('!', '', ch_line)
lines.append(ch_line)

f = open('dataWWnewwlist.txt', 'w')

for line in lines:
    print(line, file= f)

f.close()

```

문제 3

<조건>

- 다음의 데이터를 활용하여 아래 그래프와 동일하게 작성하라.

- 1번 데이터: x = 77, 78, 85, 83, 73, 77, 73, 80, y = 25, 28, 19, 30, 21, 22, 17, 35
- 2번 데이터: x = 75, 77, 86, 86, 79, 83, 83, 88, y = 56, 57, 50, 53, 60, 53, 49, 61
- 3번 데이터: x = 34, 38, 38, 41, 30, 37, 41, 35, y = 22, 25, 19, 30, 21, 24, 28, 18

```

In [9]: import matplotlib.pyplot as plt
import numpy as np

```

```

In [10]: # 데이터 만들기
xData_1 = np.array([77, 78, 85, 83, 73, 77, 73, 80])
yData_1 = np.array([25, 28, 19, 30, 21, 22, 17, 35])

xData_2 = np.array([75, 77, 86, 86, 79, 83, 83, 88])
yData_2 = np.array([56, 57, 50, 53, 60, 53, 49, 61])

xData_3 = np.array([34, 38, 38, 41, 30, 37, 41, 35])
yData_3 = np.array([22, 25, 19, 30, 21, 24, 28, 18])

```

```

In [ ]: # 1번 데이터로 그래프 그리기
plt.scatter(xData_1, yData_1, color="red", marker='o')
plt.title('Dachshund size')
plt.xlabel('Length')
plt.ylabel('Height')

plt.show()

# 2번 데이터로 그래프 그리기
plt.scatter(xData_2, yData_2, color="blue", marker='s')
plt.title('Samoyad size')
plt.xlabel('Length')
plt.ylabel('Height')

```

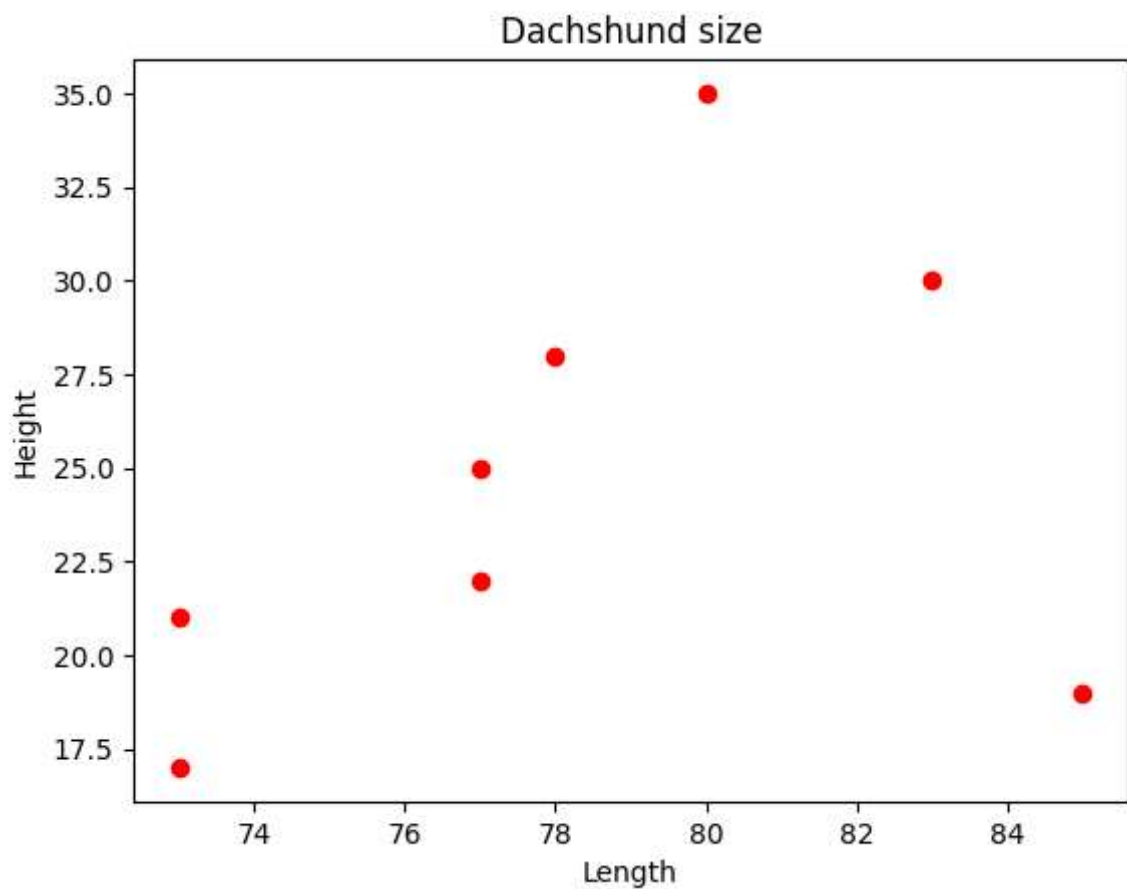
```
plt.show()

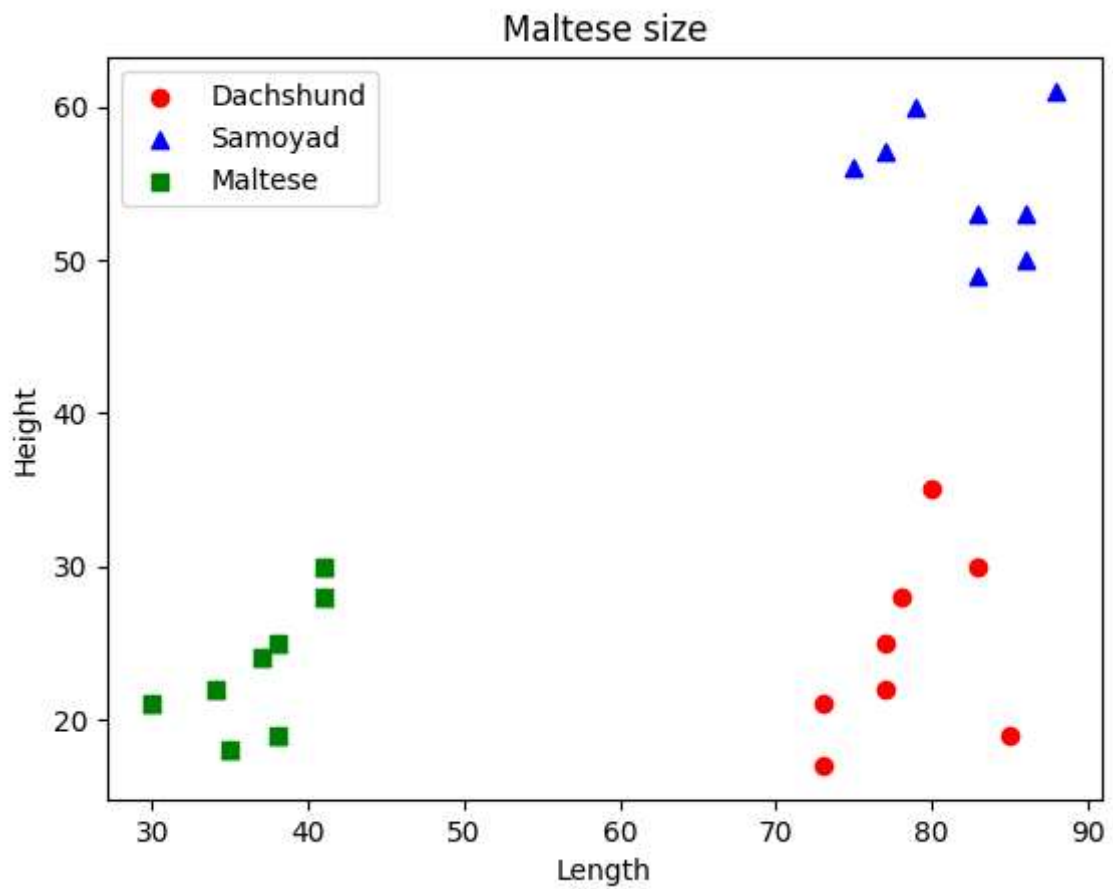
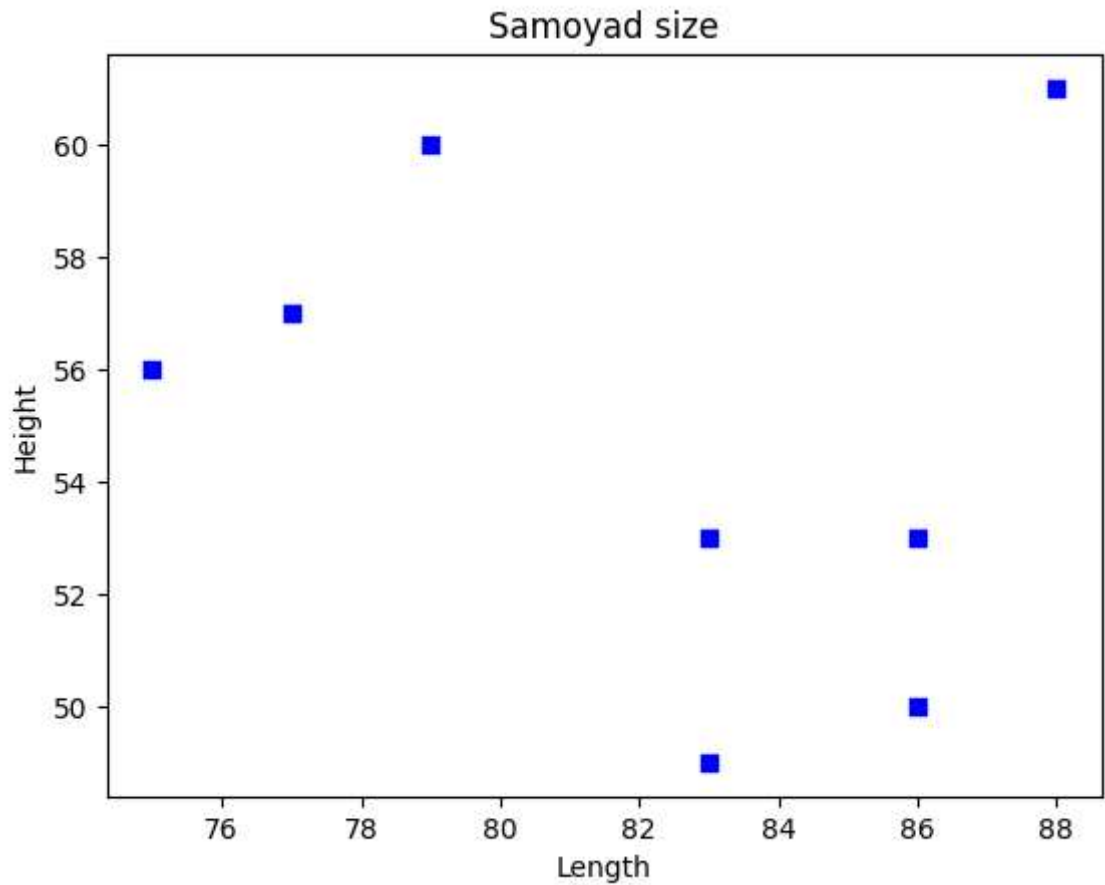
# 3번 데이터로 그래프 그리기
plt.scatter(xData_3, yData_3, color="green", marker="^")
plt.title('Maltese size')
plt.xlabel('Length')
plt.ylabel('Height')

# 총합 데이터로 그래프 그리기
plt.scatter(xData_1, yData_1, color="red", marker='o', label='Dachshund')
plt.scatter(xData_2, yData_2, color="blue", marker='^', label='Samoyad')
plt.scatter(xData_3, yData_3, color="green", marker="s", label='Maltese')
plt.xlabel('Length')
plt.ylabel('Height')

plt.legend()
plt.show()

plt.show()
```





문제 4

- 학생 관리 시스템을 클래스로 정의하고 아래 조건을 만족하는 프로그램을 작성하시오.<조건>

- 클래스에 필요한 속성과 메소드는 다음과 같다.

```
In [11]: class Person:
    def __init__(self, name, age):
        self.name = name
        self.age = age

    # Person 클래스를 상속받음
    class Student(Person):
        def __init__(self, name, age, major):
            super().__init__(name, age)
            self.major = major

    class StudentManageSystem(Student):
        def __init__(self):
            # studentList에 학생들의 정보 저장
            self.studentList = []

        # 학생 등록
        def addStudent(self, name, age, major):
            self.studentList.append([name, age, major])

        # 학생 삭제
        def removeStudent(self, name):
            if name in self.studentList[:,0]:
                removeList = []
                for list in self.studentList:
                    if name not in list:
                        removeList.append(list)

                self.studentList = removeList

            else:
                print("그런 이름의 학생은 없습니다.")

        # 전체 학생 정보 출력
        def printAllStuInfo(self):
            print(self.studentList)
```

```
In [12]: func = StudentManageSystem()
```

```
In [13]: func.addStudent("Jack", 23, "business")
func.addStudent("Bumjo", 25, "e-business")
func.addStudent("Minsu", 22, "AI")
```

```
In [14]: func.printAllStuInfo()

[['Jack', 23, 'business'], ['Bumjo', 25, 'e-business'], ['Minsu', 22, 'AI']]
```

```
In [15]: func.removeStudent("Jack")
```

```
In [16]: func.printAllStuInfo()

[['Bumjo', 25, 'e-business'], ['Minsu', 22, 'AI']]
```

```
In [17]: func.removeStudent("Jack")
```

그런 이름의 학생은 없습니다.