



<今までの問題>

[문제]

https://s3-us-west-2.amazonaws.com/secure.notion-static.com/ea9dbd37-85c8-4e7e-82d9-c990e56664c8/1._Matrix.pdf

https://s3-us-west-2.amazonaws.com/secure.notion-static.com/fba5161b-422a-499c-85f2-080bb1b4385e/2._선택문.pdf

https://s3-us-west-2.amazonaws.com/secure.notion-static.com/2cee1f67-8ff7-45de-98b5-2e3dd28e30ee/3._랜덤함수.pdf

https://s3-us-west-2.amazonaws.com/secure.notion-static.com/9a154cec-333f-4e80-88f4-0d376c3dec27/4._야구게임.pdf

[https://s3-us-west-2.amazonaws.com/secure.notion-static.com/f266e4c8-dc2f-429e-8fc1-d21829f81aec/5._문제-유효주
민등록번호_검사_알고리즘.pdf](https://s3-us-west-2.amazonaws.com/secure.notion-static.com/f266e4c8-dc2f-429e-8fc1-d21829f81aec/5._문제-유효주
민등록번호_검사_알고리즘.pdf)

[https://s3-us-west-2.amazonaws.com/secure.notion-static.com/8d784fc6-0926-4e3b-bef5-05e75fa6a5ed/6._문자열_검색
프로그램.pdf](https://s3-us-west-2.amazonaws.com/secure.notion-static.com/8d784fc6-0926-4e3b-bef5-05e75fa6a5ed/6._문자열_검색
프로그램.pdf)

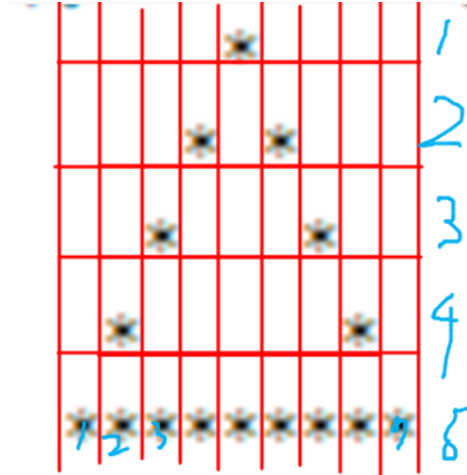
[https://s3-us-west-2.amazonaws.com/secure.notion-static.com/5c1faf9e-509f-4ea3-9010-e848206030e2/7._난수_매트릭
스.pdf](https://s3-us-west-2.amazonaws.com/secure.notion-static.com/5c1faf9e-509f-4ea3-9010-e848206030e2/7._난수_매트릭
스.pdf)

[https://s3-us-west-2.amazonaws.com/secure.notion-static.com/b9bc7f28-c132-4673-9a5a-b795c261e164/8._%ED%9
5%99%EC%83%9D_%EC%84%B1%EC%A0%81_%EC%B2%98%EB%A6%AC_%ED%94%84%EB%A1%9C%EA%B
7%B8%EB%9E%A8.pdf](https://s3-us-west-2.amazonaws.com/secure.notion-static.com/b9bc7f28-c132-4673-9a5a-b795c261e164/8._%ED%9
5%99%EC%83%9D_%EC%84%B1%EC%A0%81_%EC%B2%98%EB%A6%AC_%ED%94%84%EB%A1%9C%EA%B
7%B8%EB%9E%A8.pdf)

[https://s3-us-west-2.amazonaws.com/secure.notion-static.com/52c3a19f-dd9e-4361-aab1-18b78f5ba40b/9._%EB%8
B%A8%EC%96%B4_%EB%A7%9E%EC%B6%94%EA%B8%B0_%EA%B2%8C%EC%9E%84.pdf](https://s3-us-west-2.amazonaws.com/secure.notion-static.com/52c3a19f-dd9e-4361-aab1-18b78f5ba40b/9._%EB%8
B%A8%EC%96%B4_%EB%A7%9E%EC%B6%94%EA%B8%B0_%EA%B2%8C%EC%9E%84.pdf)

1.

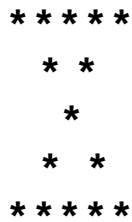
2.



9층

```
* * * * *
* * * *
* * *
* *
*
*
* *
* * *
* * * *
* * * * *
```

3.



4.

✓ Q1 반복문을 활용하여 아래와 같이 출력되는 프로그램을 작성하라

```

알의 정수를 입력 하세요8
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
1 2 3 4 5 6
1 2 3 4 5 6 7
1 2 3 4 5 6 7 8 _

```



✓ Q2 반복문을 활용하여 5 X 5 "a" Matrix 3개를 아래와 같이 출력하는 프로그램을 작성하라

```

*****
*****
*****
*****
*****

*****
*****
*****
*****
*****

*****
*****
*****
*****
*****

```



✓ Q3 반복문을 활용하여 아래와 같이 출력되는 프로그램을 작성하라

```

*****
* * *
*****
* * *
*****

****
* ***
** **
*** *
****

```



✓ Q4 반복문을 활용하여 아래와 같이 출력되는 프로그램을 작성하라

```

라인 넘버를 입력 하세요 : 5
*
**
***
**
*

```



5.

▪ 아래 프로그램을 Python 프로그래밍 언어를 사용해 작성하시오

- 1) 키보드로부터 정수 값 입력
- 2) "1" 이상의 값만 입력, "0" 이하의 값 입력 시 아래 Msg 출력 후 재입력
- 3) 현재 입력 횟수 출력 후 키보드 입력 값 화면에 출력
- 4) "짝수" or "양수" 출력
- 5) 3의 배수 또는 7의 배수이면 아래 Msg 출력
- 6) '20,000' 입력 시 아래 Msg 출력 후 프로그램 종료
- 7) 출력 값은 반드시 아래 형식 준수

▪ 출력 결과 : Happy scenario

```

1
1번째 입력 값은 = 1
      홀수 입니다.
2
2번째 입력 값은 = 2
      짝수 입니다.
3
3번째 입력 값은 = 3
      홀수 입니다.
      3의 배수 입니다.
4
4번째 입력 값은 = 4
      짝수 입니다.
5
5번째 입력 값은 = 5
      홀수 입니다.
6
6번째 입력 값은 = 6
      짝수 입니다.
      3의 배수 입니다.
7
7번째 입력 값은 = 7
      홀수 입니다.
      7의 배수 입니다.
20000
이용해주셔서 감사합니다

```

▪ 출력 결과 : Exception scenario

1) 음수 입력 시

```

5
1번째 입력 값은 = 5
                홀수 입니다.

14
2번째 입력 값은 = 14
                짝수 입니다.
                7의 배수 입니다.

-2
1이상 양수를 입력해주세요

-5
1이상 양수를 입력해주세요

19
3번째 입력 값은 = 19
                홀수 입니다.

20000
이용해주셔서 감사합니다
  
```

6.

▪ 출력 결과 : Exception scenario

1) 메뉴에서 "1" 또는 "2" 이외 값 입력 시

```

-----
1. 구구단 출력
2. 프로그램 종료
-----
4
잘못 입력하셨습니다. 다시 입력하세요.
-----
1. 구구단 출력
2. 프로그램 종료
-----
  
```

2) 구구단 출력" 메뉴 선택 후 출력 구구단의 값이 2~9 사이 이외 값 입력 시

```

-----
1. 구구단 출력
2. 프로그램 종료
-----
1
출력할 구구단의 단을 입력하세요. 구구단의 단은 2~9 사이 입력
1
2~9사이 정수를 입력해주세요
  
```

- 아래 프로그램을 Python 프로그램 언어를 사용해 작성하시오
 - 1) 메뉴 우선 출력 후 키보드로부터 정수 값 입력
 - 2) 메뉴에서 "1" 선택 시 구구단 출력, "2" 인 경우 Msg. 출력 후 프로그램 종료
 - 3) 메뉴에서 "1" 또는 "2" 이외의 값이 입력될 경우, Error Msg. 출력 후 재입력
 - 4) "구구단 출력" 메뉴 선택 시 출력 할 단을 키보드로부터 입력
 - 4-1 출력 유효 단은 2 ~ 9
 - 4-2 2 ~ 9단 이외의 값이 들어올 경우 Error Msg. 출력 후 재입력
 - 5) 출력 값은 반드시 아래 형식 준수

▪ 출력 결과 : Happy scenario

```

-----
1. 구구단 출력
2. 프로그램 종료
-----
1
출력할 구구단의 단을 입력하세요. 구구단의 단은 2~9 사이 입력
3
3 X 1 = 3
3 X 2 = 6
3 X 3 = 9
3 X 4 = 12
3 X 5 = 15
3 X 6 = 18
3 X 7 = 21
3 X 8 = 24
3 X 9 = 27
-----
1. 구구단 출력
2. 프로그램 종료
-----
2
이용해주셔서 감사합니다.

```

7.

랜덤 함수

- 랜덤함수를 이용하여 아래 프로그램을 작성하라

1. 1 ~ 20 사이 양의 정수 중 난수 값 20개 생성 후 List에 저장
2. List 내 원소 값에 대한 합계, 평균, 최대 값, 최소 값 출력
3. List 내 중복 값과 중복 횟수 정보 출력 [아래 출력 결과 참조]
4. 구간 별 히스토그램 정보 출력 [아래 출력 결과 참조]

※ 출력 결과

랜덤 값 :

19 12 19 17 3 18 11 8 9 5
20 8 4 6 12 17 11 4 3 11

최소 값 : 3

최대 값 : 20

합계 : 217

평균 : 10.85

중복 값 중복 회수

3 2
4 2
8 2
11 3
12 2
17 2
19 2

구간별 히스토그램

1 ~ 5 : *****

6 ~ 10 : *****

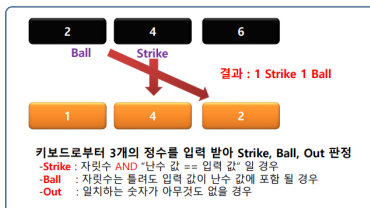
11 ~ 15 : *****

16 ~ 20 : *****

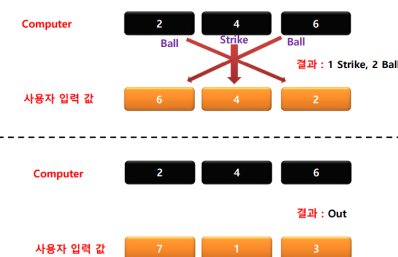
8. 야구게임

프로그래밍: 야구 게임 만들기

0~9사이의 3개 정수를 random.randint() API를 이용하여 난수로 생성.
-정수의 범위는 0~9사이
-중복 값 없이 생성, 예) 2, 2, 6 X → 2중복



판별 결과 예)



- 게임 시작 시 0~9사이 정수 중 중복 없이 있는 난수 3개 생성
- 키보드로부터 0~9사이 정수 3개를 입력 받고 결과 값을 출력 (예외처리 X)
- 아래 경우 게임 Lose
 - 시도 횟수 == 5
 - Strike Out == 3
- 아래 경우 게임 Win
 - 컴퓨터에서 생성 한 난수 값을 자리 순서대로 맞출 경우
- 출력 포맷은 아래와 동일 할 것!!
 - 출력 내용은 시도 횟수 카운트, 아웃 횟수 카운트, Strike Ball 판별 결과, 게임 결과, 횡단 포함
 - 아래 실행 결과 참조

```
실행결과
시도횟수: 1
난수 3개를 입력하세요---
2 4 6
Out: 아웃 1번

시도횟수: 2
난수 3개를 입력하세요---
2 8 9
1 Strike

시도횟수: 3
난수 3개를 입력하세요---
2 4 4
2 Strike 1 Ball

시도횟수: 4
난수 3개를 입력하세요---
2 7 6
3 Strike

게임횟수 종료
종단점: 0 0 4 1입니다.
올바라요! 성공! 축하! 누르십시오 . . .
```

9. 유효 주민등록 번호 검사

유효 주민등록번호 검사 알고리즘

주민등록번호	7	9	0	6	0	8	-	2	5	5	2	4	1	출
체크수	7	9	4	5	6	7	8	9	2	3	4	5	6	
유효 계산공식(1)	14	27	0	30	0	55	15	45	10	0	15	5		
유효 계산(2)	$14 \times 27 \times 0 \times 30 \times 0 \times 55 \times 15 \times 45 \times 10 \times 0 \times 15 \times 5 = 202500$													
유효 계산	11 - 202500 = 1로 작은 숫자이기 때문에 0이 된다													

주민번호 13자리를 입력하세요 801545-1798757
유효하지 않은 주민번호입니다.

주민번호 13자리를 입력하세요 2-17
유효한 주민번호입니다.

개인정보 가릴 처리

숫자 검사 함수

```

1 num1 = "3"
2 num2 = "-"
3
4 if num1.isdigit():
5     print("True 1")
6 else:
7     print("False 1")
8
9 if num2.isdigit():
10    print("True 1")
11 else:
12    print("False 1")

```

True 1
False 1

10.

문자열 검색 프로그램

1. 영문 문자열을 키보드로부터 입력 받아 List에 저장 후, 검색 단어를 키보드로부터 다시 입력 받아 해당 단어가 있을 경우 결과 값 출력

1.1 사용자로부터(키보드) 입력 받을 문자열의 라인 수 입력 후 리스트에 저장

1.2 검색 할 단어를 키보드로부터 입력 받고, 찾는 문자열이 있을 경우 아래와 같이 출력

1.2.1 검색 된 단어 유/무 출력

찾는 문자열이 없을 경우 계속해서 검색 문자열 입력

1.2.2 검색 된 단어의 줄, 총 검색 횟수 출력

1.3 입력받은 문자열의 단어 개수 카운트 후 출력

※ 출력 결과 :

```

입력 문자열의 줄(Line) 수를 입력하세요!
3
1 번째 라인의 문자열을 입력하세요.
hello world
2 번째 라인의 문자열을 입력하세요.
hello worldddd hello
3 번째 라인의 문자열을 입력하세요.
kkk
검색 할 문자열을 입력하세요.
ppp
찾을 수가 없습니다. 검색 할 문자열을 입력하세요.
hello
hello를 찾았습니다.
검색된 라인 수 : 1, 2
검색된 횟수 : 3
총 단어 수 : 6

```

11.

난수 발생 후 MATRIX, 최대, 최소, 중간 값 출력

랜덤 함수를 이용하여 1 ~ 50 사이 정수 중 중복되지 않은 값 25개를 선택하여 LIST에 저장하고 아래와 같이 최대, 최소, 중간 값을 찾아 출력하는 프로그램을 작성하라.

▣ 프로그램 동작 절차 설명

1. 1~50 사이의 중복되지 않은 정수형 난수 25개를 선택하여 LIST에 저장
2. 각 열, 행 별 최대, 최소, 중간 값을 찾아 출력하라.
 - 중간 값: 데이터들을 크기순으로 배열 했을 때 전체의 중앙에 위치하는 수
예) 1, 3, 5, 10, 4 -> 중간 값: 4
4. LIST 전체를 기준으로 최대, 최소, 중간 값을 출력하라.

▣ 요구조건

- ✓ 최대, 최소, 중간값 함수 사용 금지
- ✓ 중간값을 찾기 위한 sort 함수는 사용 가능

프로그램 결과 값 예시

	1	2	3	4	5
	6	7	8	9	10
	11	12	13	14	15
	16	17	18	19	20
	21	22	23	24	25
열					
최소값	1	2	3	4	5
최대값	21	22	23	24	25
중간값	11	12	13	14	15

행

최소값	최대값	중간값
1	5	3
6	10	8
11	15	13
16	20	18
21	25	23

전체

최소값	1
최대값	25
중간값	13

12.

컴퓨터사고와 SW 코딩 통합 시험

2022년 6월 22일 오전 10시

1. For 문을 사용하여 다섯 개의 정수를 키보드로부터 입력 받아, 합계와 평균을 출력하는 프로그램 작성

실행결과
1번째 값 입력1 2번째 값 입력2 3번째 값 입력3 4번째 값 입력4 5번째 값 입력5 합계 : 15 평균 : 3.0

2. 키보드로부터 정수를 입력 받고, 양수이면 "양수" 문자열을 음수이면 "음수" 문자열을 출력하는 프로그램 작성.
단 0을 입력 받을 경우 프로그램 종료
※ 요구사항 : while문 사용

실행결과
정수를 입력 하세요3 양수 입니다. 정수를 입력 하세요-2 음수 입니다. 정수를 입력 하세요0 PS D:\python_prj>

3. 다음과 같이 출력되는 프로그램을 작성하라
※ 요구사항 : For 또는 while문 사용

실행결과
***** **** *** ** *

4. 1 ~ 100까지 양의 정수 중 "3" 이 포함된 정수만 출력하라

실행결과
3 13 23 30 31 32 33 34 35 36 37 38 39 PS D:\python_prj>

5. 다음 문자열에서 특수문자 개수, 글자수, 단어 수 카운트 프로그램을 작성하라

실행결과
1 # 특수 문자는 3개 사용 : ! . , 2 myList = "!! hello world, it is awesome day." 특수문자 수 : 4 단어 수 : 6 특수문자 제외 글자수 : 24 PS D:\python_prj>

https://s3-us-west-2.amazonaws.com/security.notion-static.com/bf63ec5c-94b4-42c0-be1c-3b0e20e33767/%E4%B8%A9%EC%99%80_SW_%EC%B9%A9%ED%86%B5%E4%95%A9%EC%8B%9C%E4%97%98.pdf

<https://s3-us-west-2.amazonaws.com/security.notion-static.com/e6f45865-86fa-4e44-a284-985aa51b83dd/test1.py>

<https://s3-us-west-2.amazonaws.com/security.notion-static.com/24b1140e-f96d-43f4-8538-0e6b625e8d67/test2.py>

<https://s3-us-west-2.amazonaws.com/security.notion-static.com/710a8cfd-78b7-4564-a400-ab69ed27f8ac/test3.py>

<https://s3-us-west-2.amazonaws.com/security.notion-static.com/4656efc9-9561-49c6-8b98-54000c3884e5/test4.py>

<https://s3-us-west-2.amazonaws.com/security.notion-static.com/d6502dda-4d1c-40ab-b2fd-29744320380c/test5.py>

13. 학생 성적 처리 프로그램

재일

```
# 소수점 조정
import math

# 정보담을 리스트 (학생 정보 확인용)
Info_List = [] # 개인 정보를 2차원 리스트에 저장

# 각 출력 내용 리스트 (각 성적값 확인용)
IdOfStudent_List = []
Name_List = []
Kor_List = []
Eng_List = []
Math_List = []
```

```
# 변수
Score_Count = 1 # 성적개수
Sum = 0 # 총합
```

```
# -----
```

```
# 출력문
def Guidance () :
    Avg = Sum / Score_Count # 평균
    print("=====")
    print(" 1. 학생 성적 입력")
    print(" 2. 학생 목록 출력 (입력순)")
    print(" 3. 프로그램 종료")
    print()
    print("현 입력데이터 갯수 : ", len(Info_List))
    print("전체 학생 평균 값 : ", round(Avg,2))
    print("=====")
```

```
#
def OutputWord (Word) :
    input(f'{Word}을 입력하세요 : ')
```

```
# 전체 반복문
while True :
    # 안내문 호출
    Guidance ()
```

```
if len(Info_List) <= 0:
    Score_Count = 0
```

```
# 선택문 출력
Input_Num = int(input("1 ~ 3 번중 하나를 선택하시오 : "))
```

```
#### 1. 인 경우 ####
if Input_Num == 1 :
```

```
    Info_List.append([])
```

```
for Index in range (Input_Num) :
    # 입력 받을 내용 1)학번 2)이름 3)국어 성적 4)영어 성적 5)수학 성적
    # 인풋 값 각각의 리스트에 저장
    IdOfStudent = OutputWord ("학번")
    Info_List[len(Info_List)-1].append(IdOfStudent)
    IdOfStudent_List.append(IdOfStudent)
```

```
    Name = OutputWord ("이름 성적")
    Info_List[len(Info_List)-1].append(Name)
    Name_List.append(Name)
```

```
    Kor = int(OutputWord ("국어 성적"))
    Info_List[len(Info_List)-1].append(Kor)
    Kor_List.append(Kor)
```

```
    Eng = int(OutputWord ("영어 성적"))
    Info_List[len(Info_List)-1].append(Eng)
    Eng_List.append(Eng)
```

```
    Math = int(OutputWord ("수학 성적"))
```

주원

```
# 학생 정보를 반환하는 함수 정의
def getInputValue( informationOfStudent):
    inputValue = '' # 학
    sum = 0 # 학

    for key,value in informationOfStudent.items(): # 학
        inputValue += ' %s : '%key + str(value)

        if key == 'kor' or key == 'eng' or key == 'math' : # 점
            sum += value
    avg = sum / 3

    return avg, [inputValue + " sum : " + str(sum) + " avg : " + str(avg)]
```

```
# 값을 입력받는 함수 정의
def getValue(value):
    return input(value + "을 입력하세요\n")
```

```
inputValueList = [] # 입력데이터 담아줄 리스트
sumOfAves = 0 # 평균값들의 합, 평균값들의 평균
```

```
while True:
    # 전체 평균 점수
    avgOfAves = \
        sumOfAves/len(inputValueList) if len(inputValueList) != 0 else 0

    # 기본 출력 메시지
    print("=====")
    print(" 1. 학생 성적 입력")
    print(" 2. 학생 목록 출력(입력 순)")
    print(" 3. 프로그램 종료")
    print("현 입력데이터 갯수 :", len(inputValueList))
    print("전체 학생 평균 값 :", avgOfAves)
    print("=====")
```

```
# 선택
selectValue = input("")
```

```
# 1 학생 성적 입력
if selectValue == '1':
    avg, inputValue = \
        getInputValue(id = getValue("학번"), name = getValue("이름"))
```

```
    inputValueList.append(inputValue)
    sumOfAves += avg
```

```
# 2 학생 목록 출력
elif selectValue == '2':
    for studentList in inputValueList:
        print(studentList)
```

```
# 3 프로그램 종료
elif selectValue == '3':
    break
```

```
# 4 예외처리
else:
    selectValue = input("선택할 수 있는 값을 입력하세요")
```

용현

```
# 학생 성적 처리 프로그램 : 학생들의 성적을 키보드로부터 입력 받아 리스트에 저장
student_num_list = [] # 학번 리스트 and 반복 횟수 리스트(len)
name_list = [] # 이름 리스트
kor_list = [] # 국어 리스트
eng_list = [] # 영어 리스트
math_list = [] # 수학 리스트
sum_list = [] # 합계 리스트
avg_list = [] # 평균 리스트
date_count = 0 # 입력데이터 갯수를 나타내는 변수
all_sum = 0 # 전체 합계
```

```

        Info_List[len(Info_List)-1].append(Math)
        Math_List.append(Math)

# 입력 받은 성적값 변수에 저장
for Index_Score in range( 2 , 4+1 ) :
    # 국어
    if Index_Score == 2 :
        Kor = Info_List[len(Info_List)-1][Index_Score]
        Score_Count += 1

    # 영어
    elif Index_Score == 3 :
        Eng = Info_List[len(Info_List)-1][Index_Score]
        Score_Count += 1

    # 수학
    else :
        Math = Info_List[len(Info_List)-1][Index_Score]
        Score_Count += 1

# 합계에 넣기
Score_Subject = Kor + Eng + Math
Sum += Score_Subject
continue

##### 2. 인 경우 #####
elif Input_Num == 2 :

    for Index in range( len(Info_List) ) :
        print(["id : " , IdOfStudent_List[Index] , "name : " , Name_List[Index] , "Kor : " , Kor_List[Index] , "Eng : " , Eng_List[Index] , "Math : " , Math_List[Index] , "합계 : " , round(((Kor_List[Index] + Eng_List[Index] + Math_List[Index]) / 3) , 1) , "평균 : " , round(((Kor_List[Index] + Eng_List[Index] + Math_List[Index]) / 3) , 1) , "전체 평균 : " , all_avg] , " ")

##### 3. 인 경우 #####
else :
    print('시스템 종료')
    break

```

```

all_avg = 0 # 전체 반복
# (1) 각 입력 값 생성 -> 입력 값 각 리스트에 저장
def input_value():
    student_num = input("학번을 입력하세요")
    student_num_list.append(int(student_num))
    name = input("이름을 입력하세요")
    name_list.append(name)
    kor = input("국어 성적을 입력하세요")
    kor_list.append(int(kor))
    eng = input("영어 성적을 입력하세요")
    eng_list.append(int(eng))
    math = input("수학 성적을 입력하세요")
    math_list.append(int(math))

# (2) 메뉴 선택한 생성 및 현 입력 데이터 갯수 및 전체 학생 평균 값 생성
def menu():
    print("=====")
    print("1. 학생 성적 입력")
    print("2. 학생 목록 출력(입력 순)")
    print("3. 프로그램 종료")
    print()
    print("현 입력데이터 갯수: " , date_count)
    print("전체 학생 평균 값 : " , all_avg)
    print("=====")

# (3) 메뉴 선택
while True:
    menu()
    num = int(input())
    date_count += 1
    # 1번 메뉴 선택 -> 학생 정보 입력
    if num == 1:
        for sum_index in range(len(student_num_list)):
            sum = kor_list[sum_index] + eng_list[sum_index] + math_list[sum_index]
            if sum not in sum_list:
                sum_list.append(sum)
        # 평균 생성
        for avg_index in range(len(student_num_list)):
            avg = sum / 3
            if avg not in avg_list:
                avg_list.append(avg)
        # 전체 학생 평균
        for all_index in range(1):
            all_sum += sum
            all_avg = all_sum / 3 / len(student_num_list)
            print(all_avg)
    # 2번 메뉴 선택 -> 학생 목록 출력
    elif num == 2:
        for index in range(len(student_num_list)):
            print("[id:" , student_num_list[index] , "name:" , name_list[index] , "Kor:" , kor_list[index] , "Eng:" , eng_list[index] , "Math:" , math_list[index] , "합계:" , round(((kor_list[index] + eng_list[index] + math_list[index]) / 3) , 1) , "평균:" , round(((kor_list[index] + eng_list[index] + math_list[index]) / 3) , 1) , "전체 평균:" , all_avg] , " ")
    # 3번 메뉴 선택 -> 프로그램 종료
    elif num == 3:
        print("프로그램 종료")
        break

```

14. 단어 맞추기 게임

```

import random # 랜덤모듈 가져오기

#####

# F1. 함수 구현 리스트
def Output_Unit (Number) :
    Suffix = ["첫", "두", "세", "네", "다섯", "여섯", "일곱", "여덟", "아홉", "열"]
    Count = 0

    while len(Input_Words_List) < 3 :
        Word = input(str(Suffix[Count])+" 번째 단어를 입력 하세요\n")
        print()
        if 5 <= len(Word) <= 20 :
            Input_Words_List.append(Word)
            Count += 1
        else :
            print("5이상 20이하 글자로 구성된 단어를 입력 하세요.")
            print()

# F2. 올림 함수
def Raising (Word_Selected) :

    Blind_Count = 0 # 블라인드 처리 수

```

```

Half_Nun= len(Word_Selected)/2

# Half_Nun 짝수 일 때 반올림
if type(Half_Nun) == float and len(Word_Selected)%2 == 0:
    Blind_Count = int(Half_Nun)

# Half_Nun 홀수 일 때 반올림
elif type(Half_Nun) == float and len(Word_Selected)%2 != 0:
    Blind_Count = int(Half_Nun)+1

return Blind_Count

# F3. 출력문 반복 함수
def Trial_Count (Counting,BlankWord_List) :
    print(f'{Counting} 번째 시도, 아래 단어를 구성하는 알파벳 한 개를 입력하세요')
    # 현재 알고 있는 알파벳 출력
    for Element_BlankWord_List in BlankWord_List :
        print(Element_BlankWord_List,end=" ")
    print()
    print()

#####

Input_Words_List = [ ] # 키보드로 부터 받은 영단어 3 개 저장

# 1. 키보드로부터 영어 단어 3개를 입력 받아 리스트에 저장
Output_Unit (3)

# 2. 입력된 3개의 단어 중 한 개 단어를 임의 선택
Random_Index= random.randint(0,2) # 랜덤 인덱스

Word_Selected = Input_Words_List[Random_Index] # 랜덤 단어

# 3. 게임 시작을 알리는 문장 출력 => 단어 선택 완료 게임을 시작 합니다. 선택된 단어 :
if len(Input_Words_List) == 3 :
    print(f"단어 선택 완료 게임을 시작 합니다. 선택된 단어 : {Word_Selected}")

# 4. 선택된 단어의 글자 중 50%를 Blind 처리, Blind 처리 알파벳은 랜덤하게 선택

Blind_Num_Word = Raising(Word_Selected) # 올림처리 한 블라인드 갯수
Random_List = [] # 랜덤으로 얻은 리스트
BlankWord_List = [] # 블랭크 처리된 단어 리스트
BlankWord_Str = "" # 블랭크 처리된 단어 변수

# 4.1 반복문을 사용한 랜덤으로 단어 가리기
for Index in range (len(Word_Selected)) : # 반복 횟수는 단어 길이 만큼

    # 엘리먼트 추출
    Element = Word_Selected[Index]

    # 블라인드 처리
    while len(Random_List) < Blind_Num_Word :
        RANDOM_Num = random.randint(0,len(Word_Selected))

        # Random_List 요소 생성
        if not RANDOM_Num in Random_List :
            Random_List.append(RANDOM_Num)

    # Random_List 안에 있으면 "_" 처리
    if Index in Random_List :
        Element = "_"
        BlankWord_List.append(Element)
        BlankWord_Str += Element
    else :
        BlankWord_List.append(Element)
        BlankWord_Str += Element

print()
#####

# 5. 문제 맞추기

Duplicated_Num = 0 # 포함된 알파벳 알림 수
Counting = 1 # 시도 횟수 측정

# 5.1 문제 맞출시 까지 반복
while True :

    #출력문 (함수사용 F3 )
    Trial_Count (Counting,BlankWord_List)
    # 입력값 받기
    InputNum_Trial = input()
    print()

```

```

# 5.1.1 Blank 처리된 리스트 요소들과 "단어리스트" 비교
for Index in range (len(BlankWord_List)):
    # 블랭크 리스트에 있는 요소들 하나씩 뽑기
    Element = BlankWord_List[Index]
    # 만약 "입력값"과 "단어리스트"의 값이 같다면
    if Word_Selected[Index] == InputNum_Trial :
        # 공백 부분을 " 입력값 "으로 교환
        BlankWord_List[Index] = InputNum_Trial
        Duplicated_Num+=1
    # 입력 값이 해당 단어에 없을 경우
    elif not InputNum_Trial in Word_Selected :
        print("단어 내 포함되지 않은 알파벳 입니다.")
        print()
        Counting += 1

    #출력문 ( 함수사용 F3 )
    Trial_Count (Counting,BlankWord_List)
    # 입력값 받기
    InputNum_Trial = input()
    print()
    continue

# 5.1.2 알파벳을 하나씩 입력받아 단어안에 들어 있으면
if not " " in BlankWord_List :
    # 프로그램 종료전 출력문
    print("Clear - 선택된 단어 : ",Word_Selected," , 총 시도 횟수 : ",Counting,)
    break

# 출력문
print("입력한 알파벳 단어 내 포함 : ", Duplicated_Num,"글자")

Counting += 1      # 반복문 횟수 세기
Duplicated_Num = 0 # 포함된 단어 초기화

```