HomeWork1

- 기간 : 11월 8일부터 14일까지
- 파이썬 파일(.py)파일을 자신의 학번_hw1_1.py형식으로
  저장하여 담당TA메일로 제출하기
- 각 프로그램 문장에 comment를 넣기 (프로그램 내에도 자신의 이름_학번을 기술함)

과제 1

- 3,6,9 게임을 구현합니다
- 1~100사이의 숫자를 이용하며 3,6,9 게임과 같은 형식으로 포함된 개수만큼 박수(clap)을 나타내도록 한다.
- 즉, 박수의 횟수를 나타내도록 하며 리스트를 이용하기.
- 함수를 사용하여 나타냄 예) def game_369(n)

```
if __name__ == '__main__':
    game_369(8)
    game_369(2)
    game_369(50)
    game_369(100)
```

```
2 Claps!
0 Clap!
25 Claps!
out or range!
```

과제 2

- N*N 맵을 입력받아 시작점 0,0에서 종점 N-1,N-1까지 도달할 수 있는 경로의 개수를 반환하는 함수를 작성하기
- 이동은 오른쪽과 아래쪽으로만 하도록 제한하며, 해당 x,y좌표에 나와있는 숫자만큼 한방향으로 이동한다
- 이동후 맵을 벗어난 경우에는 더 이상 경로가 존재하지 않는다고 설정한다.
- 재귀함수를 사용하여 이용하기 – def calcPath(trace_map,x,y)
- 매개변수 : trace_map : 2중 리스트로 입력된 N*N맵으로 제시된 숫자가 오른쪽이나 아래쪽으로 이동하는 칸 수
- x: 정수형으로 x좌표, y: 정수형으로 y좌표
- 반환값 : 종점까지 도달할 수 있는 총 path의 count(종점까지 도착하는 path가 없으면 count는 0, 맵을 벗어나는 경우도 0으로 간주함)

과제 2

◦ 호출

```
if __name__=='__main__':
    trace_map=[[1,2,9,4,9],
               [9,9,9,9,9],
               [9,3,9,9,2],
               [9,9,9,9,9],
               [9,9,9,1,0]]
    print("총 횟수는 %d회" %(calcPath(trace_map,0,0)))

    trace_map=[[1,1,1],
               [1,1,2],
               [1,2,0]]
    print("총 횟수는 %d회" %(calcPath(trace_map,0,0)))
```

총 횟수는 2회
총 횟수는 0회

과제 3: 중간고사 성적정리 프로그램

- 1학년 학생들의 공통필수 교과들의 중간고사 점수를 정리하고자 하며, 공통 필수 교과목의 이름은 course_titles 리스트에, 중간고사 성적은 midterm_scores 리스트에 입력하였다
- 학생이름들과 중간고사 성적이 저장된 리스트의 값부분은 교과목의 성적들로 대응한다
- 즉 Alice의 선형대수, 세포와 생명현상, 프로그래밍 중간고사 점수는 각각 62, 60, 77이다

```
course_titles=['Linear Algebra','Cell Biology','Programming']
midterm_scores=[
    ['Alice',[62,60,77]],
    ['Bob',[84,88,87]],
    ['Carol',[88,88,97]],
    ['Chuck',[92,71,93]],
    ['Craig',[81,89,72]],
    ['Dan',[79,100,97]],
    ['Erin',[76,72,65]],
    ['Eve',[69,84,67]],
    ['Faythe',[66,60,70]],
    ['Franke',[88,76,64]],
    ['Grace',[66,72,92]],
    ['Heidi',[93,93,82]],
    ['Mallory',[77,89,82]],
    ['Oscar',[67,63,67]],
    ['Peggy',[70,81,86]],
    ['Sybil',[96,94,62]],
    ['Trent',[67,66,61]],
    ['Trudy',[85,74,90]],
    ['Victor',[82,67,94]],
    ['Walter',[67,84,86]],
    ['Wendy',[97,92,66]]]
```

함수호출 부분

```
36 ▶ if __name__ =='__main__':
37       print_scores(midterm_scores,course_titles,'Alice')
38       print_scores(midterm_scores, course_titles, 'Victor')
39       print_scores(midterm_scores, course_titles, 'Elice')
```

결과

```
Alice:
    Linear Algebra: 62
    Cell Biology: 60
    Programming: 77
    Average: 66.33333333333333
Victor:
    Linear Algebra: 82
    Cell Biology: 67
    Programming: 94
    Average: 81.0
There is no student named Elice.
```

- 함수 print_scores(test_scores, course_titles, student_name)을 작성
  - 인자
    - test_scores: midterm_scores의 값을 받는 딕셔너리
    - Corese_titles:과목명(문자열)이 저장된 리스트
    - student_name:호출한 학생의 이름
  - 반환값 : 없음
  - 함수의 동작
    - 학생이 이름이 test_scores에 없으면 학생이름이 없음을 나타내는 메시지 출력. 'Elice'는 없으므로 Elice가 없다고 나타냄
    - 학생이 이름이 test_scores에 있으면 학생의 이름, 각 과목별 점수, 평균을 출력한다. 옆의 실행결과에 맞게 출력하기

함수 추가하여 호출한 부분

```
50  if __name__ =='__main__':
51      print_scores(midterm_scores,course_titles,'Alice')
52      print_scores(midterm_scores, course_titles, 'Victor')
53      print_scores(midterm_scores, course_titles, 'Elice')
54
55      print("과목별 평균, 최대, 최소값을 출력")
56      for course_name in course_titles:
57          subject_scores = get_scores(midterm_scores,course_titles,course_name)
58          print_midterm_stat_by_course(course_name,subject_scores)
```

결과

```
Alice:
    Linear Algebra: 62
    Cell Biology: 60
    Programming: 77
    Average: 66.33333333333333
Victor:
    Linear Algebra: 82
    Cell Biology: 67
    Programming: 94
    Average: 81.0
There is no student named Elice.
과목별 평균, 최대, 최소값을 출력
Linear Algebra: 88.0 97 66
Cell Biology: 88.8 97 66
Programming: 85.0 97 66
```

- 함수 2가지 추가
- get_scores(test_scores,course_title,course_name)
  - 인자
    - test_scores: 위의 midterm_score의 형태로 저장된 리스트
    - course_titles:과목명에 대한 리스트
    - course_name:과목이름
  - 반환값
    - 과목의 이름이 없는 경우: None
    - 과목의 이름이 있는 경우: list 값(course_name으로 지정된 과목의 점수들이 저장된 리스트)
- print_midterm_stat_by_course(course_name,subject_scores)
  - 인자
    - course_name: 과목이름이 저장된 문자열
    - subject_scores:과목의 점수를 저장한 리스트
  - 반환값 : 없음