

1시간 동안 진행되는 코딩 시험입니다. 주어진 뼈대코드를 내려받아 완성한 다음, 제출 마감 시간 안에 파일을 업로드 하세요.

(6문제, 2쪽)

시험 보는 동안 타인과 소통 시도는 부정행위로 간주합니다. 컴퓨터에 소통을 위한 브라우저를 켜는 것 자체도 부정행위로 간주하고 금합니다. 부정행위로 적발되는 경우 F 처리하고, 대학 본부에 보고합니다.

**문제 1. [6점] ASCII 아트**

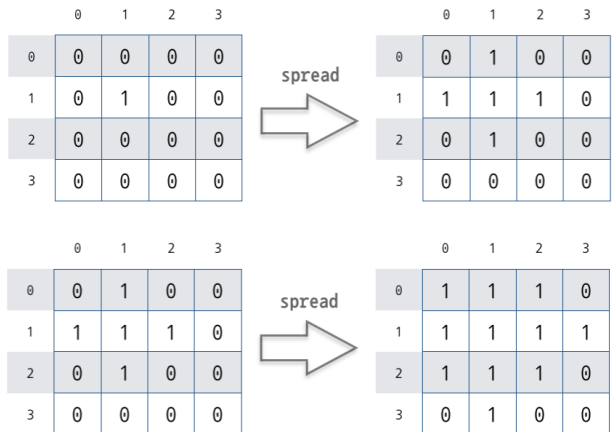
자연수를 인수로 받아서 아래의 실행 사례와 같이 실행창에 프린트하는 함수 triangle를 작성하시오.

**실행 사례**

```
>>> triangle(3)
1
222
33333
>>> triangle(6)
1
222
33333
4444444
55555555
6666666666
>>> triangle(13)
1
222
33333
4444444
55555555
6666666666
77777777777
888888888888888
999999999999999
0000000000000000000
11111111111111111111
22222222222222222222
3333333333333333333333
```

**문제 2. [6점] 퍼트리기**

0과 1로 구성된 n x n 보드를 인수로 받아서 1의 상, 하, 좌, 우 인접한 칸을 모두 1로 바꾸어 리턴하는 함수 spread를 작성하시오. 예를 들어 아래 왼쪽에 있는 보드를 인수로 받으면, 아래 오른쪽에 있는 보드를 리턴한다.



n x n 보드는 2차원 리스트로 제공되고, 위 그림의 사례는 각각 다음과 같이 실행되어야 한다.

**실행 사례**

```
>>> spread([[0,0,0,0],[0,1,0,0],[0,0,0,0],[0,0,0,0]])
[[0,1,0,0],[1,1,1,0],[0,1,0,0],[0,0,0,0]]
>>> spread([[0,1,0,0],[1,1,1,0],[0,1,0,0],[0,0,0,0]])
[[1,1,1,0],[1,1,1,1],[1,1,1,0],[0,1,0,0]]
```

**문제 3. [5점] 빈도수 기록 딕셔너리 만들기**

시퀀스 `s`를 인수로 받아 `s`에 나타나는 원소의 빈도수를 기록하는 딕셔너리를 만들어 리턴하는 함수 `create_frequency_dictionary`를 작성하시오. `s`의 원소가 키가 되고, 그 원소가 `s`에 나타나는 빈도수가 리턴하는 딕셔너리의 그 키의 값이 된다. 예를 들면 시퀀스 "hanyangansan"을 인수로 이 함수를 다음과 같이 호출하면

```
create_frequency_dictionary("hanyangansan")
```

다음과 같은 딕셔너리를 리턴해야 한다.

```
{'h': 1, 'a': 4, 'n': 4, 'y': 1, 'g': 1, 's': 1}
```

문제 4~6은 연결된 문제입니다.

- 문제 6이 맞으면 다 맞은 걸로 처리하여 18점입니다.
- 문제 6이 틀렸지만 문제 5가 맞으면, 문제 4,5 모두 맞은 걸로 처리하여 12점입니다.
- 문제 5, 6이 틀렸지만 문제 4는 맞으면 6점입니다.
- 따라서 문제를 풀면서 문제 별로 완성본을 따로 보존해두기 바랍니다.

이 함수를 실행하는데 텍스트 파일 "poem.txt"가 필요합니다. 내려받아서 사용하세요.

**문제 4. [6점] 텍스트 파일 문자 빈도수 Top 5 프린트 하기**

텍스트 파일의 이름을 인수로 받아서, 그 파일에 있는 문자의 빈도수를 세어 빈도수가 많은 순서대로 랭킹을 매겨 1위부터 5위까지 문자와 빈도수를 나열하는 프로시저 `show_character_top5`를 문제 3에서 작성한 `create_frequency_dictionary` 프로시저를 활용하여 작성하시오. 그런데 다음 특수 문자는 Top 5에서 제외한다.

```
"\n" " " ". " ", " ""
```

샘플로 주어진 파일 이름 "poem.txt"를 인수로 이 프로시저를 호출하면 다음과 같은 형식으로 실행창에 프린트 해야 한다.

**실행 사례**

```
>>> show_character_top5("poem.txt")
1. 이 = 34
2. 나 = 18
3. 에 = 14
4. 다 = 14
5. 는 = 12
```

**문제 5. [+6점] 텍스트 파일 문자 빈도수 Top 5 프린트 하기 (계속)**

문제 4의 프로시저 호출 결과를 보면 3위와 4위의 빈도수가 같다. 따라서 두 문자 모두 아래와 같이 순위를 3위로 하는 것이 타당하다. 빈도수가 동률인 경우 같은 순위를 프린트하도록 `show_character_top5` 프로시저를 수정하시오.

**실행 사례**

```
>>> show_character_top5("poem.txt")
1. 이 = 34
2. 나 = 18
3. 에 = 14
3. 다 = 14
5. 는 = 12
```

**문제 6. [+6점] 텍스트 파일 문자 빈도수 Top 5 프린트 하기 (계속)**

그런데 5개만 프린트 하도록 하면 문제가 있다. 10위까지 프린트한 다음 결과를 보면 무엇이 문제인지 알 수 있다.

```
1. 이 = 34
2. 나 = 18
3. 에 = 14
3. 다 = 14
5. 는 = 12
5. 니 = 12
5. 별 = 12
8. 하 = 11
9. 아 = 9
9. 름 = 9
```

"는" 뿐 아니라 "니"와 "별"도 공동 5위인데, Top 5에서 제외되었다. 5위까지 공동 순위도 모두 아래와 같이 프린트하도록 `show_character_top5` 프로시저를 수정하시오.

**실행 사례**

```
>>> show_character_top5("poem.txt")
1. 이 = 34
2. 나 = 18
3. 에 = 14
3. 다 = 14
5. 는 = 12
5. 니 = 12
5. 별 = 12
```