

---

# 소프트웨어 입문 설계

실습 5주차

과제 5-1

김윤호 교수님

2021 봄학기

# 목차

---

- 과제 4-2 리뷰
- 과제 5-1 안내

# 과제 4-2 리뷰

- 1번 문항
  - 1부터 n까지의 숫자를 이용해 예시와 같은 삼각형 모양을 출력
  - 반드시 아래와 같은 함수 작성 후 여러 번 호출하여 삼각형 모양을 출력
    - `printLine(n)` : 1부터 n까지의 숫자를 한 줄로 출력 후 맨 끝에서 '줄바꿈 문자' 출력  
해당 함수를 한 번 호출 시, 단 한 줄의 출력 결과만 나와야 함

```
def printLine(n):                # 1부터 최대 n까지 한 줄에 출력
    for i in range(1, n+1):
        print(i, '', end='')    # 출력할 수 뒤에 공백 삽입
    print()                    # 개행 (줄 바꿈)

maxNum = int(input())

for i in range(1, maxNum+1):    # 최대 1부터 maxNum까지
    printLine(i)

for i in range(maxNum, 0, -1):  # 최대 maxNum부터 1까지
    printLine(i)
```

(실행 예)

```
5↵
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
1 2 3 4 5
1 2 3 4
1 2 3
1 2
1
```

# 과제 4-2 리뷰

- 2번 문항
  - 하단 두 함수를 구현하고 두 개의 정수를 입력 받음
  - 두 개의 정수에 대한 약수를 구하여 출력
    - **problemDescription()**: '\* Number of divisors \*' 문구 출력
    - **getNumOfDivisors(number)**: number의 약수 개수를 반환

(실행 예)

```
* Number of divisors *
Type the first number:
12↵
Type the second number:
6↵
Number of divisors of the first number is 6
Number of divisors of the second number is 4
```

```
def problemDescription():
    print('* Number of divisors *')

def getNumOfDivisors(number):
    divisorCnt = 0

    for i in range(1, number+1):
        if number % i == 0:
            divisorCnt += 1

    return divisorCnt
```

```
num1 = int(input('Type the first number:\n'))
num2 = int(input('Type the second number:\n'))

divisor1 = getNumOfDivisors(num1)
divisor2 = getNumOfDivisors(num2)

print('Number of divisors of the first number is', divisor1)
print('Number of divisors of the second number is', divisor2)
```

# 과제 4-2 리뷰

- 3번 문항

- 하단 함수를 작성하고 호출해 다이아몬드 모양으로 별(\*)을 출력

- printStarDia(n):** 다이아몬드 모양을 출력
    - printStarDia 함수 안에서 반드시 중첩된 반복문(for 혹은 while)을 사용

```
def printStarDia(n):  
    for numStars in range(1, n+1):          # 각 줄에 출력될 '*' 개수  
        for i in range(n-numStars):  
            print(' ', end='')              # 앞부분 공백 출력  
        for i in range(numStars):  
            print('* ', end='')              # '*' ' ' 출력  
        print()  
  
    for numStars in range(n, 0, -1):  
        for i in range(n-numStars):  
            print(' ', end='')              # 앞부분 공백 출력  
        for i in range(numStars):  
            print('* ', end='')              # '*' ' ' 출력  
        print()
```

(실행 예)

5↵

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

```
maxStars = int(input())  
printStarDia(maxStars)
```

# 과제 5-1

- 제출 안내
  - **제출 기한: 4월 7일 (수) 23시 59분** (git push 완료 시점 기준)
  - 프로젝트 하위에 <과제번호>/<문제번호>/<답안 파일> 구조를 갖도록 제출
- 프로젝트 구조 예시

```
+ 2021_ITE1014_2021000001
+ 5-1/
+ 1/
+   - 1.py
+ 2/
+   - 2.py
+ 3/
+   - 3.py
+ 4/
+   - 4.py
+ 5/
+   - 5.py
```

# 과제 5-1

- 1번 문항
  - 3명의 학생 이름과 시험 점수(정수값)를 차례로 입력
  - 학생 이름을 입력하면 해당 학생의 시험 점수를 출력 후 종료
  - 점수 정보가 등록되지 않은 학생 이름이 입력되면 별도 메시지 출력 후 종료
  - 반복문을 이용하지 않고, Dictionary를 사용해 구현
  - 제출 파일: 1.py

(실행 예 1)

```
Tom 90↵  
Amy 80↵  
Bob 85↵  
Which student's score?  
Bob  
Bob's score: 85
```

(실행 예 2)

```
Tom 90↵  
Amy 80↵  
Bob 85↵  
Which student's score?  
Nancy  
Nancy is not in the database.
```

# 과제 5-1

- 2번 문항
  - 1~1000 범위에서 100개의 난수를 생성해 하나의 List에 저장
  - for문을 이용해 이 List 안의 숫자들을 한 칸의 공백으로 구분하며 모두 출력
  - for문을 이용해 이 List 안에서 가장 값이 큰 숫자를 max() 함수 호출 없이 찾아 출력
  - 제출 파일: 2.py

(실행 예)

```
766 130 704 821 206 333 264 400 26 560 973 607 64 76 591 8 307 374
390 231 509 345 846 694 245 835 6 893 146 615 941 358 897 795 921
870 504 784 109 461 266 439 227 727 750 13 407 386 616 176 846 465
451 160 985 742 817 378 673 784 186 626 470 921 871 401 506 193 68
1 482 545 741 567 120 831 270 296 663 561 733 976 577 261 998 655
906 860 672 688 505 402 129 384 619 820 625 985 93 908 362
```

```
max value: 998
```



# 과제 5-1

- 3번 문항
  - 공백으로 구분된 여러 단어로 이루어진 문자열을 하나 입력
  - 문자열 내 각 단어들의 등장 횟수 출력
  - 반드시 `str.count()` 함수 호출 없이 List와 Dictionary를 사용할 것
  - 제출 파일: 3.py

(실행 예)

```
what are you doing doing how are doing do you wanna doing↵  
what: 1  
are: 2  
doing: 4  
how: 1  
you: 2  
do: 1  
wanna: 1
```

# 과제 5-1

- 4번 문항
  - 문자열의 길이를 입력
  - 영어 소문자만으로 이뤄진 해당 길이만큼의 무작위 문자열 하나를 출력
    - `getRandomString(leng)`: 길이가 `leng`인 무작위 문자열을 생성해 반환하는 함수 작성
  - Hint
    - `str.join(ls)`: `str`의 내용을 구분자로 하여 List `ls`의 모든 문자열을 이어 붙인 새 문자열을 반환
    - `chr(i)`: ASCII 코드 숫자값 `i`가 나타내는 알파벳 문자 하나를 담은 문자열을 반환
  - 제출 파일: `4.py`

(실행 예 1)

10 ↵

dmbzkndzga

(실행 예 2)

5 ↵

mghwl

# 과제 5-1

- 5번 문항
  - 치즈 크기(문자열의 길이)를 입력
  - 해당 길이만큼의 알파벳으로 이뤄진 무작위 치즈(문자열)를 생성 (4번 문항 함수 활용)
  - 주는 치즈를 먹을 때 한 번에 한 종류의 특정 알파벳이 쓰여진 치즈만을 탐색
  - 쥐가 치즈를 먹고 나면 해당 알파벳이 있던 자리는 모두 '\_'로 대체
  - 쥐가 치즈를 최대 10번 먹는 동안, 치즈를 먹을 때마다 다음 정보들을 출력
    - 이번에 먹은 치즈의 알파벳
    - 이번 것을 포함해 지금까지 먹어온 치즈들의 각 알파벳 목록
    - 기존 치즈가 나타내던 원본 문자열
    - 현재 치즈 상태가 나타내는 문자열
  - 자세한 내용은 과제 명세 및 실행 예제 참고
  - 제출 파일: 5.py

# 과제 5-1

- 5번 문항 실행 예제

```
Input the length of the string : 20
Generated Cheese is 'jmeksuascztmzxwloghx'
Mouse starts eating!!
Start eating 'z'
Eaten alphabet of cheese : z
Original cheese : jmeksuascztmzxwloghx
Current cheese status :  jmeksuasc_t_m_xwloghx

Start eating 'n'
Eaten alphabet of cheese : z n
Original cheese : jmeksuascztmzxwloghx
Current cheese status :  jmeksuasc_t_m_xwloghx

Start eating 'e'
Eaten alphabet of cheese : z n e
Original cheese : jmeksuascztmzxwloghx
Current cheese status :  jm_ksuasc_t_m_xwloghx

Start eating 'c'
Eaten alphabet of cheese : z n e c
Original cheese : jmeksuascztmzxwloghx
Current cheese status :  jm_ksuasc_t_m_xwloghx

Start eating 'd'
Eaten alphabet of cheese : z n e c d
Original cheese : jmeksuascztmzxwloghx
Current cheese status :  jm_ksuasc_t_m_xwloghx

Start eating 'y'
Eaten alphabet of cheese : z n e c d y
Original cheese : jmeksuascztmzxwloghx
Current cheese status :  jm_ksuasc_t_m_xwloghx
```

```
Start eating 'y'
Eaten alphabet of cheese : z n e c d y
Original cheese : jmeksuascztmzxwloghx
Current cheese status :  jm_ksuasc_t_m_xwloghx
```

```
Start eating 'g'
Eaten alphabet of cheese : z n e c d y g
Original cheese : jmeksuascztmzxwloghx
Current cheese status :  jm_ksuasc_t_m_xwlo_hx
```

```
Start eating 'p'
Eaten alphabet of cheese : z n e c d y g p
Original cheese : jmeksuascztmzxwloghx
Current cheese status :  jm_ksuasc_t_m_xwlo_hx
```

```
Start eating 'x'
Eaten alphabet of cheese : z n e c d y g p x
Original cheese : jmeksuascztmzxwloghx
Current cheese status :  jm_ksuasc_t_m_wlo_h_
```

```
Start eating 'k'
Eaten alphabet of cheese : z n e c d y g p x k
Original cheese : jmeksuascztmzxwloghx
Current cheese status :  jm__suasc_t_m_wlo_h_
```

```
Finally remained Cheese Status :
Eaten alphabet of cheese : z n e c d y g p x k
Original cheese : jmeksuascztmzxwloghx
Current cheese status :  jm__suasc_t_m_wlo_h_
```

---

**수고하셨습니다.**