

| 연습문제 |

- 2.1 구구단 6단을 계산하여 결과를 순서대로 출력하는 코드를 for 문을 이용하여 구현하라.
- 2.2 위 문제를 while 문을 이용하여 다시 구현하라.
- 2.3 구구단 6단을 계산 결과를 역순으로 출력하는 코드를 for와 range를 이용하여 구현하라.
- 2.4 섭씨온도를 화씨온도로 변환하는 함수를 구현하라. 섭씨온도($^{\circ}\text{C}$)를 화씨온도($^{\circ}\text{F}$)로 변환하는 수식은 $^{\circ}\text{F} = 32 + \frac{180}{100} \times ^{\circ}\text{C}$ 이다.
- 2.5 화씨온도를 섭씨온도로 변환하는 함수를 구현하라.
- 2.6 $A = [1, 2, 3, 4]$ 와 같이 리스트를 선언하고 이 리스트의 모든 값을 역순으로 출력하는 코드를 구현하라. 단, 인덱스 값으로는 음수만을 사용하라.
- 2.7 리스트의 모든 값을 더해 결과를 반환하는 함수를 구현하라.
- 2.8 문자열 `msg = "Data Structures in Python"`를 선언하고, 이 문자열을 먼저 그대로 출력하고, 다음으로 모두 대문자로 바꿔서 출력하고, 마지막으로 모두 소문자로 바꿔 출력하는 코드를 작성하라.
- 2.9 어느 식당의 음식에 대한 가격 정보를 다음과 같이 딕셔너리로 표현하자.
`price = { '콩나물해장국':4500, '갈비탕':9000, '돈가스':8000 }`
 여기에 새로운 메뉴 '팻타이'를 7000원에 추가하고, 모든 메뉴와 가격을 출력하는 코드를 작성해 보라.
- 2.10 위 문제에서 모든 메뉴의 가격을 500원 내리는 코드를 작성해 보라.

2.11 1부터 n 까지의 숫자를 전부 합하여 반환하는 순환적인 함수를 작성하라.

$$1 + 2 + 3 + \dots + n$$

2.12 다음을 계산하는 순환적인 함수를 작성하라.

$$1 + 1/2 + 1/3 + \dots + 1/n$$

2.13 이항계수(binomial coefficient)를 계산하는 순환함수를 작성하라. 이항계수는 다음과 같이 순환적으로 정의된다.

$${}_nC_k = \begin{cases} {}_{n-1}C_{k-1} + {}_{n-1}C_k & \text{if } 0 < k < n \\ 1 & \text{if } k=0 \text{ or } k=n \end{cases}$$

2.14 이항계수를 구하는 함수를 반복 구조로 구현하라.

2.15 문자열의 내용을 반대로 바꾸는 순환적인 함수 `reverse()`를 구현하라. 예를 들어 `reverse("ABCDE")`는 "EDCBA"를 반환해야 한다.

2.16 순환을 사용하여 1부터 n 까지의 숫자를 화면에 순서대로 출력하는 함수 `printNum(n)`과 역순으로 출력하는 함수 `printRevNum(n)`을 작성하라. 다음은 각각 `printNum(10)`과 `printRevNum(10)`의 출력 결과이다.

```
1 2 3 4 5 6 7 8 9 10
```

```
10 9 8 7 6 5 4 3 2 1
```

2.17 순환적인 방법으로 피보나치수열을 호출하였을 때 함수가 중복되어 호출되는 것을 확인할 수 있도록 각 함수의 매개 변수별 호출 빈도를 측정해 출력하라.

예) $n=10$ 을 넣었을 때

`Fibo(10)` = 1번

`Fibo(9)` = ??번

...

`Fibo(0)` = ??번

실습문제

P2.1 다음은 근로소득세율 표를 보여주고 있다. 소득을 입력하면 세금을 계산하고 세금과 세 후 소득을 출력하는 프로그램을 작성하라. 수입이 조금이라도 많으면 세금을 세 후 소득이 더 많아야 한다. 예를 들어, 소득이 2,000만 원이라면 1,200만 원까지는 6%를 내고, 다음 구간(1,200만 원~4,600만 원)에 포함되는 800만 원에 대해서는 15%를 내야 하므로, 전체 세금은 $1200 \times 6\% + 800 \times 5\%$ 이다.

소득	근로소득세율
1,200만 원 이하	6%
1,200만 원 ~ 4,600만 원	15%
4,600만 원 ~ 8,800만 원	24%
8,800만 원 ~ 1억 5,000만 원	35%
1억 5,000만 원 초과	38%

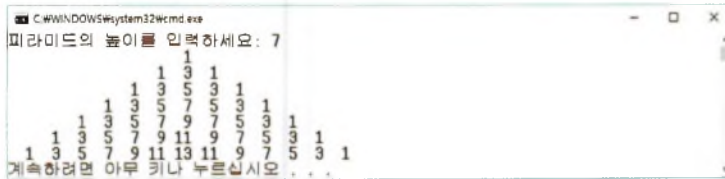
P2.2 번호 맞추기 게임(Up-and-Down 게임)을 구현하자. 숨겨진 두 자리의 숫자를 추측하여 맞추는 것이다. 게이머가 숫자를 예측하면 컴퓨터는 정답과 비교하여 “더 큰 숫자입니다”나 “더 작은 숫자입니다” 그리고 맞힌 경우 “정답입니다”를 출력한다. 중간에 맞히거나 10번 동안 맞히지 못하면 게임이 끝난다.

- 정답을 `answer` 추측 문자를 `guess`이라 하면, `answer`와 `guess`를 비교하여 결과를 출력하면 된다. 정답 범위를 힌트로 제공하기 위해 `min`과 `max` 변수를 사용한다.
- 반복문으로는 `for`를 사용하고 최대 10번 반복하면서, 중간에 정답을 맞히면 `break` 문을 이용해 루프를 빠져나와 게임을 종료한다.

```

C:\WINDOWS\system32\cmd.exe
숫자를 입력하세요(범위:0~99): 50
아닙니다. 더 큰 숫자입니다!
숫자를 입력하세요(범위:50~99): 70
아닙니다. 더 작은 숫자입니다!
숫자를 입력하세요(범위:50~70): 60
아닙니다. 더 큰 숫자입니다!
숫자를 입력하세요(범위:60~70): 65
아닙니다. 더 큰 숫자입니다!
숫자를 입력하세요(범위:65~70): 67
정답입니다. 5번 만에 맞추셨습니다.
게임이 끝났습니다.
계속하려면 아무 키나 누르십시오 . . .
  
```

P2.3 높이를 입력받아 다음과 같은 숫자 피라미드를 출력하는 다음 게임(?)을 만들어보자. 먼저 숫자가 출력되는 규칙을 잘 찾아보라.

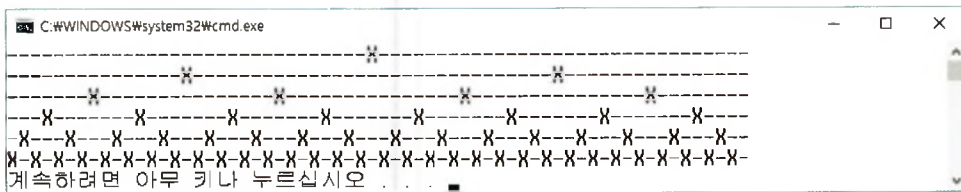


- 2중 루프를 사용한다. 바깥 루프는 세로를 나타내고 높이만큼 반복하면 된다.
- 내부 루프는 한 줄에서의 위치를 나타낸다. 한 줄에서의 출력은 세 부분으로 나누어야 할 것이다. 맨 처음 공백을 적절히 출력한다. 다음으로 숫자가 증가하는 방향으로 출력하고, 마지막으로 감소하는 방향으로 숫자를 출력한다.

P2.4 다음과 같은 모양을 출력하는 순환적인 함수를 작성하여 하라. 이 함수의 원형(prototype)은 다음과 같다.

```
def draw_tree( row, left, right ) :
```

- row: X를 그리는 행을 표시한다. 가장 위에 있는 행이 0이고 아래로 내려갈수록 숫자는 증가한다고 생각하자.
- left와 right: 각각 주어진 영역의 왼쪽 끝과 오른쪽 끝을 나타낸다.

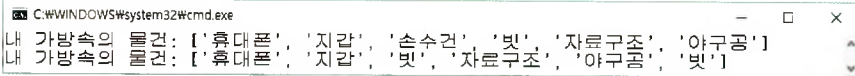


P2.5 1.2절에서 Bag의 추상 자료형을 정의하고 함수를 이용해 구현하였다. 사실 추상 자료형을 구현하는 가장 좋은 방법은 클래스를 사용하는 것이다. 함수로 구현된 Bag을 클래스로 다시 구현하라. 다음은 구현된 Bag 클래스를 사용하는 코드의 예이다. 실행 결과는 1장 예서와 동일하다.

클래스로 구현한 Bag 사용 코드

```
myBag = Bag()                                # 새로운 Bag 객체를 만들
myBag.insert('휴대폰')                       # 객체의 insert()메소드 호출
myBag.insert('지갑')                         # myBag에 '지갑'을 삽입
myBag.insert('손수건')
myBag.insert('빗')
myBag.insert('자료구조')
myBag.insert('야구공')
print('내 가방속의 물건:', myBag.bag)

myBag.insert('빗')
myBag.remove('손수건')                       # 객체의 remove()메소드 호출
print('내 가방속의 물건:', myBag.bag)
```



```
C:\WINDOWS\system32\cmd.exe
내 가방속의 물건: ['휴대폰', '지갑', '손수건', '빗', '자료구조', '야구공']
내 가방속의 물건: ['휴대폰', '지갑', '빗', '자료구조', '야구공', '빗']
```