

컴퓨팅 사고와 SW 코딩

02

파이썬 코딩

경북대학교 배준현 교수
(joonion@knu.ac.kr)



- 문제해결을 위한 파이썬 코딩
 - 입력과 출력: `input()`, `print()`
 - 자료형, 변수, 연산자: *duck typing*, operators(arithmetic, logical)
 - 조건문과 반복문: if-else, while, for-loop
 - 함수: user-defined functions, *functional programming*(map, lambda)
 - 내장형 자료구조: list, tuple, set, dict



02. 파이썬 코딩

■ BOJ 2753: 윤년

• 문제:

- 윤년은 연도가 4의 배수이면서 100의 배수가 아닐 때, 또는, 400의 배수일 때이다.
- 연도가 주어졌을 때, 윤년이면 1, 아니면 0을 출력하는 프로그램을 작성하시오.
 - 예를 들어, 2012년은 4의 배수이면서 100의 배수가 아니라서 윤년이다.
 - 1900년은 100의 배수이고 400의 배수는 아니기 때문에 윤년이 아니다.
 - 하지만, 2000년은 400의 배수이기 때문에 윤년이다.

• 입력:

- 첫째 줄에 연도가 주어진다.
- 연도는 1보다 크거나 같고, 4000보다 작거나 같은 자연수이다.

• 출력:

- 첫째 줄에 윤년이면 1, 아니면 0을 출력한다.



02. 파이썬 코딩

예제 입력

2000

1999

예제 출력

1

0



02. 파이썬 코딩

```
def solve(y):  
    if y % 4 == 0 and y % 100 != 0:  
        return True  
    elif y % 400 == 0:  
        return True  
    else:  
        return False  
  
Y = int(input())  
if solve(Y):  
    print(1)  
else:  
    print(0)
```





02. 파이썬 코딩

- 오리의 구분과 동적 자료형: *duck typing*
 - 오리를 구분하는 방법: *duck test*
 - If it looks like a duck and quacks like a duck, it must be a duck.
 - 정적 자료형과 동적 자료형:
 - 정적 자료형: *static* typing
 - 변수를 선언할 때 그 변수의 자료형을 결정: C, C++, Java, etc.
 - 동적 자료형: *dynamic* typing
 - 변수를 사용할 때 그 변수의 자료형을 결정: Python, JavaScript, etc.



02. 파이썬 코딩

```
class Duck:
    def say(self):
        print("꽹꽹")

class Chicken:
    def say(self):
        print("꼬끼오")

def say(animal):
    animal.say()

say(Duck())
say(Chicken())
```



■ BOJ 2884: 알람 시계

- 문제:

- 알람 시계의 알람을 알람 시각보다 45분 일찍 설정하고자 한다.
- 알람 시각이 주어졌을 때, 위의 방법으로 설정 시간을 구하시오.

- 입력:

- 첫째 줄에 두 정수 H와 M이 주어진다. ($0 \leq H \leq 23$, $0 \leq M \leq 59$)
- 입력 시간은 24시간 표현을 사용한다.
- 하루의 시작은 0:0(자정)이고, 끝은 23:59(다음날 자정 1분 전)이다.
- 시간을 나타낼 때, 불필요한 0은 사용하지 않는다.

- 출력:

- 첫째 줄에 설정해야 하는 알람 시간을 출력한다.
- 입력과 같은 형태로 출력하면 된다.



02. 파이썬 코딩

예제 입력

10 10

0 30

23 40

예제 출력

9 25

23 45

22 55



02. 파이썬 코딩

```
def solve(h, m):  
    if m >= 45:  
        m = m - 45  
    else:  
        m = (m - 45) % 60  
        h = (h - 1) % 24  
    return h, m
```

```
H, M = map(int, input().split())  
h, m = solve(H, M)  
print(h, m)
```

※ 주의: 이 문제는 반드시 C 언어로도 풀어볼 것.

- 값이 나옴



02. 파이썬 코딩

■ 함수형 프로그래밍

- 절차형 프로그래밍: *procedural* programming
 - 자료를 처리하는 여러 모듈(프로시저, 서브루틴, 함수)로 프로그램을 작성
 - 변수(객체)와 함수(메서드)는 서로 분리되어 있음
- 함수형 프로그래밍: *functional* programming
 - 자료를 처리하는 여러 함수로 프로그램을 작성
 - 변수(상태)와 함수(순수 함수)가 서로 결합되어 있음 (일급 시민)
 - 함수를 변수처럼 다른 함수의 매개변수, 리턴값으로 사용가능함



람다 함수: 란다 대수에서 유래된 이름으로 일종의 익명 함수



02. 파이썬 코딩

```
a = list(range(1, 11))
```

```
f = lambda x: x**2  
print(list(map(f, a)))
```

```
g = lambda x: x*2  
print(list(map(g, a)))
```

```
★for y in map(lambda x: f(x) if x % 2 == 0 else g(x), a):  
    print(y, end = ' ')
```



02. 파이썬 코딩

■ BOJ 2231: 분해합

- 문제:
 - 어떤 자연수 N 의 분해합은 N 과 N 을 이루는 각 자리수의 합을 의미한다.
 - 어떤 자연수 M 의 분해합이 N 인 경우, M 을 N 의 생성자라 한다.
 - 예를 들어, 245의 분해합은 $256(=245+2+4+5)$ 이 된다.
 - 따라서 245는 256의 생성자가 된다.
 - 물론, 어떤 자연수의 경우에는 생성자가 없을 수도 있다.
 - 반대로, 생성자가 여러 개인 자연수도 있을 수 있다.
 - 자연수 N 이 주어졌을 때, N 의 가장 작은 생성자를 구하시오.
- 입력: 첫째 줄에 자연수 $N(1 \leq N \leq 1,000,000)$ 이 주어진다.
- 출력: 첫째 줄에 답을 출력한다. 생성자가 없는 경우에는 0을 출력한다.



02. 파이썬 코딩

예제 입력

216

예제 출력

198



02. 파이썬 코딩

```
def sumofdigits(n):
```

```
    s, t = 0, n
```

```
    while t > 0:
```

```
        s += t % 10
```

```
        t = t // 10
```

```
    return s
```

```
def solve(n):
```

```
    for m in range(1, n + 1):
```

```
        if n == m + sumofdigits(m):
```

```
            return m
```

```
    return 0
```

```
N = int(input())
```

```
print(solve(N))
```

```
def solve(n):
```

```
    for m in range(1, n + 1):
```

```
        if n == m + sum(map(int, str(m))):
```

```
            return m
```

```
    return 0
```



02. 파이썬 코딩

■ BOJ 2738: 행렬 덧셈

- 문제:
 - $N \times M$ 크기의 두 행렬 A와 B가 주어졌을 때, 두 행렬을 더하는 프로그램을 작성하시오.
- 입력:
 - 첫째 줄에 행렬의 크기 N 과 M이 주어진다.
 - 둘째 줄부터 N개의 줄에 행렬 A의 원소 M개가 차례대로 주어진다.
 - 이어서 N개의 줄에 행렬 B의 원소 M개가 차례대로 주어진다.
 - N과 M은 100보다 작거나 같고, 행렬의 원소는 절댓값이 100보다 작거나 같은 정수이다.
- 출력:
 - 첫째 줄부터 N개의 줄에 행렬 A와 B를 더한 행렬을 출력한다.
 - 행렬의 각 원소는 공백으로 구분한다.



02. 파이썬 코딩

예제 입력

3 3
1 1 1
2 2 2
0 1 0
3 3 3
4 4 4
5 5 100

예제 출력

4 4 4
6 6 6
5 6 100



02. 파이썬 코딩

```
def solve(n, m, A, B):  
    C = [[0] * m for _ in range(n)]  
    for i in range(n):  
        for j in range(m):  
            C[i][j] = A[i][j] + B[i][j]  
    return C  
  
N, M = map(int, input().split())  
A = [list(map(int, input().split())) for _ in range(N)]  
B = [list(map(int, input().split())) for _ in range(N)]  
C = solve(N, M, A, B)  
for i in range(N):  
    print(" ".join(map(str, C[i])))
```



02. 파이썬 코딩

■ 리스트 포함: *list comprehension*

- 간편하게 리스트를 생성하는 파이썬 문법

- <https://docs.python.org/3/tutorial/datastructures.html#list-comprehensions>

```
squares = []  
for x in range(10):  
    squares.append(x ** 2)  
print(squares)
```

```
squares = list(map(lambda x: x**2, range(10)))  
print(squares)
```

```
squares = [x**2 for x in range(10)]  
print(squares)
```





02. 파이썬 코딩

```
M1 = [(i, j) for j in range(3) for i in range(2)]  
print(M1)
```

```
M2 = [[(i, j) for j in range(3)] for i in range(2)]  
print(M2)
```

```
a = [i for i in range(1, 11)]  
b = [j for j in a if a[-1] % j == 0]  
print(b)
```



02. 파이썬 코딩

■ BOJ 1110: 더하기 사이클

- N이 주어졌을 때, **N의 사이클의 길이**를 구하는 프로그램을 작성하시오.
 - N가 10보다 작다면 앞에 0을 붙여 두 자리 수로 만들고, **각 자리의 숫자를 더한다.**
 - 그 다음, N의 가장 **오른쪽 자리 수와 앞에서 구한 합의 가장 오른쪽 자리 수를 이어 붙이면 새로운 수를 만들 수 있다.**
- 예를 들어, 26부터 시작한다.
 - $2+6 = 8$ 이다. 새로운 수는 68이다.
 - $6+8 = 14$ 이다. 새로운 수는 84이다.
 - $8+4 = 12$ 이다. 새로운 수는 42이다.
 - $4+2 = 6$ 이다. 새로운 수는 26이다.
 - 위의 예는 4번만에 원래 수로 돌아올 수 있다. 따라서 26의 사이클의 길이는 4이다.
- 입력: 첫째 줄에 N이 주어진다. N은 0보다 크거나 같고, 99보다 작거나 같은 정수이다.
- 출력: 첫째 줄에 N의 사이클 길이를 출력한다.



02. 파이썬 코딩

예제 입력

26

55

1

0

71

예제 출력

4

3

60

1

12



02. 파이썬 코딩

```
def solve(n):  
    m, cnt = n, 1  
    while True:  
        s = sum(map(int, list(str(m))))  
        m = (m % 10) * 10 + (s % 10)  
        if m == n:  
            break  
        cnt += 1  
    return cnt  
  
N = int(input())  
print(solve(N))
```





02. 파이썬 코딩

■ SWAP: 두 변수의 교환

- C/C++: 포인터, 레퍼런스 연산자의 활용

```
#include <stdio.h>

void swap(int *a, int *b) {
    int t = *a;
    *a = *b;
    *b = t;
}

int main() {
    int a, b;
    scanf("%d %d", &a, &b);
    swap(&a, &b);
    printf("%d %d\n", a, b);
}
```

```
#include <iostream>
using namespace std;

void swap(int &a, int &b) {
    int t = a;
    a = b;
    b = t;
}

int main() {
    int a, b;
    cin >> a >> b;
    swap(a, b);
    cout << a << " " << b << endl;
}
```




02. 파이썬 코딩

- Python: 다중 할당문의 활용

```
a, b = map(int, input().split())  
a, b = b, a  
print(a, b)
```



02. 파이썬 코딩

■ 연습문제 2.1:

- 임시 변수를 쓰지 않고 SWAP을 할 수 있을까?
 - 덧셈과 뺄셈만을 이용하여 SWAP 구현하기
 - 곱셈과 나눗셈만을 이용하여 SWAP 구현하기
 - XOR 연산을 이용하여 SWAP 구현하기



02. 파이썬 코딩

■ 더 풀어볼 문제:

- BOJ 1308: D-Day
- BOJ 2525: 오븐 시계
- BOJ 2475: 검증수
- BOJ 2740: 행렬 곱셈
- BOJ 1977: 완전제곱수

Any Questions?

