

1장 프로그래밍 실습_파이썬 코드

[프로그래밍 실습 1]

```
n = int(input("정점의 개수를 입력하라.: ")) # 정점의 수

isEulerGraph = True

# 각 정점의 차수를 입력받기
for i in range(1, n + 1):
    print("정점", i, "의 차수를 입력하라.: ", end = "")
    degree = int(input()) # 차수의 수

    # 차수가 홀수이면 해당 그래프는 오일러 그래프가 아니다
    if degree % 2 == 1:
        isEulerGraph = False

if isEulerGraph:
    print("\n오일러 그래프이다!")
else:
    print("\n오일러 그래프가 아니다!")
```

[프로그래밍 실습 2]

```
# addStrings 함수가 두 개의 문자열로 표현된 값을 더해서 반환
def addStrings(num1: str, num2: str) -> str:
    res = [] # 반환될 결과값

    carry = 0 # 받아올림수를 위한 변수
    p1 = len(num1) - 1 # 첫 번째 문자열을 위한 인덱스 변수
    p2 = len(num2) - 1 # 두 번째 문자열을 위한 인덱스 변수

    while p1 >= 0 or p2 >= 0:
        x1 = ord(num1[p1]) - ord('0') if p1 >= 0 else 0 # num1의 p1번째 문자를 정수로 x1에 저장
        x2 = ord(num2[p2]) - ord('0') if p2 >= 0 else 0 # num2의 p2번째 문자를 정수로 x2에 저장

        value = (x1 + x2 + carry) % 10 # x1+x2+carry에서의 받아올림수를 제외한 값
        carry = (x1 + x2 + carry) // 10 # x1+x2+carry에서의 받아올림수
        res.append(value) # 결과값 끝에 value를 추가
        p1 -= 1
        p2 -= 1

    # 받아올림수가 0이 아닐 때 결과값 끝에 추가
    if carry:
        res.append(carry)

    # 결과값을 역순으로 문자열로 바꾸고 반환
    return ''.join(str(x) for x in res[::-1])

# multiply 함수가 두 개의 문자열로 표현된 값을 곱해서 반환
def multiply(num1: str, num2: str) -> str:
```

```

if num1 == "0" or num2 == "0":
    return "0"

N = len(num1) + len(num2)          # 결과의 길이를 위한 변수
answer = [0] * N                  # N개의 0으로 초기화된 배열
first_number = num1[::-1]         # num1을 역순으로 저장
second_number = num2[::-1]        # num2를 역순으로 저장

# 해당 이중 for 문은 두 개의 정수에 대한 곱하기를 처리
for place2, digit2 in enumerate(second_number):
    for place1, digit1 in enumerate(first_number):
        # second_number[place2]*first_number[place1]의 결과가 저장될 answer의 인덱스값
        num_zeros = place1 + place2

        carry = answer[num_zeros] # 기존 answer[num_zeros]의 값을 받아올림수로 저장
        multiplication = int(digit1) * int(digit2) + carry

        answer[num_zeros] = multiplication % 10 # multiplication에서의 받아올림수를
                                                # 제외한 값

        answer[num_zeros + 1] += multiplication // 10

if answer[-1] == 0:                # answer 안에 0인 원소들을 제거
    answer.pop()

return ''.join(str(digit) for digit in reversed(answer)) # answer을 역순으로 바꾸고
                                                         # 문자열형으로 반환

def main():
    binary = input("2진수를 입력하시오: ")
    binary = binary[::-1]          # 이진수(문자열)를 역순으로 변경
    answer = "0"
    multiplier = "1"              # 각 비트를 곱하기 위한 변수

    for b in binary:
        if b == '1':              # 비트가 1일 때만 answer과 multiplier 값을 더함
            answer = addStrings(answer, multiplier)
            multiplier = multiply(multiplier, "2") # 매번 multiplier에 2를 곱함

    print("10진수: ", answer)      # 계산된 결과인 10진수를 출력

if __name__ == "__main__":
    main()                        # main 함수 호출

```

[프로그래밍 실습 3]

```
from __future__ import print_function
from sys import stdin
import time                # sleep 함수를 포함한 라이브러리

def printf(str, *args):
    print(str % args, end='')

# my_opening 함수는 처음으로 화면에 안내를 출력
def my_opening():
    printf("# 지금부터 여러분을 신비한 마술의 세계로 초대한다.\n\n\n")
    printf("1 - 31의 숫자 중 마음에 드는 숫자를 생각하라. \n\n")

# my_card 함수가 index번째 카드에 있는 숫자들을 화면에 출력해 사용자부터 0이나 1을 입력받음
def my_card(index):
    cardname = chr(ord('A') + index - 1)

    while True:
        printf(" ----- %s 카드 -----\n" % cardname )
        # 해당 for 문은 index번째 카드에 있는 숫자들을 출력
        for i in range(0, 4):
            for j in range(0, 4):
                printf("%7d" % card[index-1][4*i + j])
            printf("\n")
        printf(" -----\n")
        printf("\n\n")
        printf("%s 카드에 생각한 숫자가 있다면 YES(1번), 없다면 NO(0번)를 선택하라. : \n" %
cardname)
        result = int(stdin.readline()) # 사용자가 생각한 수가 index번째 카드에 있으면 1을,
                                       # 아니면 0을 입력하면 그것을 a에 저장

        printf("\n\n")
        if result>-1 and result<2:    # 사용자가 입력한 문자가 0이나 1이면 a를 반환하고,
                                       # 잘못된 입력이면 다시 index번째 카드를 출력

            return result

# my_res 함수는 사용자가 생각한 숫자를 출력
def my_res(a,b,c,d,e):
    '''2진수를 10진수로 변환하는 것처럼 사용자로부터 얻은 a,b,c,d,e에 대해 계산을 진행, 즉
a,b,c,d,e 비트열이라고 가정해 이를 10진수로 변경하면 사용자가 생각한 숫자가 나타남'''
    res = (e*2*2*2*2) + (d*2*2*2) + (c*2*2) + (b*2) + (a)
    printf("당신이 마음에 드는 숫자는 %d이다.\n\n\n" % res);
    printf("신기하지 않은가?\n");

# 5개의 카드를 위한 2차원 배열 정의 및 초기화
card = [
    [1,3,5,7,9,11,13,15,17,19,21,23,25,27,29,31],
    [2,3,6,7,10,11,14,15,18,19,22,23,26,27,30,31],
    [4,5,6,7,12,13,14,15,20,21,22,23,28,29,30,31],
    [8,9,10,11,12,13,14,15,24,25,26,27,28,29,30,31],
    [16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31]
]

my_opening()
# 이 함수로 화면에 안내를 출력

time.sleep(3)
```

```
printf("\n생각했는가? 시작하려면 엔터를 클릭하라.")
stdin.readline()

print("\n\n")
a = my_card(1)
    # 사용자에게 첫 번째 카드를 보여주고 1이나 0을 a에 저장
b = my_card(2)
    # 사용자에게 두 번째 카드를 보여주고 1이나 0을 b에 저장
c = my_card(3)
    # 사용자에게 세 번째 카드를 보여주고 1이나 0을 c에 저장
d = my_card(4)
    # 사용자에게 네 번째 카드를 보여주고 1이나 0을 d에 저장
e = my_card(5)
    # 사용자에게 다섯 번째 카드를 보여주고 1이나 0을 e에 저장

my_res(a,b,c,d,e)    # 사용자가 생각한 숫자를 계산해서 화면에 출력
```