

# [IT CookBook, 파이썬 for Beginner(2판)]

## 연습문제 해답

본 자료의 저작권은 저자 우재남과 한빛아카데미(주)에 있습니다. 이 자료는 강의 보조자료로 제공 되는 것으로, 학생들에게 배포되어서는 안 됩니다.

### 1 장

1. 다음 중 컴퓨터와 관련 없는 것을 하나 고르시오. **사람의 언어**

프로그래밍 언어, 소프트웨어, 앱, 프로그래머, 사람의 언어

2. 다음 중 프로그래밍 언어가 아닌 것을 하나 고르시오. **B++**

파이썬, Java, B++, C, HTML, PHP

3. 파이썬의 특징에 대한 설명 중 틀린 것은?

- ① 프로그래밍 언어이다.
- ② 비단뱀이라는 뜻이다.
- ③ 귀도 반 로섬이라는 프로그래머가 만들었다.
- ④ 다른 프로그래밍 언어에 비해서 가장 전통이 오래되어 그 기능이 막강하다.
- ⑤ 개인에게 상당히 인기가 있지만, 기업 시장에는 아직 진출하지 못했다.
- ⑥ 개인 사용자에게는 무료이지만, 학교나 기업 등은 유료로 사용해야 한다.

4. 파이썬 설치에 대한 설명 중 틀린 것은?

- ① 윈도우용 설치 파일의 확장자는 exe이다.
- ② 설치 과정이 매우 간단하다.
- ③ 윈도우, 맥, 리눅스 등에 모두 설치할 수 있다.
- ④ 설치 과정에서 설치되는 폴더를 바꿀 수 없다.

5. 다음과 출력 결과가 나오도록 프로그램을 작성하시오.

`print("아... 파이썬 너무 재미있어요")`

아... 파이썬 너무 재미있어요^^

6. 다음 코드의 출력 결과를 쓰시오.

`"11+22-33"`

## Syntax Error

```
print("11+22-33")
print(안녕? 파이썬)
```

## 2 장

1. 다음 중 파이썬 문법이 틀린 것을 모두 고르시오.

- ① a = 1000
- ② b = 500.0
- ③ a =
- ④ 10 = a

2. print() 함수의 결과를 예측하시오.

- ① print("하하하") 하하하
- ② print(200 + 100) 300
- ③ print("200 + 100") 200+100
- ④ print(int("200") + 100) 300

3. 다음 중 틀린 설명을 모두 고르시오. 다음 설명 중에서 틀린 것을 모두 고르시오.

- ① 대화형 모드에서 사용한 변수는 IDLE을 종료했다가 다시 실행해도 남아 있다.
- ② 변수는 기본적으로 메모리라는 공간에 저장된다.
- ③ \*.py로 저장한 파이썬 코드는 다음에 재사용할 수 있다.
- ④ 스크립트 모드에서 코드를 한 줄씩 입력하면 바로 실행된다.

4. 다음 코드를 실행한 후 111과 222를 입력한다면 어떤 내용이 출력될지 설명하시오. 111222

```
var1=input()
var2=input()
result=var1+var2
print(var1 , "+" , var2 , "=", result)
```

5. 코드를 실행한 후 111과 222를 입력하면 오류가 발생한다. 오류가 발생하는 이유를 간단히 설명하시오. input() 함수는 모든 것을 문자로 입력 받는다. 그런데 문자끼리의 곱셈은 허용되지 않기 때문에 오류가 발생한다.

```
var1=input()
var2=input()
result=var1*var2
```

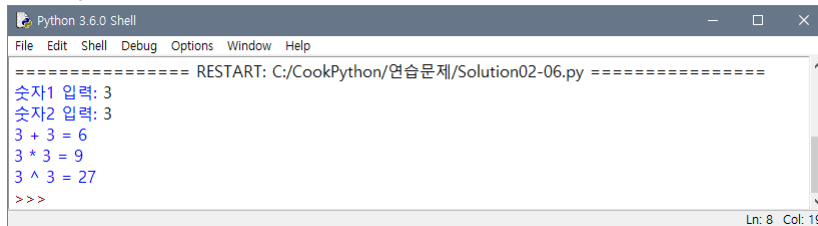
```
print(var1 , "*" , var2 , "=" , result)
```

6. 5에서 코드가 오류 없이 실행되려면 셋째 줄을 어떻게 바꾸어야 할지 설명하시오.

```
result=int(var1)*int(var2)
```

7. 정수 2개를 입력받아서 더하기, 곱하기, 제곱 연산을 하는 프로그램을 작성하시오.

힌트) pow(밑수,지수) 함수는 제곱의 결과를 구한다.



```
Python 3.6.0 Shell
File Edit Shell Debug Options Window Help
===== RESTART: C:/CookPython/연습문제/Solution02-06.py =====
숫자1 입력: 3
숫자2 입력: 3
3 + 3 = 6
3 * 3 = 9
3 ^ 3 = 27
>>>
```

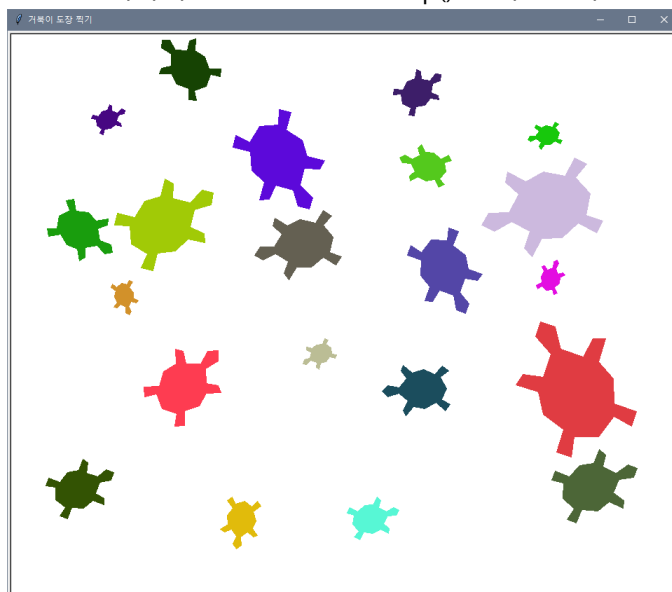
Solution02-07.py

```
a=int(input("숫자1 입력: "))
b=int(input("숫자2 입력: "))
result=a+b
print(a , "+" , b , "=" , result)
result=a*b
print(a , "*" , b , "=" , result)
result=pow(a,b)
print(a , "^" , b , "=" , result)
```

8. [심화문제] 화면에 마우스 왼쪽 버튼을 클릭하면 클릭한 위치에 랜덤한 색상, 크기, 각도의 거북이 도장이 찍히는 프로그램을 작성하자.

힌트) 1. 거북이의 색상은 turtle.color(r, g, b)로 설정한다.

2. 거북이 도장은 turtle.stamp()를 사용한다.



Solution02-08.py

```
import turtle
import random

## 함수 선언 부분 ##

def screenLeftClick(x,y):
    tSize = random.randrange(2,10)
    turtle.shapesize(tSize)
    r = random.random()
    g = random.random()
    b = random.random()
    turtle.color((r, g, b))
    tAngle = random.randrange(0, 360)

    turtle.penup()
    turtle.goto(x,y)
    turtle.left(tAngle)
    turtle.stamp()

## 변수 선언 부분 ##
tSize, tAngle = 0, 0
r, g, b = 0.0, 0.0, 0.0

## 메인 코드 부분 ##
turtle.title('거북이 도장 찍기')
turtle.shape('turtle')

turtle.onscreenclick(screenLeftClick,1)

turtle.done()
```

## 3 장

1. 실행하면 오류가 발생하는 것을 모두 고르시오

- ① `print("500 + 500")`
- ② `print("%d" % (500 + 500))`
- ③ `print("%d" % (500, 600))`
- ④ `print("%d %d" % (700))`

2. `print()` 함수의 결과를 예측하시오.

- ① `print("%d / %d = %d" % (10, 4, 10 / 4))` 10 4 2
- ② `print("%d / %d = %f " % (10, 4, 10 / 4))` 10 4 2.500000
- ③ `print("%d / %d = %5.1f " % (10, 4, 10 / 4))` 10 4 2.5
- ④ `print("%d / %d = %5.0f " % (10, 4, 10 / 4))` 10 4 2

3. `print()` 함수의 결과를 예측하시오.

- ① `print("%05d" % 543)` 00543
- ② `print("%10s" % "파이썬")` 빈칸후Python
- ③ `print("%1.1f " % 123.45)` 123.5

4. 이스케이프 문자에 대한 설명 중 틀린 것을 모두 고르시오.

- ① `\n`은 새로운 줄로 넘기며, `\t` 을 누른 효과를 준다.
- ② `\t`는 다음 탭으로 이동한다.
- ③ `\w`는 `w`를 출력한다.
- ④ `\b`는 뽀 소리를 발생시킨다.

5. 변수에 대한 설명으로 틀린 것을 모두 고르시오.

- ① 파이썬은 변수의 선언을 생략해도 된다.
- ② 변수 종류에는 정수형, 실수형, 불형, 문자열 등이 있다.
- ③ `a=b`처럼 같이 변수에 변수를 대입할 수 없다.
- ④ `type()` 함수는 변수에 저장된 값을 출력한다.

6. 코드를 실행하면 오류가 발생한다. 그 이유를 설명하시오. 중간에 변수가 아닌 숫자가 들어 있기 때문이다.

`a=b=10=c=d=20`

7. 각 진수를 10진수로 변환하시오.

- ① 2진수 0011 3
- ② 2진수 01010 10
- ③ 16진수 11 17
- ④ 8진수 17 15

8. 오류가 발생하는 것을 모두 고르고, 그 이유를 간단히 설명하시오. 1

- ① `int('1002', 2)` 2진수는 0~1만 표현한다.
- ② `int('1008', 8)` 8진수는 0~7만 표현한다.
- ③ `int('AAFG', 16)` 16진수는 0~F만 표현한다.

9. 코드의 출력값을 예측하시오.

```
bin(12); hex(12); oct(12)
```

'0b1100'

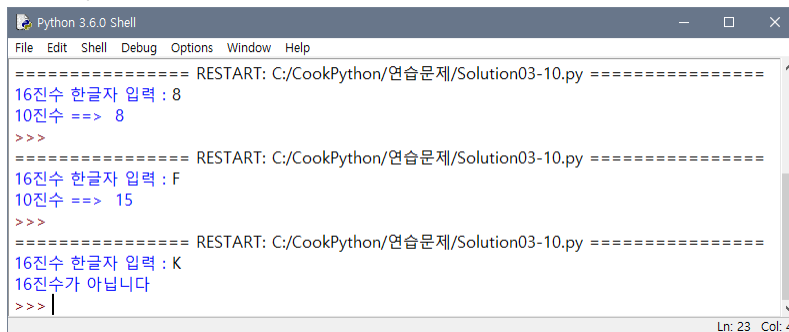
'0xc'

'0o14'

10. **[심화문제]** 16진수 글자 하나를 입력하면 16진수인지 아닌지를 구분하는 코드를 작성해 보자.

힌트1) if ~ else 문을 사용하자 (5장에서 다룬다)

힌트2) 둘 다 만족해야 하는 and와 둘 중에 하나만 만족하면 되는 or를 활용하자. (4장에서 다룬다)



```
Python 3.6.0 Shell
File Edit Shell Debug Options Window Help
===== RESTART: C:/CookPython/연습문제/Solution03-10.py =====
16진수 한글자 입력 : 8
10진수 ==> 8
>>>
===== RESTART: C:/CookPython/연습문제/Solution03-10.py =====
16진수 한글자 입력 : F
10진수 ==> 15
>>>
===== RESTART: C:/CookPython/연습문제/Solution03-10.py =====
16진수 한글자 입력 : K
16진수가 아닙니다
>>>
```

Solution03-10.py

```
num=input("16진수 한 글자 입력 : ")
```

```
if ('0' < num and num < '9') or ('a' <= num and num <= 'f') or ('A' <= num and num <= 'F') :
    print("10진수 ==> ", int(num, 16))
```

```
else :
```

```
    print("16진수가 아닙니다")
```

## 4 장

1. 다음 중 산술 연산자가 아닌 것을 모두 고르시오. **& ^ ~ \*\*\* ?**

+ - \* ^ ~ / // % \*\* & \*\*\* ?

2. 다음 계산식의 결과를 예측하시오. **30 -10 10 0**

a,b = 10,20

① a+b ② a-b ③ a%b ④ a//b

3. 다음 계산식의 결과를 예측하시오. **3 -1 1.5**

a, b, c = 1, 2, 3

① a+b%c ② a\*b-c ③ a/b\*c

4. 문자열을 숫자로 변환한 후, 계산하는 식이다. 오류가 발생하는 것을 고르고, 오류가 발생하지 않도록 수정하시오. **② float(s2) + 111.11**

s1, s2, s3 = "111", "111.11", "9999999999999999"

① int(s1) + 111.11

② int(s2) + 111.11

③ int(s3) + 111.11

5. 대입 연산자의 활용이다. 결과를 예측하시오. **30 10 0**

① a,b = 10,20; a+=b; print(a)

② a,b = 10,20; a%=b; print(a)

③ a,b = 10,20; a//=b; print(a)

6. 다음 중 논리 연산자가 아닌 것을 모두 고르시오. **nor xor equal**

and or not nor xor equal

7. 비트 연산자의 활용이다. 결과를 16진수로 예측하시오. **0x0000 0xFFFF 0xFFFF**

0xFF00 & 0x00FF

0xFF00 | 0x00FF

0xFF00 ^ 0x00FF

8. 비트 시프트 연산자의 활용이다. 결과를 예측하시오. **100**

a=100; a=a<<100 ; a=a>>100; print(a)

9. 입력한 금액을 5만원, 1만원, 5천원, 1천원, 500원, 100원, 50원, 10원 동전으로 교환하는 프로그램의 출력 결과다. 프로그램을 작성해 보자.

```

Python 3.6.0 Shell
File Edit Shell Debug Options Window Help
===== RESTART: C:/CookPython/연습문제/Solution04-11.py =====
교환할 돈은 얼마 ? 777777
50000원 15장, 10000원 2장, 5000원 1장, 1000원 2장
500원 1개, 100원 2개, 50원 1개, 10원 2개
바꾸지 못한 돈 ==> 7원
>>> |
Ln: 58 Col: 4

```

## Solution04-09.py

```

## 변수 선언 부분
money, p50, p10, p5, p1, c500, c100, c50, c10 = 0,0,0,0,0,0,0,0,0,0

## 메인(main) 코드 부분
money=int(input("교환할 돈은 얼마 ? "))

p50 = money // 50000; money %= 50000
p10= money // 10000; money %= 10000
p5 = money // 5000; money %= 5000
p1 = money // 1000; money %= 1000

c500 = money // 500; money %= 500
c100 = money // 100; money %= 100
c50 = money // 50; money %= 50
c10 = money // 10; money %= 10

print(" 50000원 %d장, 10000원 %d장, 5000원 %d장, 1000원 %d장" % (p50, p10, p5, p1))
print(" 500원 %d개, 100원 %d개, 50원 %d개, 10원 %d개" % (c500, c100, c50, c10))
print(" 바꾸지 못한 돈 ==> %d원 \n"% money)

```

10. [심화문제] 응용 예제2를 활용해서 비트 논리곱을 구현한다. 숫자를 2개 입력 받아서 각 숫자에 대한 2진수와 비트 논리곱의 결과 2진수를 출력하는 프로그램을 작성하자. 예로 123과 456을 입력하면 1111011 & 111001000의 결과 1001000이 차례로 출력된 결과는 다음과 같다.





#### Solution04-10.py

```
import turtle

## 전역 변수 선언 부분 ##
num = 0
swidth, sheight= 1000, 300
curX, curY = 0, 0

## 메인 코드 부분 ##
if __name__ == "__main__" :
    turtle.title('거북이로 두 숫자 비트 논리곱(&) 연산하기')
    turtle.shape('turtle')
    turtle.setup(width=swidth+50, height=sheight+50)
    turtle.screensize(swidth, sheight)
    turtle.penup()
    turtle.left(90)

    num1=int(input("숫자1을 입력하세요 : "))
    num2=int(input("숫자2을 입력하세요 : "))
    num3= num1 & num2

    binary = bin(num1)
    curX = swidth / 2
    curY = 100
    for i in range(len(binary)-2) :
        turtle.goto( curX, curY )
        if num1 & 1 :
            turtle.color('red')
            turtle.turtlesize②
        else :
            turtle.color('blue')
            turtle.turtlesize①
        turtle.stamp()
        curX -= 50
        num1 >>= 1

    binary = bin(num2)
```

```

curX = swidth / 2
curY = 0
for i in range(len(binary)-2) :
    turtle.goto( curX, curY )
    if num2 & 1 :
        turtle.color('red')
        turtle.turtlesize②
    else :
        turtle.color('blue')
        turtle.turtlesize①
    turtle.stamp()
    curX -= 50
    num2 >>= 1

binary = bin(num3)
curX = swidth / 2
curY = -100
for i in range(len(binary)-2) :
    turtle.goto( curX, curY )
    if num3 & 1 :
        turtle.color('red')
        turtle.turtlesize②
    else :
        turtle.color('blue')
        turtle.turtlesize①
    turtle.stamp()
    curX -= 50
    num3 >>= 1

turtle.done()

```

## 5 장

1. if문의 기본 구조 이다. 빈 칸을 제시된 예로 채우시오. 조건식, 실행할 문장1, 실행할 문장2

실행할 문장1, 실행할 문장2, 조건식

```
if  :  
      
else :  
    
```

2. 다음 if문의 결과를 예측하시오. **100이군요.**

```
a=100  
if a != 100 :  
    print("100이 아니군요")  
else :  
    print("100이군요")
```

3. 다음 if문의 결과를 예측하시오. **아무것도 안나옴.**

```
a=123  
if a == 100 :  
    print("100이군요")  
    print("100이 아니군요")
```

4. 입력한 숫자가 3의 배수인지를 판단하는 프로그램이다. 빈 곳을 채우시오. **a%3 ==0**

```
a = int(input("정수를 입력하세요 : "))  
  
if  :  
    print("3의 배수를 입력했군요.")  
else :  
    print("3의 배수가 아니군요.")
```

5. 다음 if문의 결과를 예측하시오. **적당한 수입입니다.**

```
a=123  
if a > 100 :  
    if a > 200 :  
        print("완전히 큰 수입입니다.")  
    else :
```

```

        print("적당한 수입입니다.")
else :
    print("완전히 작은 수입입니다.")
    print("프로그램 끝입니다.")

```

6 다음 중첩 if문을 elif를 사용하는 코드로 변경해 보자.

```

score=55

if score >= 60 :
    print("합격이다.")
else :
    if score >= 40 :
        print("불합격 이지만 과락은 아닙니다.")
    else :
        print("불합격 이면서 과락입니다.")

```

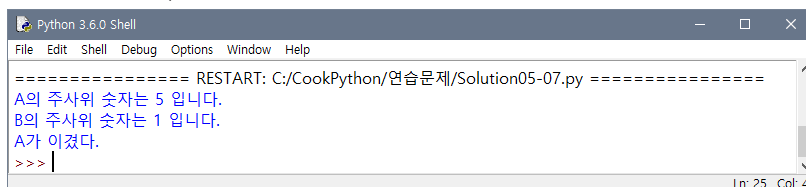
```

score=55

if score >= 60 :
    print("합격이다.")
elif score >= 40 :
    print("불합격 이지만 과락은 아닙니다.")
else :
    print("불합격 이면서 과락입니다.")

```

7. 두 사람이 주사위를 던져서 높은 숫자가 나오면 이기는 게임이다. 출력결과는 A가 이기거나, B가 이기거나, 비기는 결과가 나와야 한다. 코드를 작성해 보자.



Solution05-07.py

```

import random

num_a = random.randrange(1, 7)

```

```

print ("A의 주사위 숫자는 %d 입니다." % num_a)

num_b = random.randrange(1, 7)
print ("B의 주사위 숫자는 %d 입니다." % num_b)

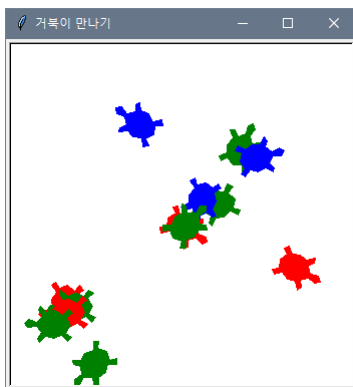
if num_a > num_b :
    print("A가 이겼다.")
elif num_a < num_b :
    print("B가 이겼다.")
else :
    print("둘이 비겼네요.")

```

8. [심화문제] 응용 예제2를 변경해서 거북이 3마리가 화면에 돌아다닐 때, 서로 만나면 만난 위치에 도장을 찍고 계속 움직이는 프로그램을 작성한다. 즉, 무한루프로 반복된다. 만약, 거북이가 화면 밖으로 나가면, 다시 각자의 초기 위치(거북이1은 (-100,-100), 거북이2는 (0,0), 거북이3은 (100,100))로 옮겨온다.

힌트1) 도장은 turtle.stamp() 함수를 사용한다.

힌트2) 거북이의 속도를 올리려면 turtle.spped(10)을 사용하면 된다.



Solution05-08.py

```

import turtle
import math
import random

## 전역 변수 선언 부분 ##
t1, t2, t3 = [None] * 3
t1X, t1Y, t2X, t2Y, t3X, t3Y = [0] * 6
swidth, sheight=300, 300

```

```

## 메인 코드 부분 ##
if __name__ == "__main__" :
    turtle.title('거북이 만나기')
    turtle.setup(width=swidth+50, height=sheight+50)
    turtle.screensize(swidth, sheight)

    t1 = turtle.Turtle('turtle');    t1.color('red');    t1.penup();
    t2 = turtle.Turtle('turtle');    t2.color('green');  t2.penup();
    t3 = turtle.Turtle('turtle');    t3.color('blue');   t3.penup();

    t1.goto(-100,-100);    t2.goto(0,0);    t3.goto(100,100)
    t1.speed(10); t2.speed(10); t3.speed(10);
    t1.turtlesize(2); t2.turtlesize(2); t3.turtlesize(2)

    while True :

        angle = random.randrange(0,360)
        dist = random.randrange(1,50)
        t1.left(angle);        t1.forward(dist)
        angle = random.randrange(0,360)
        dist = random.randrange(1,50)
        t2.left(angle);        t2.forward(dist)
        angle = random.randrange(0,360)
        dist = random.randrange(1,50)
        t3.left(angle);        t3.forward(dist)

        t1X = t1.xcor();  t1Y = t1.ycor()
        t2X = t2.xcor();  t2Y = t2.ycor()
        t3X = t3.xcor();  t3Y = t3.ycor()

        if not ( (-swidth/2 < t1X and t1X <= swidth/2) and (-sheight/2 < t1Y and t1Y <=
sheight/2) ):
            t1.goto(-100,-100)
        if not ( (-swidth/2 < t2X and t2X <= swidth/2) and (-sheight/2 < t2Y and t2Y <=
sheight/2) ):
            t2.goto(0,0)
        if not ( (-swidth/2 < t3X and t3X <= swidth/2) and (-sheight/2 < t3Y and t3Y <=

```

```

sheight/2) ):
    t3.goto(100,100)

    if math.sqrt( ((t1X-t2X)*(t1X-t2X)) +((t1Y-t2Y)*(t1Y-t2Y)) )  <= 20 :
        t1.stamp(); t2.stamp()
    if math.sqrt( ((t1X-t3X)*(t1X-t3X)) +((t1Y-t3Y)*(t1Y-t3Y)) )  <= 20 :
        t1.stamp(); t3.stamp()
    if math.sqrt( ((t2X-t3X)*(t2X-t3X)) +((t2Y-t3Y)*(t2Y-t3Y)) ) <= 20 :
        t2.stamp(); t3.stamp()

turtle.done()

```

## 6 장

1. for문의 기본 구조이다. 빈 칸을 제시된 예로 채우시오. ①시작값 ②끝값+1 ③증가값

시작값, 증가값, 끝값, 끝값+1,

```

for 변수 in range( , ,  ) :
    이 부분을 반복

```

2. 몇 번이 출력될지 예측하시오. 99번

```

for i in range(1, 100, 1) :
    print("파이썬 완전 꿀잼이에요~~. ^^")

```

3. 100부터 200까지 2의 배수만 더하는 프로그램이다. 빈 곳을 채워보자.

`range(100, 201, 2)`

```
i, hap = 0, 0
```

```
for i in,  :
```

```
    hap = hap + i
```

```
print("100부터 200까지 2의 배수 합계 : %d" % hap)
```

4. 몇 번이 출력될지 예측하시오. 81번

```
for i in range(1,10) :
```

```
    for k in range (1, 10) :
```

```
        print("파이썬 정말 꿀잼이에요~~. ^^")
```

5. 구구단의 단을 입력하면 단이 다음과 같이 거꾸로 출력되도록 프로그램을 만들어 보자.

힌트 : Code06-07.py을 참조하고, range() 함수를 수정하면 된다.

[출력 결과]

몇 단 ? 9 ← 사용자가 입력한 값

9 X 9 = 81

9 X 8 = 72

~~ 중간 생략 ~~

9 X 1 = 9

```
i, dan = 0, 0
```

```
dan = int(input(" 몇 단 ? "))
```

```
for i in range(9, 0, -1) :
```

```
    print(" %d X %d = %2d" % (dan, i, dan*i))
```

6. for문으로 5회 출력하는 0에서 100까지 2의 배수의 합계를 구하는 프로그램이다. 이 프로그램을 while문으로 수정해 보자.

힌트 : Code06-10.py을 참조하자.

```
i, hap = 0, 0
```

```
for i in range(0, 101, 2) :
```

```
    hap = hap + i
```

```
print("0에서 100까지의 짝수의 합 : %d" % hap)
```

[출력 결과]



0에서 100까지의 짝수의 합 : 2550

```
i, hap = 0, 0
for i in range(0, 101, 2) :
    hap = hap + i
print("0에서 100까지의 짝수의 합 : %d" % hap)
```

7. 1부터 1000 까지의 홀수의 합계 중에서, 최초로 1000이 넘는 숫자는 어떤 것이는 구하는 프로그램을 작성해 보자.

힌트 : Code06-14.py를 참조하자.

[출력 결과]

1~100의 홀수의 합에서 최초로 1000이 넘는 위치 : 63

```
hap,i = 0,0

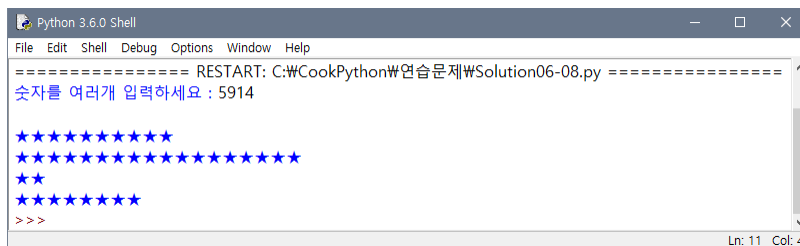
for i in range(1,101,2) :
    hap += i

    if hap >= 1000 :
        break

print("1~100의 홀수의 합에서 최초로 1000이 넘는 위치 : %d" % i)
```

8. 응용 예제1을 수정해서 입력한 숫자의 2배만큼 별모양이 출력되도록 코드를 작성 해보자. 예로 "5914"를 입력하면 다음과 같이 출력되도록 한다.

힌트 : 별모양의 코드는 \u2605를 사용하자.



```
Python 3.6.0 Shell
File Edit Shell Debug Options Window Help
===== RESTART: C:\CookPython\연습문제\Solution06-08.py =====
숫자를 여러개 입력하세요 : 5914

★★★★★★★★
★★★★★★★★★★★★★★★★
★★
★★★★★★★★
>>>
```

Solution06-08.py

```
## 변수 선언 부분
i, k, heartNum = 0, 0, 0
numStr, ch, heartStr = "", "", ""
```

```

## 메인(main) 코드 부분
numStr=input("숫자를 여러개 입력하세요 : ")
print("")

i=0
ch = numStr[i]
while True :
    heartNum = int(ch)

    heartStr = ""
    for k in range (0, heartNum*2) :
        heartStr += "Wu2605"
        k += 1

    print(heartStr)

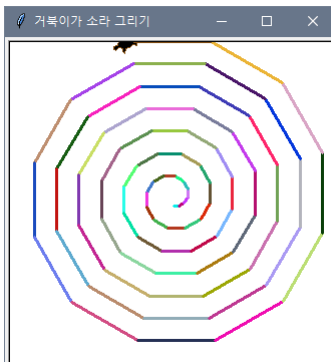
    i += 1
    if (i > len(numStr) -1 ) :
        break

    ch = numStr[i]

```

9. **[심화문제]** 거북이가 화면 중앙에서 출발해서 80개의 선분을 다음 그림과 같이 소라 모양으로 그리도록 코드를 작성해 보자.

힌트) 초기 선분의 길이는 5로 시작해서 1씩 증가하게 만든다. 각도는 30도씩 회전하면서 그린다.



Solution06-09.py

```

import turtle
import random

```

```

## 전역 변수 선언 부분 ##
swidth, sheight, pSize = 300, 300, 3
r, g, b, dist = [0] * 4

## 메인 코드 부분 ##
turtle.title('거북이가 소라 그리기')
turtle.shape('turtle')
turtle.pensize(pSize)
turtle.setup(width=swidth+30, height=sheight+30)
turtle.screensize(swidth, sheight)

dist = 5
for i in range(1, 80) :

    r = random.random();
    g = random.random()
    b = random.random()
    turtle.pencolor((r, g, b))

    turtle.forward(dist)
    turtle.left(30)
    dist += 1

```

## 7 장

1. 리스트를 만들고 출력하는 방식이다. 오류가 나는 부분을 고르자. **aa[3]**

```

aa = [ 10, 20, 30 ]
print(aa[0])
print(aa[1])
print(aa[2])
print(aa[3])

```

2. 100개의 빈 리스트를 만들고 길이를 구하는 코드다. 빈 부분을 예에서 골라서 채우시오. ①**for**  
②**range** ③**append**

```

append, for, range

```

```
myList = []
① i in ② (0, 100) :
    myList.③(0)
len(myList)
```

3. 사용자가 3개의 숫자를 입력하고 입력한 숫자의 합계를 구하는 프로그램이다. 빈 부분을 for문으로 완성해 보자. `for k in range(0, 3)`

```
myList = []
for i in range(0, 3) :
    myList.append(0)
hap = 0

for i in range(0, 3) :
    myList[i] = int(input(str(i + 1) + "번째 숫자 :"))

    hap = hap + myList[k]

print("합계 ==> %d" % hap)
```

4. 비어 있는 자료구조를 초기화 하기 위한 코드이다. 서로 맞는 것끼리 줄로 이어 보자.

① 나 ② 다 ③ 가 ④ 라

① aa = [ ]  
② bb= { }  
③ cc = ( )  
④ dd = None

가. 튜플  
나. 리스트  
다. 딕셔너리  
라. 일반변수

5. 다음과 같은 리스트가 있을 때, 각 문제의 출력되는 결과를 예측하시오. 출력 결과가 하나의 값일 수 도 있고, 리스트 일수도 있으니 잘 생각해 보자.

① '지민' ② '민아' ③ ['설현'] ④ ['민아', '찬미'] (5) ['초아', '지민'] (6) ['초아', '유경', '찬미']

aoa=['설현', '초아', '지민', '유나', '유경', '혜정', '민아', '찬미']

① aoa[2] ② aoa[-2] ③ aoa[0:1] ④ aoa[6:] (5) aoa[-7:-5] (6) aoa[1::3]

6. 리스트 조작 함수에 대한 설명이다. 보기에서 고르자.

① extend ② remove ③ reverse ④ pop

```
append(), pop(), sort(), reverse(), index(), insert(), remove(), extend(), count()
```

- ① 리스트의 뒤에 리스트를 추가한다.
- ② 리스트에서 지정한 값을 제거한다.
- ③ 리스트 항목의 순서를 역순으로 만든다.
- ④ 리스트 제일 뒤의 항목을 빼내고, 빼낸 항목은 삭제한다.

7. 2차원 리스트의 예이다. 각 문제의 출력되는 결과를 예측하시오.

① 2 ② 8 ③ [9, 10, 11, 12]

```
myList = [ [ 1, 2, 3, 4] ,  
           [5, 6, 7, 8] ,  
           [9, 10, 11, 12] ]
```

- ① myList [0][1]
- ② myList [1][3]
- ③ myList [2]

8. 다음을 실행하면 myData에 저장된 값은 몇 개인가?

3개

```
myData = { 1, 1, 1, 2, 2, 3, 3, 3 }  
myData.add①
```

9. 다음을 실행하면 myData에 저장된 값은?

[2, 8]

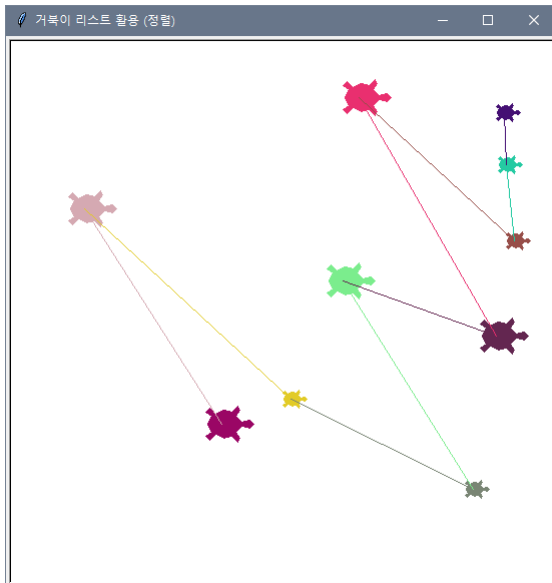
```
import math  
myData = [ int(math.pow(2,num)) for num in range(0, 5) if num%2 != 0]
```

10. **[심화문제]** [프로그램 2]를 수정해서 거북이의 X, Y좌표가 가장 작은 거북이부터 정렬한 후, 거북이끼리 선을 긋는다. 예를 들면 (10, 10), (-100, -50), (50, 10) 세 좌표의 거북이를 X+Y 값으로 정렬하면 (-100, 50), (10, 10), (50, 10)이 된다. 이 세 거북이를 선으로 이으면 된다. 다음 그림은 10마리 거북이를 좌표로 정렬한 후에 선을 그은 결과다.

힌트1) 각 거북이 리스트의 2번째인 X좌표와 3번째값인 Y좌표를 더한 결과로 playerTurtles를 오름차순 정렬해야 한다.

힌트2) 시작 거북이는 정렬 후의 가장 첫번째 거북이로 해야 한다.

힌트3) 거북이 사이에 선을 긋기 위해서는 앞 거북이의 좌표를 저장하고 있어야 한다.



Solution07-10.py

```
import turtle
import random

## 전역 변수 선언 부분 ##
myTurtle , tX, tY, tColor, tSize, tShape= [None] * 6
shapeList = []
playerTurtles = [] # 거북이 2차원 리스트.
swidth, sheight=500, 500

## 메인 코드 부분 ##
if __name__ == "__main__" :
    turtle.title('거북이 리스트 활용 (정렬)')
    turtle.setup(width=swidth+50, height=sheight+50)
    turtle.screensize(swidth, sheight)

    #shapeList = turtle.getshapes()
    for i in range(0, 10) :
        random.shuffle(shapeList)
        myTurtle = turtle.Turtle('turtle')
        tX = random.randrange(-swidth/2, swidth/2)
        tY = random.randrange(-sheight/2, sheight/2)
        r = random.random() ; g = random.random(); b = random.random()
        tSize = random.randrange(1, 3)
        playerTurtles.append( [myTurtle, tX, tY, tSize, r, g, b])
```

```

# tX + tY로 정렬
for i in range(0, len(playerTurtles)-1) :
    for k in range(i+1, len(playerTurtles)) :
        iVal = playerTurtles[i][1] + playerTurtles[i][2]
        kVal = playerTurtles[k][1] + playerTurtles[k][2]
        if iVal > kVal :
            playerTurtles[i], playerTurtles[k] = playerTurtles[k], playerTurtles[i]

saveX, saveY = playerTurtles[0][1], playerTurtles[0][2]
for tList in playerTurtles :
    myTurtle = tList[0]
    myTurtle.penup()
    myTurtle.goto(saveX, saveY)
    myTurtle.color( (tList[4], tList[5], tList[6]))
    myTurtle.pencolor(( tList[4], tList[5], tList[6]))
    myTurtle.turtlesize(tList[3])
    myTurtle.pendown()
    myTurtle.goto(tList[1], tList[2])
    saveX = tList[1]
    saveY = tList[2]

turtle.done()

```

## 8 장

1. 문자열을 추출하는 코드이다. 출력되는 내용을 예측하시오. P, y, ho, hon

```

ss='Python'
print(ss[0])
print(ss[1:2])
print(ss[-3:-1])
print(ss[3:])

```

2. 다음과 같은 문자열이 있을 때 오류가 발생하는 것을 모두 고르시오. ② ④

```

ss1='Python'
ss2='Good'

```

- ① ss1 + ss2
- ② ss1 - ss2
- ③ ss1 \* 3
- ④ ss1 / ss2

3. 문자열의 중간마다 #을 끼워넣는 코드이다. 빈 부분을 채우시오.

① for ② len ③ ss[i]

```
ss = 'Python'

[1] i in range(0, [2]) :
    print([3] + '$', end = '')
```

4. 문자열을 반대의 순서로 출력하는 프로그램이다. 빈 곳을 채우시오.

inStr[strLen - (i+1)]

```
inStr, outStr = "Python", ""
strLen = len(inStr)

for i in range(0, strLen) :
    outStr += [ ]
print("내용을 거꾸로 출력 --> %s" % outStr)
```

5. 다음 각 문자열 함수의 결과를 예측하시오.

① HANBIT ② hanbit ③ hANBIT ④ Hanbit

```
myStr = 'Hanbit'
```

- ① myStr.upper()
- ② myStr.lower()
- ③ myStr.swapcase()
- ④ myStr.title();

6. 다음 각 문자열 함수의 결과를 예측하시오. 3, 14, 7, True

```
myStr = 'Hanbit Media, Hanbit Academy, Hanbit Life'
```

- ① myStr.count('Hanbit')
- ② myStr.find('Hanbit', 2)



- ③ myStr.rindex('Media')
- ④ myStr.startswith('Hanbit')

7. 입력된 문자열 중에서 숫자를 제거하는 프로그램을 작성하자.

예) 문자열 --> IT CookBook 1234 파이썬

숫자 제거 --> IT CookBook 파이썬

Solution08-07.py

```
inStr, outStr = "", ""

inStr = input('문자열 -->')

for i in range(0, len(inStr)) :
    if inStr[i].isdigit() == False :
        outStr += inStr[i]

print('숫자 제거 --> ' + outStr)
```

8. 입력한 문자열에서 숫자, 영문 소문자, 영문 대문자, 한글, 기타 문자의 개수를 세는 프로그램을 작성하자.

힌트) 각 문자는 고유한 번호가 할당되어 있으며, ord() 함수로 번호를 확인할 수 있다.

[출력 결과]

문자열을 입력하세요 : IT CookBook 파이썬을 공부하고 있습니다. ^\_\_^

대문자 : 4 소문자: 6 숫자: 0 한글: 12 기타: 11

Solution08-08.py

```
## 전역 변수 선언 부분
numCnt, lowerCnt, upperCnt, hanCnt, etcCnt = [0] * 5
ch = ""

## 메인(main) 코드 부분
inStr = input("문자열을 입력하세요 : ")

for ch in inStr :
    if ( ord(ch) >= ord("A") and ord(ch) <= ord("Z")) :
        upperCnt += 1
    elif ( ord(ch) >= ord("a") and ord(ch) <= ord("z")) :
        lowerCnt += 1
```

```

elif ( ord(ch) >= ord("0") and ord(ch) <= ord("9")) :
    numCnt += 1
elif ( ord(ch) >= ord("가") and ord(ch) <= ord("힉")) :
    hanCnt += 1
else :
    etcCnt += 1

print("대문자 :", upperCnt, " 소문자:", lowerCnt, " 숫자:", numCnt, " 한글:", hanCnt, " 기타:",
etcCnt)

```

9. **[심화문제]** [프로그램 1]를 변경해서 다음과 같이 글자가 둥그렇게 써지도록 코딩해 보자.

힌트 1) 중심에서 반지름은 100정도로 한다. 또 글자의 크기는 20정도로 고정한다.

힌트 2) 360도에서 문자열의 개수만큼 나눠서 각도를 회전시키면서 글자를 쓴다.

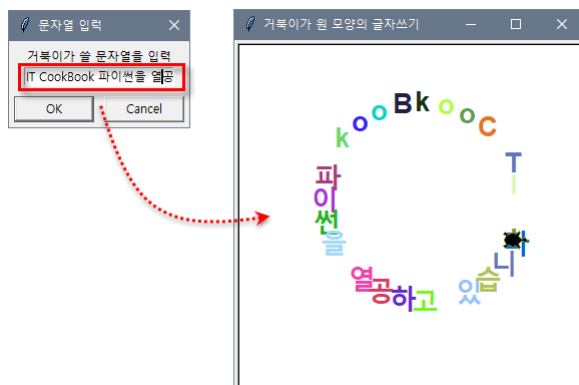
힌트 3) 거리와 각도를 통해서 좌표를 구하는 공식은 다음과 같다.

$X\text{좌표} = \text{거리} * \cos(\text{각도})$

$Y\text{좌표} = \text{거리} * \sin(\text{각도})$

힌트 4)  $\cos()$  및  $\sin()$  함수의 각도는 라디안 값으로 처리해야 한다.

$\text{라디안} = 3.14 * \text{각도} / 180$



Solution08-09.py

```

import turtle
import random
from tkinter.simpledialog import *
import math

## 전역 변수 선언 부분 ##
inStr = ""
swidth, sheight=300, 300
tX, tY, txtSize = [0] * 3

```

```

## 메인 코드 부분 ##
turtle.title('거북이가 원 모양의 글자쓰기')
turtle.shape('turtle')
turtle.setup(width=swidth+50, height=sheight+50)
turtle.screensize(swidth, sheight)
turtle.penup()

inStr = askstring('문자열 입력', '거북이가 쓸 문자열을 입력')

dist= 100
angle = 0
value = int( 360 / len(inStr))

for ch in inStr :

    rad = 3.141592 * angle / 180
    tX = dist * math.cos(rad)
    tY = dist * math.sin(rad)
    angle += value
    r = random.random() ; g = random.random(); b = random.random()
    txtSize = random.randrange(20, 21)

    turtle.goto(tX,tY)

    turtle.pencolor( (r,g,b) )
    turtle.write(ch, font=('맑은고딕', txtSize, 'bold'))

turtle.done()

```

## 9 장

1. 다음 중 함수의 설명 중 맞는 것을 모두 고르자. 1, 2, 3
  - ① 한번 만들어 놓으면 호출만 하면 된다.
  - ② 내부의 작동을 신경쓰지 않는다는 의미로 블랙박스라고도 부른다.
  - ③ 커피 자판기는 함수의 좋은 비유다.
  - ④ 함수는 여러 번 호출하는 것보다는 한번만 호출해서 사용하는 것이 효율적이다.

2. 두 수를 더해서 결과를 반환하는 함수이다. 빈 부분을 채우시오. ① def ② return

```
① sum(v1, v2) :  
    result = 0  
    result = v1 + v2  
    ② result
```

3. 위 3번의 함수를 호출하는 방법이다. 옳은 것을 하나 고르자. 4

- ① hap = sum()
- ② hap = sum(10)
- ③ hap = sum (10 20)
- ④ hap = sum (10, 20)
- (5) hap = sum (10 : 20)

4. 함수의 반환값과 관련된 내용이다. 틀린 것을 모두 고르자. 4

- ① 함수에는 반환값이 없어도 된다.
- ② 함수에서 반환할 수 있는 값의 개수는 1개 뿐이다.
- ③ return 문을 사용한다.
- ④ 반환값이 없을 때도 retrun문을 써줘야 한다.

5. myFunc.py에 fun1(), fun2()가 정의되어 있다고 가정하겠다. 다음 중에서 오류가 발생하는 코드를 모두 고르자. 1, 3

- ① import myFunc.py  
myFunc.fun1()
- ② import myFunc  
myFunc.fun1()
- ③ from myFunc import fun1()  
fun1()
- ④ from myFunc import \*  
fun2()

6. 다음 func() 함수를 호출할 때, 오류가 발생하는 것을 고르자. 4

```
def func(v1, v2=0, v3=0) :  
    result = 0  
    result = v1 + v2 + v3  
    return result
```

- ① func①
- ② func(1, 2)
- ③ func(0, 0, 0)
- ④ func( )

7. 전달받은 매개변수의 숫자와 관계없이 모든 매개변수를 곱하는 함수이다. 빈 부분을 채우시오.

① \*var ② num ③ result

```
def multiply(  ) :
    result = 0
    for  in var :
        result =  * num

    return result
```

8. 다음 코드를 실행하면 출력되는 값을 차례로 써보자. 222, 111

```
a = 222

def func1() :
    print(a)

def func2() :
    a = 111
    print(a)

func1()
func2()
```

9. 다음 코드의 결과는 무엇이 출력되는가?

9

Solution09-09.py

```
def myRange(start, end, hop = 1) :
    retVal = start
    while retVal <= end :
        yield retVal
        retVal += hop
```

```

hap = 0
for i in myRange(1, 5, 2) :
    hap += i
print(hap)

```

10. [심화문제] 재귀 호출 함수를 이용해서 피보나치 수열을 구현하라.

[출력 결과]

피보나치 수열  $F(N)$ 의  $N$ 값을 입력 -->7  
 $F(7) = 13$   
 피보나치 수열  $F(N)$ 의  $N$ 값을 입력 -->20  
 $F(20) = 6765$   
 피보나치 수열  $F(N)$ 의  $N$ 값을 입력 -->30  
 $F(30) = 832040$

설명 : 피보나치 수열이란 0과 1을 제외하고 자신의 앞 숫자와 앞앞 숫자를 더하는 수열을 말한다. 피보나치 수열을 나열한 결과다.

0, 1, 0+1 = 1, 1+(0+1)=2, 1+2=3, 2+3=5, 3+5=8, 5+8=13 .... 식으로 진행된다. 이를 결과값만 표현하면 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987, .... 이 된다.

이를 공식으로 표현하면 다음과 같다.

```

F(0) → 0
F① → 1
F(N) → F(N-1) + F(N-2) (n이 2이상이면)

```

Solution09-10.py

```

## 함수 선언 부분
def fibo(n) :
    if n == 0 :
        return 0
    elif n == 1 :
        return 1
    else :
        return fibo(n-1) + fibo(n-2)

## 전역 변수 선언 부분
num = 0

## 메인 코드 부분
num = int(input('피보나치 수열 F(N)의 N값을 입력 -->'))

```

```
print('F(', num, ') = ', fibo(num))
```