

1. "파이썬"을 출력하는 print_python() 함수를 선언하고, print_python() 함수를 호출하여 실행해보자.

```
파이썬
```

2. "환영합니다."를 출력하는 welcome() 함수를 선언해보자. for 문을 이용하여 3회 반복하면서 welcome() 함수를 호출하여 실행해보자.

```
환영합니다.  
환영합니다.  
환영합니다.
```

3. 이름(예:홍길동)을 매개 변수로 전달 받으면 "환영합니다. 홍길동 님"으로 출력하는 welcome() 함수를 선언해보자. 이름을 입력 받아 welcome() 함수를 호출하여 실행해보자.

```
이름 : 홍길동  
환영합니다. 홍길동 님
```

```
def welcome(name) 형태로 함수를 선언하고, 문자열 이름을 입력 받아 welcome(n) 형태로 함수를 호출한다.
```

4. 문자열과 횟수를 매개 변수로 전달 받으면 횟수만큼 문자열을 반복하여 출력하는 print_str() 함수를 선언해보자. 문자열과 횟수를 각각 입력 받아 print_str() 함수를 호출하여 실행해보자.

```
문자열 : 파이썬  
횟수 : 3  
파이썬  
파이썬  
파이썬
```

5. 이름(예:홍길동)과 환영메시지(예:환영합니다.)를 매개 변수로 전달 받으면 "환영합니다. 홍길동 님"으로 출력하는 welcome() 함수를 선언해보자. 단, 환영메시지의 디폴트 매개변수 값은 "환영합니다."로 선언한다. 만약 welcome() 함수를 호출할 때 welcome(n, "반갑습니다.")와 같이 환영메시지 인수를 지정할 경우 "환영합니다." 대신에 "반갑습니다."를 출력한다.

```
이름 : 홍길동  
환영합니다. 홍길동 님  
반갑습니다. 홍길동 님
```

6. 원의 반지름을 입력 받아 원의 넓이를 구하여 반환하는 circle_area() 함수를 선언해보자. 반지름을 입력 받아 함수를 호출하고 결과를 반환 받아 출력해보자.

```
반지름 : 5  
반지름 5 의 넓이 : 78.5398
```

```
def circle_area(radius) 형태로 함수를 선언하며, 원의 넓이는 반지름*반지름*3.141592로 계산한다.
```

7. x^y의 거듭제곱을 구하는 pow_xy(x, y) 함수를 선언하고, 3 * 2⁴ + 5 수식의 결과를 pow_xy() 함수를 이용하여 계산해보자.

```
3 * 2**4 + 5 = 53
```

```
def pow_xy(x, y) 형태로 함수를 선언하며, 2의 4승은 pow_xy(2, 4) 형태로 호출한다.
```

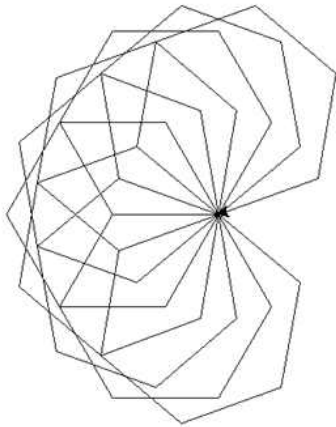
8. 정수를 전달 받아 양수이면 1, 0이면 0, 음수이면 -1을 반환하는 pzn() 함수를 선언해보자. 그리고 while 문을 이용하여 무한 반복하면서 사용자로부터 정수를 입력 받아 pzn() 함수를 호출하고, 함수의 결과를 전달 받아 결과가 1이면 "양수"를 출력하고, -1이면 "음수"를 출력하고, 0이면 "0"을 출력하고 무한 반복을 종료해보자.

```
정수 : 5  
양수  
정수 : -3  
음수  
정수 : 0  
0
```

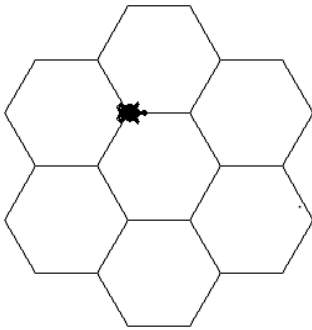
9. 함수의 매개변수로 2개 이상의 정수를 가변적으로 전달받아 합계를 구하여 반환하는 vsum() 함수를 선언해보자. vsum() 함수를 vsum(2, 3), vsum(2, 3, 4), vsum(2, 3, 4, 5) 순으로 인수의 수를 변경하면서 호출해보자.

```
2+3= 5  
2+3+4= 9  
2+3+4+5= 14
```

10. n -각형을 그리는 함수를 작성하고, 이 함수를 호출하여 다음과 같은 그림을 그려 보자.



11. 6각형을 그리는 `draw_hexa()` 함수를 작성하고 이 함수를 호출하여 다음과 같은 벌집모양을 화면에 그려보자



12. 터틀그래픽에서 거북이를 움직이지 않고 선을 긋는 함수 `draw_line()`을 정의하고 이것을 이용하여 다음과 같은 거미줄과 같은 모양을 그려보자. 거북이는 항상 중앙에 위치한다

