Exercise 8. While문

Section 8.1 - While문과 break

1. 10부터 시작해서 1씩 감소하며 출력하다가, 0이 되면 "발사!"를 출력하고 종료하는 카운트다운 프로그램을 작성하세요.

출력예시

```
10
9
8
7
6
5
4
3
2
1
0
발사!
```

> **힌트:** *countdown 변수를 10으로 시작해서 0이 되면 break하세요.*

2. 올바른 비밀번호는 "1234"입니다. 사용자가 비밀번호를 입력할 때마다 확인하고, 맞으면 "로그인 성공!"을 출력하고 종료하고, 틀리면 "다시 시도하세요"를 출력하고 계속 입력받는 프로그램을 작성하세요.

입력예시

```
wrong
1234
```

출력예시

```
다시 시도하세요
로그인 성공!
```

> **힌트:** *while True에서 비밀번호를 받고, "1234"와 비교해서 break하세요.*

3. 사용자가 단어를 계속 입력하면 리스트에 저장하고, "done"을 입력하면 지금까지 입력한 모든 단어를 출력하고 종료하는 프로그램을 작성하세요.

입력예시

```
apple
banana
orange
done
```

출력예시

```
단어 추가됨: apple
단어 추가됨: banana
단어 추가됨: orange
입력된 단어들: ['apple', 'banana', 'orange']
```

> **힌트:** *빈 리스트를 만들고 단어를 append하다가 "done"이면 break하세요.*

4. 정답이 7인 숫자 맞추기 게임을 만드세요. 사용자가 숫자를 입력하면 "틀렸습니다"를 출력하고, 7을 입력하면 "정답입니다!"를 출력하고 게임을 종료하세요.

입력예시

```
3
5
7
```

출력예시

```
틀렸습니다
틀렸습니다
정답입니다!
```

> **힌트:** *int(input())으로 숫자를 받고, 7과 비교해서 break하세요.*

5. 섭씨온도를 입력받아 화씨온도로 변환해서 출력하고, 999를 입력하면 "변환기를 종료합니다"를 출력하고 끝나는 프로그램을 작성하세요. (화씨 = 섭씨 * 9/5 + 32)

입력예시

```
0
100
999
```

출력예시

```
섭씨 0도 = 화씨 32.0도
섭씨 100도 = 화씨 212.0도
변환기를 종료합니다
```

> **힌트:** *while True에서 온도를 받고, 999가 아니면 변환 공식을 적용하세요.*

6. 사용자가 숫자를 계속 입력하면 누적 합계를 출력하고, 0을 입력하면 "최종 합계: X"를 출력하고 종료하는 프로그램을 작성하세요.

입력예시

```
5
3
2
0
```

출력예시

```
현재 합계: 5
현재 합계: 8
현재 합계: 10
최종 합계: 10
```

> **힌트:** *total 변수를 0으로 시작해서 누적하고, 입력이 0이면 break하세요.*

Section 8.2 - While문과 break, continue

7. 1부터 100 사이의 숫자만 입력받아 처리하고, 범위를 벗어나면 무시하며, 0을 입력하면 "유효한 숫자 X개를 입력했습니다"를 출력하고 종료하세요.

입력예시

```
50
150
-10
75
200
25
0
```

출력예시

```
유효한 숫자: 50
범위를 벗어났습니다
범위를 벗어났습니다
유효한 숫자: 75
범위를 벗어났습니다
유효한 숫자: 25
유효한 숫자 3개를 입력했습니다
```

> **힌트:** *1~100 범위를 벗어나면 continue로 건너뛰세요.*

8. 단어를 계속 입력받되, 3글자 미만인 단어는 무시하고 3글자 이상인 단어만 저장하며, "end"를 입력하면 저장된 단어들을 출력하고 종료하세요.

입력예시

```
hi
apple
go
banana
I
end
```

출력예시

```
너무 짧은 단어입니다
단어 저장: apple
너무 짧은 단어입니다
단어 저장: banana
너무 짧은 단어입니다
저장된 단어들: ['apple', 'banana']
```

> **힌트:** *단어 길이가 3 미만이면 continue로 건너뛰세요.*

9. 1부터 10까지 숫자를 확인하면서 홀수만 출력하는 프로그램을 작성하세요. (continue 사용)

출력예시

```
1
3
5
7
9
```

> **힌트:** *짝수일 때 continue를 사용해서 출력을 건너뛰세요.*

10. 시험 점수를 계속 입력받되, 0점은 무시하고 다른 점수만 출력하며, -1을 입력하면 종료하는 프로그램을 작성하세요.

입력예시

```
85
0
92
0
78
-1
```

출력예시

```
유효한 점수: 85
유효한 점수: 92
유효한 점수: 78
점수 입력 종료
```

> **힌트:** *0점이면 continue로 건너뛰고, -1이면 break하세요.*

11. 비밀번호를 입력받되, 4글자 미만이면 무시하고 다시 입력받으며, "1234"를 입력하면 로그인 성공하고 종료하는 프로그램을 작성하세요.

입력예시

```
12
abc
5678
1234
```

출력예시

```
비밀번호가 너무 짧습니다
비밀번호가 너무 짧습니다
비밀번호가 틀렸습니다
로그인 성공!
```

> **힌트:** *길이가 4 미만이면 continue로 건너뛰세요.*

12. 1부터 20까지 숫자를 출력하되, 5의 배수는 건너뛰는 프로그램을 작성하세요.

출력예시

```
1
2
3
4
6
7
8
9
11
12
13
14
16
17
18
19
```

> **힌트:** *5의 배수일 때 continue를 사용하세요.*

Section 8.3 - While문 안전장치 만들기

13. 다음 코드에 두 가지 안전장치를 추가하세요: 1) 100번 넘게 반복하면 종료 2) 합계가 1000 넘으면 종료

```
total = 0
while True:
    number = int(input("숫자 입력: "))
    total += number
    print(f"현재 합계: {total}")
```

입력예시

```
100
200
300
400
500
```

출력예시

```
숫자 입력: 100
현재 합계: 100
숫자 입력: 200
현재 합계: 300
숫자 입력: 300
현재 합계: 600
숫자 입력: 400
현재 합계: 1000
숫자 입력: 500
현재 합계: 1500
안전장치 작동! 합계가 1000을 초과했습니다.
```

> **힌트:** *반복 횟수와 합계를 모두 체크하는 안전장치를 추가하세요.*

14. 다음 게임 루프에 안전장치를 추가하세요. 10턴이 지나면 "게임 시간 초과"로 강제 종료되도록 만드세요.

```
score = 0
while True:
    action = input("행동을 선택하세요 (attack/run/quit): ")
    if action == "attack":
        score += 10
        print(f"공격! 점수: {score}")
    elif action == "run":
        print("도망쳤습니다!")
    elif action == "quit":
        print(f"게임 종료! 최종 점수: {score}")
        break
    else:
        print("잘못된 명령입니다.")
```

입력예시

```
attack
attack
run
attack
attack
attack
attack
run
attack
attack
attack
```

출력예시

```
행동을 선택하세요 (attack/run/quit): attack
공격! 점수: 10
행동을 선택하세요 (attack/run/quit): attack
공격! 점수: 20
행동을 선택하세요 (attack/run/quit): run
도망쳤습니다!
행동을 선택하세요 (attack/run/quit): attack
공격! 점수: 30
행동을 선택하세요 (attack/run/quit): attack
공격! 점수: 40
행동을 선택하세요 (attack/run/quit): attack
공격! 점수: 50
행동을 선택하세요 (attack/run/quit): attack
공격! 점수: 60
행동을 선택하세요 (attack/run/quit): run
도망쳤습니다!
행동을 선택하세요 (attack/run/quit): attack
공격! 점수: 70
행동을 선택하세요 (attack/run/quit): attack
공격! 점수: 80
게임 시간 초과! 최종 점수: 80
```

> **힌트:** *턴 수를 세는 변수를 추가하고 10턴이 되면 break하세요.*

15. 다음 코드에 안전장치를 추가하세요. 최대 10번 반복하면 강제로 종료되도록 만드세요.

```
while True:
    number = int(input("숫자 입력 (0이면 종료): "))
    if number == 0:
        break
    print(f"입력한 숫자: {number}")
```

입력예시

```
5
3
8
2
1
7
4
6
9
11
12
```

출력예시

```
숫자 입력 (0이면 종료): 5
입력한 숫자: 5
숫자 입력 (0이면 종료): 3
입력한 숫자: 3
숫자 입력 (0이면 종료): 8
입력한 숫자: 8
숫자 입력 (0이면 종료): 2
입력한 숫자: 2
숫자 입력 (0이면 종료): 1
입력한 숫자: 1
숫자 입력 (0이면 종료): 7
입력한 숫자: 7
숫자 입력 (0이면 종료): 4
입력한 숫자: 4
숫자 입력 (0이면 종료): 6
입력한 숫자: 6
숫자 입력 (0이면 종료): 9
입력한 숫자: 9
숫자 입력 (0이면 종료): 11
입력한 숫자: 11
안전장치 작동! 최대 입력 횟수에 도달했습니다.
```

> **힌트:** *반복 횟수를 세는 변수를 추가하고 10번 넘으면 break하세요.*

16. 다음 코드는 무한루프입니다. 5부터 1까지 카운트다운하도록 수정하세요.

```
countdown = 5
while countdown <= 5:  # 조건이 잘못됨!
    print(f"카운트다운: {countdown}")
    countdown -= 1
```

출력예시

```
카운트다운: 5
카운트다운: 4
카운트다운: 3
카운트다운: 2
카운트다운: 1
발사!
```

> **힌트:** *조건을 countdown > 0으로 바꾸고 발사 메시지를 추가하세요.*

17. 다음 비밀번호 체크 시스템의 무한루프를 수정하세요. 3번 틀리면 "계정 잠김", 맞으면 "로그인 성공"으로 종료하세요.

```
correct_password = "secret123"
attempts = 0

while True:
    password = input("비밀번호 입력: ")
    attempts += 1

    if password == correct_password:
        print("로그인 성공!")
        # break가 없음!
    else:
        print(f"틀렸습니다. 시도 횟수: {attempts}")
        # 3번 틀렸을 때 처리가 없음!
```

입력예시

```
wrong1
wrong2
wrong3
```

출력예시

```
비밀번호 입력: wrong1
틀렸습니다. 시도 횟수: 1
비밀번호 입력: wrong2
틀렸습니다. 시도 횟수: 2
비밀번호 입력: wrong3
틀렸습니다. 시도 횟수: 3
계정이 잠겼습니다.
```

> **힌트:** *성공 시 break를 추가하고, 3번 틀렸을 때도 break를 추가하세요.*

18. 다음 코드는 무한루프입니다. 리스트에서 3을 모두 제거하고 안전하게 종료하도록 수정하세요.

```
numbers = [1, 3, 2, 3, 4, 3, 5]
while 3 in numbers:
    print(f"3을 찾았습니다. 현재 리스트: {numbers}")
    # numbers.remove(3) 이 빠져있음!
```

출력예시

```
3을 찾았습니다. 현재 리스트: [1, 3, 2, 3, 4, 3, 5]
3을 제거했습니다. 현재 리스트: [1, 2, 3, 4, 3, 5]
3을 찾았습니다. 현재 리스트: [1, 2, 3, 4, 3, 5]
3을 제거했습니다. 현재 리스트: [1, 2, 4, 3, 5]
3을 찾았습니다. 현재 리스트: [1, 2, 4, 3, 5]
3을 제거했습니다. 현재 리스트: [1, 2, 4, 5]
모든 3이 제거되었습니다: [1, 2, 4, 5]
```

> **힌트:** *numbers.remove(3)을 추가하고 완료 메시지도 넣으세요.*

19. 다음 코드는 무한루프입니다. 1부터 5까지 출력하고 종료하도록 수정하세요.

```
count = 1
while count <= 5:
    print(f"숫자: {count}")
    # count += 1 이 빠져있음!
```

출력예시

```
숫자: 1
숫자: 2
숫자: 3
숫자: 4
숫자: 5
```

> **힌트:** *count를 증가시키는 코드를 추가하세요.*

20. 다음 코드는 무한루프입니다. 리스트에서 target을 찾으면 인덱스를 출력하고 종료하도록 수정하세요.

```
numbers = [10, 20, 30, 40, 50]
target = 30
index = 0
found = False

while not found:
    if numbers[index] == target:
        found = True
        print(f"{target}을 인덱스 {index}에서 찾았습니다!")
    # index += 1 이 빠져있음!
```

출력예시

```
30을 인덱스 2에서 찾았습니다!
```

> **힌트:** *index를 증가시키는 코드를 else 부분에 추가하세요.*