



AJOU UNIVERSITY



컴퓨터 프로그래밍 및 실습

5주차. 반복문

실습 안내

■ 실습 제출 안내

- 솔루션 이름은 "Practice week 5"
- 프로젝트 이름과 소스코드 이름은 Problem1, Problem2, ...
 - 실습1의 프로젝트 이름은 Problem1, 소스코드 이름은 problem1.c
 - 실습 2의 프로젝트 이름은 Problem2, 소스코드 이름은 problem2.c ...
- 솔루션 폴더를 압축하여 Practice_week5_학번_이름.zip 으로 제출
- 제출기한: 당일 19시 까지
- 실습 관련 코드: <https://github.com/lani009/Ajou-c-programming>

1. While loop

■ While loop의 구조

- Condition(조건식)이 참일 경우 거짓이 될 때 까지 아래 statement를 실행

Syntax: while 문

예

```
while( i < 10 )
```

조건식

```
printf("Hello World!\n");
```

조건식이 참이면 문장을 반복 실행한다.

```
#include <stdio.h>
```

```
int main(void) {  
    int index = 0;
```

```
    while (index++ < 10) {  
        // 1~10 까지 반복  
        printf("%d\n", index);  
    }
```

```
    index = 0;
```

```
    while (index < 10) {  
        // 0~9 까지 반복  
        printf("%d\n", index);  
        index++;  
    }
```

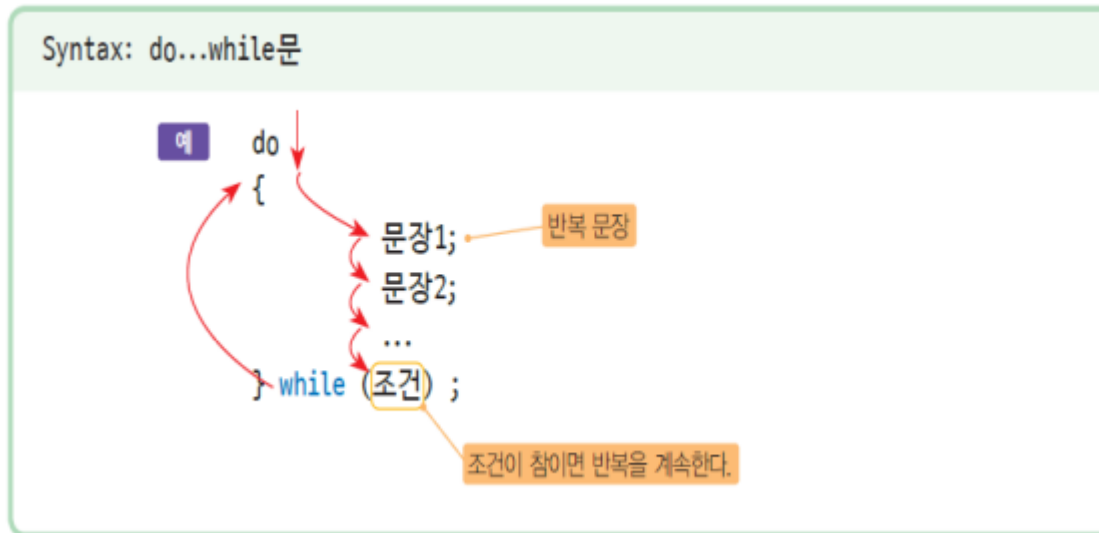
```
    return 0;
```

```
}
```

1. While loop

■ Do-While loop의 구조

- 루프에 처음 진입할 때, 조건 값 체크를 생략. 무조건 한번 이상은 반복된다.
- Condition(조건식)이 참일 경우 거짓이 될 때 까지 아래 statement를 실행



1. While loop

■ 무한 루프와 종료 조건

- 시스템 프로그래밍, 소켓, 커널 등의 환경에서 무한 루프를 쓸 일은 굉장히 많다.
- 루프의 조건이 true로 고정된다면 반복문이 영원히 실행된다.
- break 키워드로 반복문을 빠져나가는 것도 가능

```
#include <stdio.h>

int main(void) {
    while (1) {
        printf("Hello World!\n");
    }

    // 이 부분은 영원히 실행되지 않음
    return 0;
}
```

```
#include <stdio.h>

int main(void) {
    while (1)
    {
        // do something ...
        if (condition)
            break;
    }
    return 0;
}
```

1. While loop

■ 실습 1 - 약수 출력

- 사용자로부터 1 이상의 정수 n 을 받아온다. ($1 \leq n$)
- n 의 약수를 찾아서 출력한다.
- 예시
 - $n = 12$ 일 경우
 - 1 2 3 4 6 12 을 출력

1. While loop

■ 실습 2 - 업다운 게임

- 1이상 99이하의 범위를 가지는 랜덤한 수 하나를 생성한다.
- 사용자로부터 1이상 99 이하의 숫자를 입력받는다.
- 만약 입력 받은 수가 랜덤수 보다 크면 DOWN을 출력한다.
- 만약 입력 받은 수가 랜덤수 보다 작으면 UP을 출력한다.
- 만약 입력 받은 수가 랜덤수와 일치하면 CORRECT를 출력한다.
- 사용자가 입력한 수가 랜덤수와 일치할 때 까지 반복한다.

Microsoft Visual Studio 디버그 콘솔

```
입력>> 10
UP
입력>> 50
DOWN
입력>> 20
DOWN
입력>> 15
DOWN
입력>> 12
CORRECT

D:\git\WC-TAWC Programming\Practice - 실습\Practice Week 5\64\Deb
코드: 0개).
이 창을 닫으려면 아무 키나 누르세요...
```

2. For loop

■ For loop

- 순회할 변수의 initial condition을 정해줄 수 있다.

Syntax: for문

예 for(초기식; 조건식; 증감식) {
 printf("Hello World!");
}

반복되는 문장

2. For loop

- 실습 3 - 약수 출력
 - 실습 1번의 반복문을 For loop으로 고쳐보자

2. For loop

■ 중첩 반복문

```
#include <stdio.h>
```

```
int main(void) {  
    for (int i = 0; i < 5; i++)  
    {  
        printf("i=%d\n", i);  
        for (int j = 0; j < 5; j++)  
        {  
            printf("\tj=%d\n", j);  
        }  
    }  
  
    return 0;  
}
```

```
i=1  
    j=1  
    j=2  
    j=3  
    j=4  
    j=5  
  
i=2  
    j=1  
    j=2  
    j=3  
    j=4  
    j=5  
  
i=3  
    j=1  
    j=2  
    j=3  
    j=4  
    j=5
```

2. For loop

- 실습 4 - 구구단 출력
 - 2~9단 까지를 출력해보자

```
2 * 1 = 2
2 * 2 = 4
2 * 3 = 6
2 * 4 = 8
2 * 5 = 10
2 * 6 = 12
2 * 7 = 14
2 * 8 = 16
2 * 9 = 18

3 * 1 = 3
3 * 2 = 6
3 * 3 = 9
3 * 4 = 12
3 * 5 = 15
3 * 6 = 18
3 * 7 = 21
3 * 8 = 24
3 * 9 = 27

4 * 1 = 4
4 * 2 = 8
4 * 3 = 12
4 * 4 = 16
4 * 5 = 20
4 * 6 = 24
4 * 7 = 28
4 * 8 = 32
4 * 9 = 36
```

2. For loop

- 실습 5 - 순열 계산
 - 사용자로부터 n과 r값을 받아서 순열 값을 출력한다.

$${}_nP_r = n(n-1)(n-2) \cdots (n-r+1)$$

2. For loop

- 실습 6 - 피보나치 수열

- 사용자로부터 n값을 받아서 피보나치 수열의 값을 출력해보자

$$f(n) = \begin{cases} 0 & \text{if } n = 0 \\ 1 & \text{if } n = 1 \\ F(n-1) + F(n-2) & \text{if } n > 1 \end{cases}$$