K-Digital Training 스마트 팩토리 4기

반복문

반복문 ??

똑같은 명령을 일정 횟수만큼 반복해 수행하도록 하는 실행문

for

while

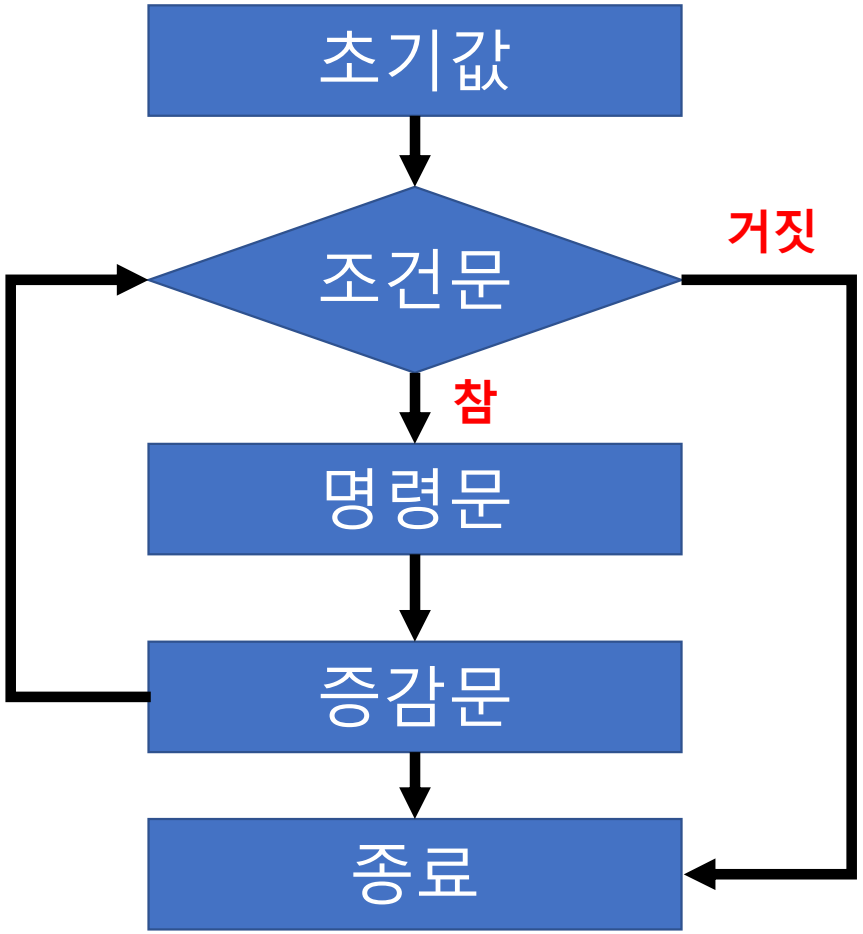
for-each

do / while

기본 For문

```
for ( 초기값; 조건문; 증감문 ){
        // 조건문의 결과가 참인 경우 실행
        // 실행할 코드
}
```

```cpp
for (i = 1; i <= 5; i++) {
    std::cout << "Iteration " << i << std::endl;
}
```



For-each문

배열의 원소에 대한 반복을 수행할 때 사용함.

(추후 배열에 대해 배운 후 알아볼 예정!)

While문

while (조건문) {

        // 조건문이 참인 경우 반복적으로 실행하는 명령문

}

```
i = 1;  // Reset i
while (i <= 5) {
    std::cout << "Iteration " << i << std::endl;
    i++;
}
```

※ 주의사항

- while문의 경우 명령문에서 조건문의 결과를 바꾸지 않으면 무한 루프에 빠질 수 있다.

do/while문

do {

	// 조건문이 참인 경우 반복적으로 실행하는 명령문

	// 단, 처음 한 번은 무조건 실행됨!

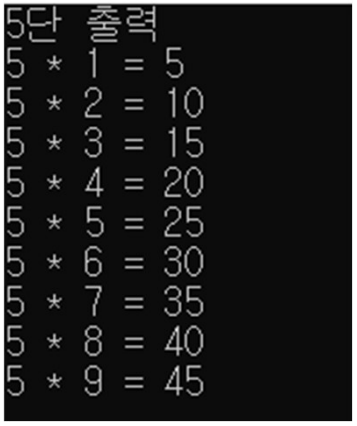
}

while (조건문);

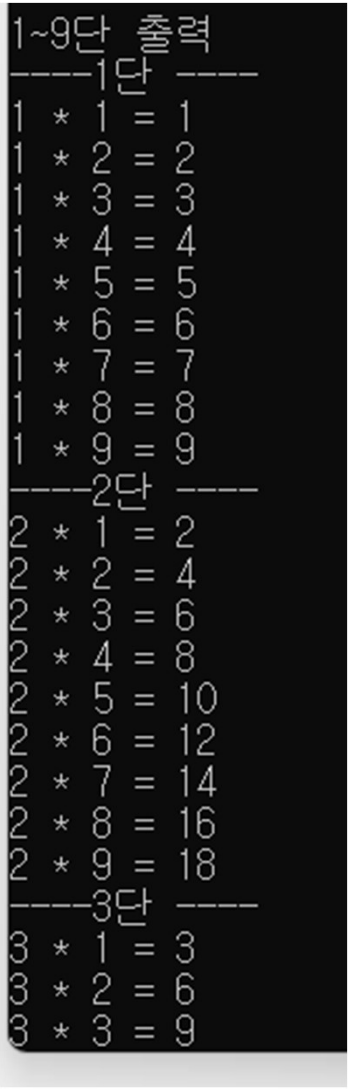
```
do {
    std::cout << "Iteration " << i << std::endl;
    i++;
} while (i <= 5);
```

실습1 구구단 만들기

- (1) 5단만 출력해보기
- (2) 1~9단까지 모두 출력해보기



(1)번 예시

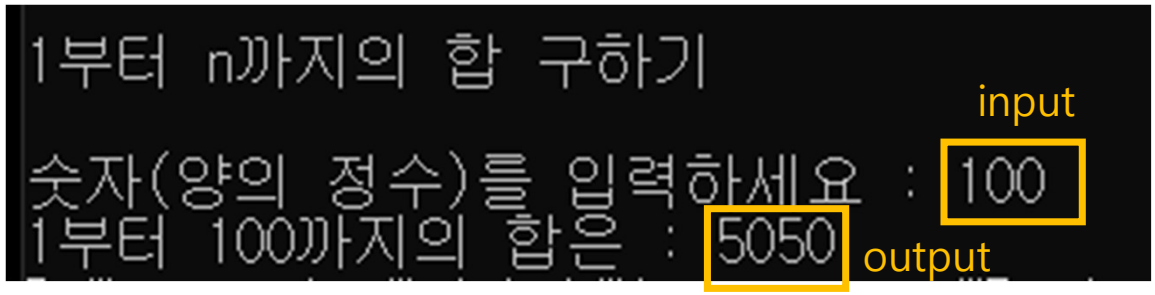


(2)번 예시

실습2 1부터 n까지의 합 구하기

- 1부터 사용자가 입력한 숫자까지의 합을 출력하는 코드 작성

input, output 예시



실습3 while문 사용해보기

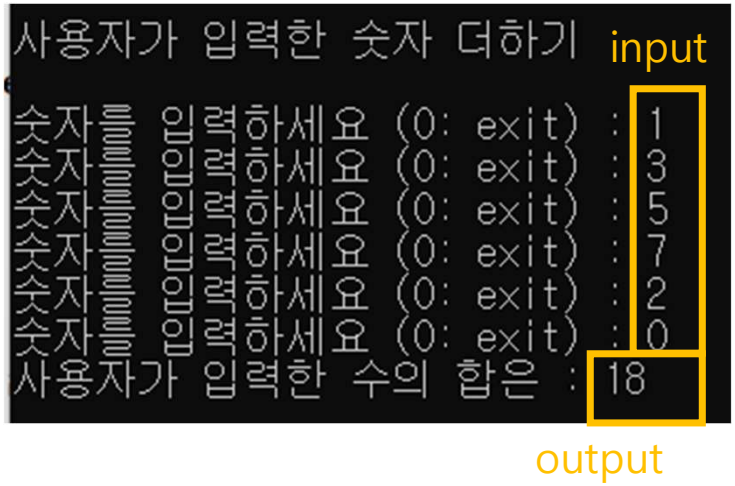
- (1) 앞에서 진행한 실습1, 2를 while문으로 작성해보기

- (2) for문과 while문을 비교하여 생각해보기
  - 보통 반복할 횟수가 정해져 있다면 for문을, 그렇지 않다면 while문을 사용함.

실습4 사용자가 입력한 숫자 더하기

- 사용자가 입력한 숫자들을 계속 더하는 프로그램 만들기
  - 사용자는 더하고 싶은 숫자를 횟수 제한 없이 입력할 수 있다.
  - 단, 0을 입력하면 종료

input, output 예시



[번외]실습. 입력한 숫자만큼 별 찍기

다 하신 분들이 계셔서 다 하신 분들은 이 실습도 한 번 해보세요~ 😃
아래 입출력 예시 참고하시면 됩니다!

input : 5
output :
*
**
***
****
*****

함수

함수 ??

어떤 일을 수행하는 코드의 묶음. 즉, 기능을 따로 빼서 묶는 것

함수를 사용하는 이유 ??

- 필요할 때마다 호출이 가능하다.
  - 반복적으로 수행해야 하는 기능을 한번 만들어 놓으면 필요할 때 마다 호출해서 사용할 수 있음.

- 논리적인 단위로 분할이 가능하다.
  - 코드를 기능에 따라 나눠서 볼 수 있음.
  - 코드를 분석할 때 함수로 구분이 되어있으면 분석이 쉬워짐.

함수 문법

```
리턴 타입 함수 이름( 인수 목록 )
{
    // 함수의 본문
}
```

함수 문법 – 리턴 타입

```
리턴 타입 함수 이름( 매개 변수 )
{
        // 함수의 본문
}
```

리턴 타입 : 이 함수가 결과로 어떤 유형의 값을 리턴(반환)할 지 선언

함수 문법 – 함수 이름

```
리턴 타입 함수 이름( 매개 변수 )
{
        // 함수의 본문
}
```

함수 이름 : 함수의 이름을 결정. 추후 호출 할 때 사용될 이름.

* 함수의 이름은 상대방이 이해하기 쉽도록 합리적으로 작성
  ex) 숫자들의 합을 구하는 함수를 작성 할 땐,
  aaa → X , sum → O

함수 문법 – 매개 변수

```
리턴 타입 함수 이름( 매개 변수 )
{
        // 함수의 본문
}
```

매개 변수 : 함수를 호출할 때 전달된 값을 함수 내부에서 사용할 수 있게 해주는 변수

함수 문법

리턴 타입 함수 이름(매개 변수) {

    // 함수의 본문

}

함수의 선언

```
int funcEx_1() {
    return 10;
}

std::string funcEx_2() {
    return "hello";
}

void funcEx_3() {
    std::cout << "리턴 타입이 void인 함수" << std::endl;
}
```

함수의 호출

```
int a = funcEx_1();
std::cout << a << std::endl;

std::string str = funcEx_2();
std::cout << str << std::endl;

funcEx_3();
```

함수 문법

```
리턴 타입 함수 이름( 매개 변수 ) {
        // 함수의 본문
}
```

함수의 선언 (매개변수 有)

```
int funcEx_sum(int n1, int n2) {
    return n1 + n2;
}
```

함수의 호출

```
std::cout << funcEx_sum(2,3) << std::endl;
```

실습1 사칙연산 함수 만들기

(1) 두 개의 정수를 매개 변수로 받아 그 합을 리턴하는 add함수를 만드세요.

(2) 두 개의 정수를 매개 변수로 받아 그 차를 리턴하는 sub함수를 만드세요.

단, 인수의 순서에 상관없이 큰 수에서 작은 수를 뺀 결과를 리턴

(3) 두 개의 정수를 매개 변수로 받아 그 곱을 리턴하는 mul함수를 만드세요.

(4) 두 개의 정수를 매개 변수로 받아 그 나눗셈을 리턴하는 divide함수를 만드세요.

단, divide함수는 실수를 리턴 (hint, 강제 형변환)

(5) main함수에서 위의 4개의 함수를 한번씩 호출하여 리턴된 값을 출력하세요.

[번외]실습

(1) 하나의 정수를 매개변수로 받아서

그 수가 짝수이면 "짝수", 홀수이면 "홀수" 라고 출력하는 함수 작성

(2) 세 개의 수를 매개변수로 받아서 그중 가장 큰 수를 반환하는 함수 작성.