

Lab 5

두 명씩 짝을 지어 문제마다 주어진 뼈대코드를 빈 부분은 채우며 코딩 하는데,
한 학생은 개발자 역할을 다른 학생은 코치 역할을 수행합니다.

10분마다 역할을 바꿉니다.

각 문제는 명기한 시간 안에 완성하고 조교의 점검을 받고 다음 문제로 넘어갑니다.

각 문제별로 정해진 시간을 초과할 수 없습니다.

문제 1~3

다음 코드를 읽고 어떤 값이 프린트 될지 예측해보자.

그리고 나서, 이를 직접 실행하여 확인하고 왜 그런지 실습 파트너와 토론하자.

문제 1 논리연산 이해 (10분)

```
char c = 'A';
int i = 5, j = 10;
printf("%d %d %d\n", !c, !!c, !!!c);
printf("%d %d %d\n", -!i, !-i, !-i-!j);
printf("%d %d %d\n", !(6*j+i-c), !i-5, !j-10);
```

0	1	0
0	0	0
1	-5	-10

문제 2 단축계산 이해 (10분)

```
int a = 0; b = 0; x;
x = 0 && (a = b = 777);
printf("%d %d %d\n", a, b, x);
x = 777 || (a = ++b);
printf("%d %d %d\n", a, b, x);
```

x = 0

x != 0

0	0	0
0	0	1

문제 3 중첩 if else 이해 (10분)

```
int i = 7, j = 7;
if (i == 1)
    if (j == 2)
        printf("%d\n", i = i + j);
else
    printf("%d\n", i = i - j);
printf("%d\n", i);
```

{ }

이 프로그램을 실행하면 Warning 메시지가 나온다. 이 메시지가 나오지 않도록 하는 두 가지 방법을 적용하고 확인해보자.

문제 4 (30분)

정수 값 n 을 받아서, n 이 양수이면 n 에서 $2 * n$ 까지의 정수의 합을 구하고, n 이 음수이면 $2 * n$ 에서 n 까지의 정수의 합을 계산하여 내주는 함수 작성하는데, 하나는 **for** 루프를 사용하고, 다른 하나는 **while** 루프를 사용하여 다음 함수 프로토타입에 맞추어 각각 작성하자.

```
int double_sum_for(int n);  
int double_sum_while(int n);
```

테스트호출 포함 뼈대코드: 4.c

문제 5 (60분)

정수 값 n 을 받아서, 첫 n 개 짝수의 합과 첫 n 개 홀수의 합을 각각 계산하는 함수 `odd_even_sum`을 다음 함수 프로토타입에 맞추어 작성하려고 한다.

```
void odd_even_sum(int n, int sums[]);
```

버전 1

첫 프로그램에서는 다음 코드를 사용한다.

```
for (cnt = 0, i = 1, j = 2; cnt < n; ++cnt, i += 2, j += 2)  
    odd += i, even += j;
```

이 코드는 콤마연산자를 많이 사용한다, 이렇게 사용하는 것이 좋은 코드는 아니지만, 콤마 연산자를 사용하는 방법을 알려주고 있다. 뼈대코드: 5.c에 맞추어 이 프로그램을 완성하고 테스트하여 맞는 결과를 출력하는지 확인하자.

버전 2

이번엔 콤마연산자 없이 하나 이상의 **for** 문을 사용하여 재작성하자.

버전 3

이번엔 **while** 문만 사용하여 재작성하자.