

강의명 : 프로그래밍

실습 번호 : 2

실습 제목 : Lab02-introduction(소개)

학생 이름 : 임지빈

학번 : 202211051

## 1. 두 정수의 합

### 1.1

변수 i,j,sum / int형

### 1.2

이 줄은 결과물을 출력하는것 같다.

### 1.3

이 프로그램은 더하기를 하려고 하는것같다.

### 1.4

```
i=10;  
j=20;  
sum=i+j;
```

```
i=100;
```

```
j=200;
```

```
sum=i+j;
```

```
i=1234;
```

```
j=5678;
```

```
sum=i+j;
```

### 1.5

```
gcc sum.c -o sum
```

### 1.6

```
s2211051@oak:~$ gcc sum.c -o sum
s2211051@oak:~$ ./sum
i=10, j=20, sum=30
i=100, j=200, sum=300
i=1234, j=5678, sum=6912
s2211051@oak:~$
```

## 2. 두 정수의 합 – return 값 없는 함수 사용

### 2.1

I, j = int

Return=없음

### 2.2

더하기를 하려고 하는것같다

### 2.3

```
sum=i+j;
printf("i=%d, j=%d, sum=%d\n", i, j, sum);
```

### 2.4

```
[s2211051@oak:lab02$ gcc sum-f1.c -o sum-f1
[s2211051@oak:lab02$ ./sum-f1
i=10, j=20, sum=30
i=100, j=200, sum=300
i=1234, j=5678, sum=6912
s2211051@oak:lab02$
```

## 3. 두 정수의 합 – return 값 있는 함수 사용

### 3.1

Int j, int I, return int형

### 3.2

더하기를 하는것같다

### 3.3

Return i+j;

### 3.4

```
[s2211051@oak:lab02$ gcc sum-f2.c -o sum-f2
[s2211051@oak:lab02$ ./sum-f2
i=10, j=20, sum=30
i=100, j=200, sum=300
i=1234, j=5678, sum=6912
s2211051@oak:lab02$
```

## 4. 윤년 계산

## 4.1

Int year, int leap

## 4.2

```
year=2019;
    leap=(year%4 == 0 && year % 100 != 0 )? 1:0;
year=2020;
    leap=(year%4 == 0 && year % 100 != 0 )? 1:0;
```

## 4.3

윤년은 4로 나뉘지고 100으로 나뉘지지 않는 수이다  
윤년이면 leap는 1, 아니면 0으로 설정함

## 4.4

```
[s2211051@oak:lab02$ gcc leapyear.c -o leapyear
[s2211051@oak:lab02$ ./leapyear
2019 is 0
2020 is 1
[s2211051@oak:lab02$ █
```

## 5. 윤년 계산 – return 값 없는 함수 사용

### 5.1

Int year, return의 형은 없다

### 5.2

```
leap=(year%4 == 0 && year % 100 !=0) ? 1:0;
printf("%d is %d\n", year, leap);
```

### 5.3

Leapyear에서 변수를 설정하고 main에서 인수만 받음

### 5.4

```
[s2211051@oak:lab02$ gcc leapyear-f1.c -o leapyear-f1
[s2211051@oak:lab02$ ./leapyear-f1
2019 is 0
2020 is 1
[s2211051@oak:lab02$ █
```

## 6. 윤년 계산 – return 값 있는 함수 사용

### 6.1

Int year, return의 형은 없다

### 6.2

```
(year % 4 == 0 && year % 100 != 0) ? 1:0;
printf("%d is a leap year\n", year);
```

### 6.3

### 6.4

```
s2211051@oak:lab02$ gcc leapyear-f2.c -o leapyear-f2
s2211051@oak:lab02$ ./leapyear-f2
2019 is a leap year
2019 is 20
2020 is a leap year
2020 is 20
s2211051@oak:lab02$
```

## 7. 실습 5와 실습 6의 비교 분석

### 7.1

실습 5는 0,1로 leapyear이 표현되지만 실습6은 leapyear이다 아니  
다로 표현된다

### 7.2

실습 6의 프로그램이 코딩하기엔 어렵지만 보기에는 편하고 실습 5  
는 코딩하기 쉽지만 보기에는 어렵다

### 7.3

실습 6이 보기 더 편하기에 선호한다.