

# Exercise 3

## 1. for loop

### 1-1. 설명

양의 정수 `N` (`int` 범위의 ) 을 입력받아서 `1` 부터 `N` 까지의 합을 출력하라 .

### 1-2. 예제 입력

```
10
```

### 1-3. 예제 출력

```
55
```

## 2. Book class

### 2-1. 설명

다음의 조건을 만족하는 `Book` class를 구현하고,

- 책의 이름을 멤버 변수로 가진다.
- 책의 가격을 멤버 변수로 가진다.
- 모든 멤버 변수는 `getter`와 `setter`를 가진다.

`N` 개의 책을 입력 받아서 이름들을 먼저, 가격들을 나중에 출력한다.

### 2-2. 입력 설명

- 첫줄에는 총 책의 개수를 의미하는 양의 정수 `N` (`int` 범위의 수) 가 입력된다.
- 둘째줄부터 `N` 줄 동안 책의 이름과 가격이 입력된다.
- 책 이름의 길이는 `100` 을 넘지 않는다.
- 책의 이름은 ' ' (빈칸)을 포함하지 않는다.
- 책 이름과 가격은 ' ' (빈칸)으로 구분된다.수

### 2-3. 출력 설명

- 첫줄부터 `N` 줄동안 책의 이름을 입력받은 순서대로 출력한다.
- 그 다음, `N + 1` 번째줄 부터 `N` 줄동안 책의 가격을 입력받은 순서대로 출력한다.

### 2-4. 예제 입력

```
2
Harry_Potter_and_the_Chamber_of_Secrets 13000
Harry_Potter_and_the_Prisoner_of_Azkaban 16000
```

### 2-5. 예제 출력

```
Harry_Potter_and_the_Chamber_of_Secrets
Harry_Potter_and_the_Prisoner_of_Azkaban
13000
16000
```

## 3. sizeof

### 3-1. 설명

다음의 data type들의 크기를 `sizeof` 통해 순서대로 출력한다.

- `long double`
- `double`
- `float`
- `unsigned long int`
- `long int`
- `unsigned int`
- `int`
- `unsigned short int`
- `short int`
- `unsigned char`
- `char`
- `bool`
- `enum season {SPRING, SUMMER, FALL, WINTER}`
- 

```
class Student {  
public:  
    int id;  
    int age;  
}
```

## 4. Vending class

### 4-1. 설명

돈을 입력으로 주면, 해당 돈을 최대한으로 써서 커피는 뽑아주는 기계를 class로 구현하려고한다.

다음 signature를 가지는 `Vending` class를 구현하고,

```
class Vending {  
private:  
    int price; // 한 컵당 가격  
    int count; // 기계가 보유한 커피의 개수  
public:  
    int coffee(int money);  
    void setPrice(int price); // `price`의 setter  
    void setCount(int count); // `count`의 setter  
}
```

- `int coffee(int money)` 는 입력으로 받은 돈 ( `int money` )을 최대한 써서 살 수 있는 커피의 개수를 반환하고, 그 수만큼 `count`를 줄인다.

입력받은 돈으로 살 수 있는 커피를 출력한다.

### 4-2. 입력 설명

- 첫줄에는 `Vending`의 초기값 `price` (int 범위)와 `count` (int 범위)가 ' '(공백)으로 구분지어 입력된다.
- 둘째줄에 돈 `N`이 입력된다.

### 4-3. 출력 설명

- 입력된 돈 `N`으로 살 수 있는 커피의 수를 출력한다.

### 4-4. 예제 입력

- 입력 1

```
200 10  
2200
```

- 입력 2

200 1  
150

4-5. 예제 출력

- 출력 1

10

- 출력 2

0