

# JAVA 프로그램 실습

반복문

경북대학교  
소프트웨어융합과  
교수 배희호

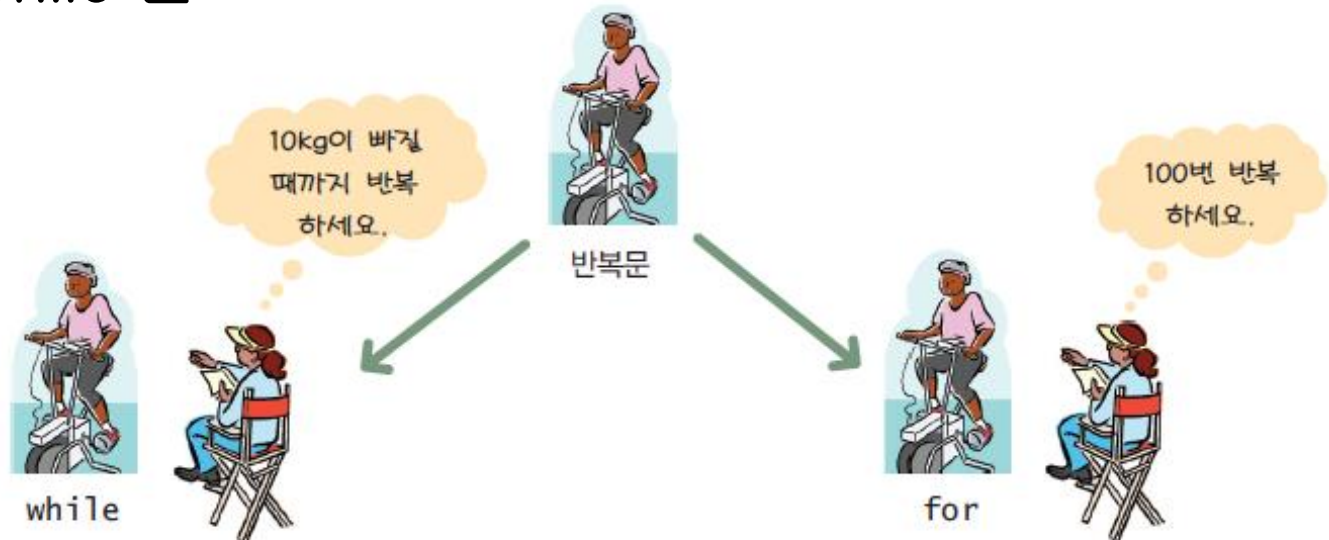
# 반복문

- 반복 구조는 왜 필요한가?
  - 같은 처리 과정을 되풀이하는 것이 필요하기 때문이다.  
학생 30명의 평균 성적을 구하려면 같은 과정을 30번 반복하여야 한다



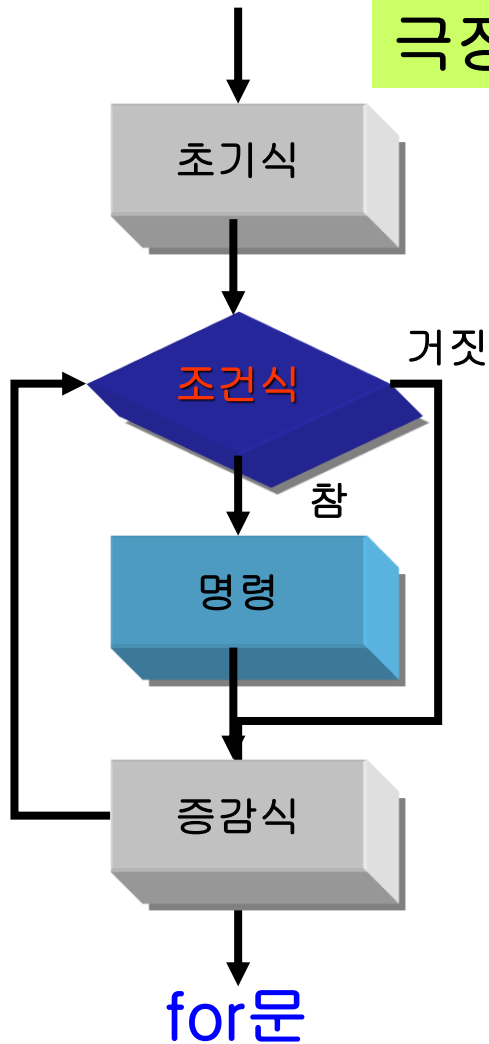
# 반복문

- 동일한 작업을 반복해서 처리하고자 할 때 사용하는 문장
- 반복문의 종류
  - for 문
  - while 문
  - do ~ while 문

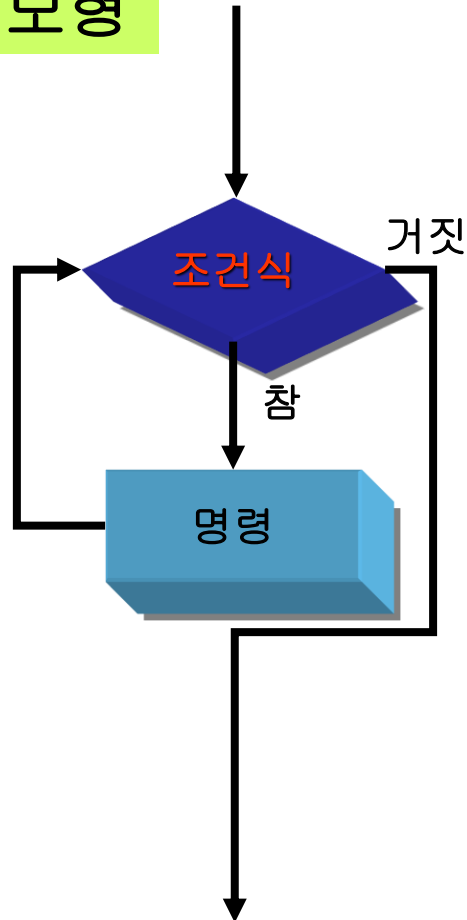


# 반복문

극장식 모형

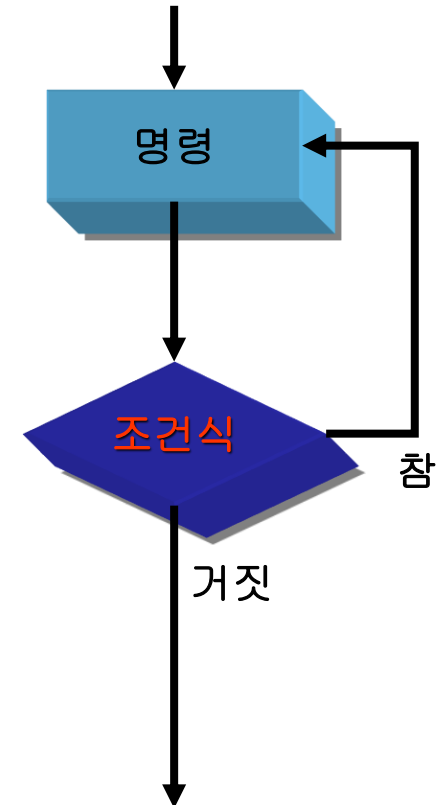


for문



while문

쇼핑센터 모형



do ~ while문

# 반복문

- 선 검사 반복문(극장, 스포츠 센터)
  - 각 반복을 수행하기 전에 루프 조건식(control expression)을 먼저 검사 한 후, 참(true)이면 Loop를 수행하고, 거짓(false)이면 Loop 종료
  - while 문, for 문
- 후 검사 반복문(쇼핑 센터)
  - 각 반복에서 Loop 문을 수행한 후, 루프 조건식(control expression)을 검사해서 참(true)이면 Loop를 계속 수행하고, 거짓(false)이면 Loop를 벗어남
  - do ~ while 문

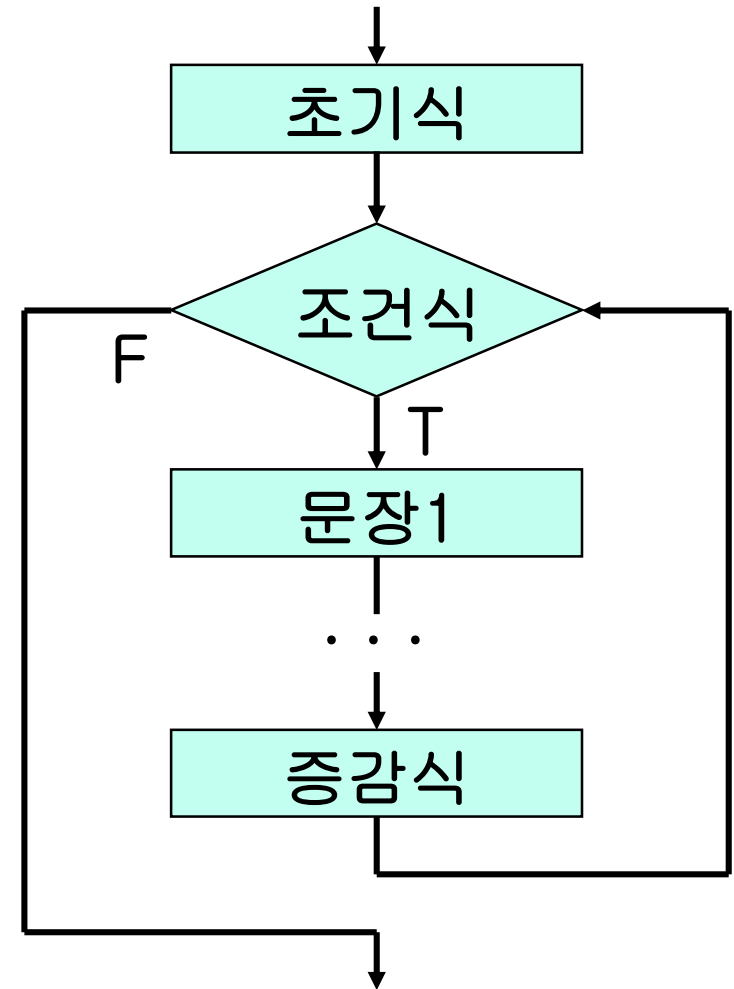
# 반복문

- 선(先) 검사 반복문(while, for 문)
  - Event 제어 루프
    - 특정 상황(Event)이 발생할 때 까지 반복 수행
      - 조건 수식을 이용해 Event 발생 여부 Check
    - 반복 횟수가 명확하지 않을 때 사용
    - while 문
  - Counter 제어 루프
    - Counter(제어) 변수 값을 증·감시켜 일정 횟수에 도달할 때까지 반복 수행
      - 조건 수식을 이용해 반복 횟수 Check
    - 반복 횟수가 명확할 때 사용
    - for 문

# 반복문

## ■ for 문

```
for (초기식; 조건식; 증감식) {  
    문장1;  
    ...  
    문장n;  
}
```



# 반복문

## ■ For 문

```
for (초기화부; 조건 검사부; 증감부)
{
    실행문장(들);
}
```

3 부분은  
각각  
생략 가능

- 실행 순서 : **1** **2** **4** **3** **2** **4** **3** **2** **4** **3**
- 초기화부 : 반복 실행을 제어하는 제어 변수 및 그 외 변수를 초기화하는 부분으로, 오직 한 번만 실행
- 조건 검사부 : for문 내의 문장을 실행하기 위하여 매번 실행하여 거짓(false)이면 루프를 빠져나감
- 증감부 : for문 내의 문장을 실행하고 나서 매번 실행



# 반복문

- for 문을 쓸 때 발생하는 오류
  - 세 수식을 분리할 때는 콤마( , )가 아니고 세미콜론( ; )

## 잘못된 사용법(콤마 사용)

```
for (index = 0, index <= 10, index++) {  
    .....  
}
```

## 올바른 사용법(세미콜론)

```
for (index = 0; index <= 10; index++) {  
    .....  
}
```

# 반복문

- for 문을 쓸 때 발생하는 오류
  - 제어 변수의 조건 수식을 잘못 사용하는 경우

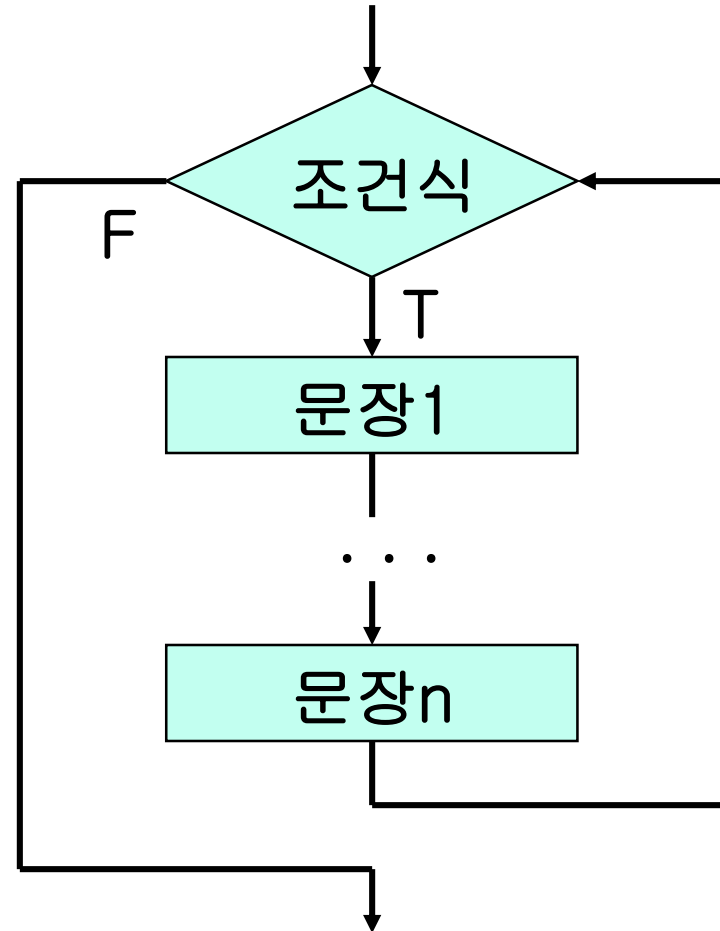
```
for (i = 1; i <= 10; i++) {  
    :
```

- 예는 반복문을 10번 반복할 목적으로 작성하였으나 loop를 한 번도 실행하지 않음
- 왜냐하면 조건 수식  $i == 10$ 은 처음부터 거짓(false)으로 판단되므로 loop를 실행하지 않고 끝냄

# 반복문

## ■ while 문

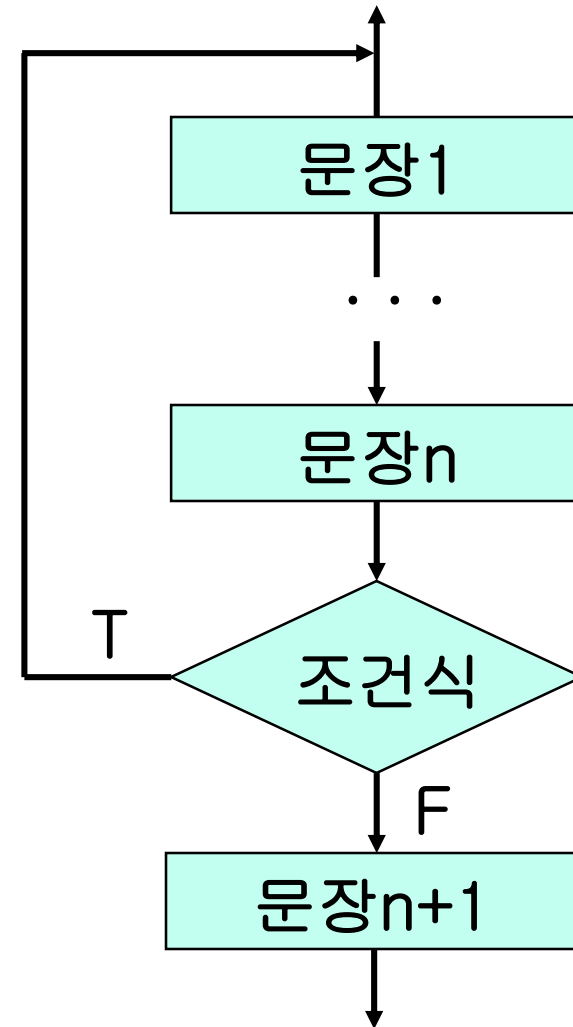
```
while (조건식) {  
    문장1;  
    ...  
    문장n;  
}
```



# 반복문

## ■ do-while 문

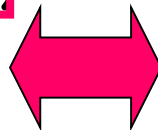
```
do {  
    문장1;  
    ...  
    문장n;  
} while (조건식);  
  
문장n+1;
```



# for문을 while문으로 변환

제어변수가 한 곳에 모여있어  
반복회수를 알 수 있는 곳에 사용

```
for (expr1 ; expr2 ; expr3)  
    Statement;  
next statement;
```



```
expr1;  
while (expr2)  
{  
    Statement;  
    expr3;  
}  
next statement;
```

# for문을 while문으로 변환 예제

```
for (int i = 10; i >= 0; i--)  
    System.out.println(i);
```

```
int i = 10;  
while (i >= 0)  
    System.out.println(i--);
```

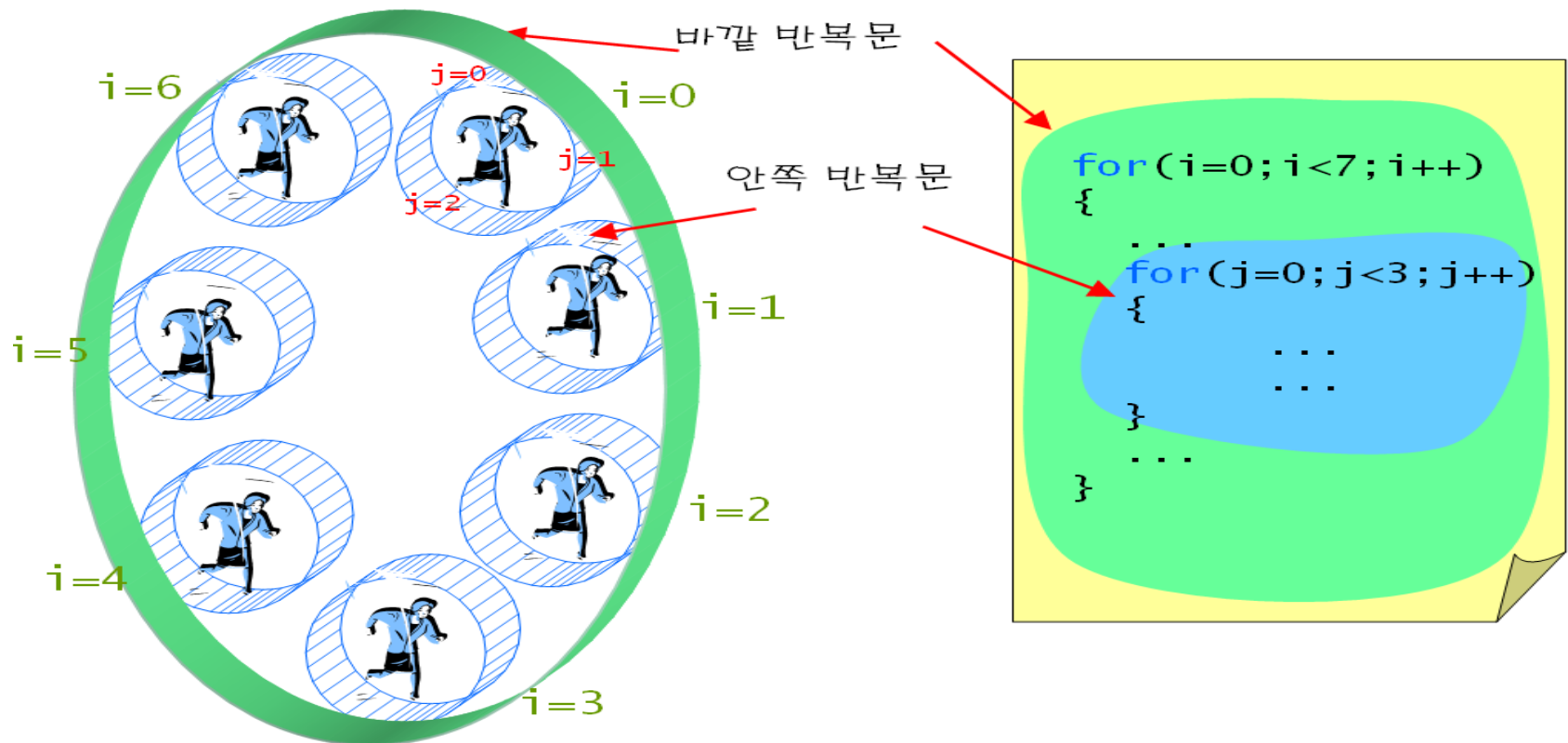
# for문을 while문으로 변환 예제

```
for (int i = 2; i <= 9; i++)  
    for (int j = 1; j <= 9; j++)  
        System.out.println(i + "*" + j + "=" + i * j);
```

```
int i = 2;  
while (i <= 9)  
    int j = 1;  
    while (j <= 9) {  
        System.out.println(i + "*" + j + "=" + i * j);  
        j++;  
    }  
    i++;
```

# Nested Loop

- 반복문 안에 다른 반복문이 위치
- 반복문을 중첩하여 사용할 때는 하나의 반복 구조 속에 다른 반복 구조가 완전히 포함되도록 하여야 함





# Nested Loop

- JAVA에서는 내부의 반복문과 외부의 반복문이 독립적으로 존재해야 한다는 것 이외에는 아무런 제한이 없음
- 교차되는 반복문의 사용은 허용되지 않음

```
{ for (cnt = 1; cnt < 100; cnt++){  
  { do {  
    /* for loop의 끝 */  
  } while (x != 0);  
}
```

안돼 !!

```
{ for (cnt = 1; cnt < 100; cnt++){  
  { do {  
    } while (x != 0);  
} /* for loop의 끝 */
```

음 !!!

# 무한 반복문

- 반복 조건이 항상 참인 반복문

```
for ( ; true )    // 수식2가 참(true)이다  
    문장;
```

```
while (true)      // 참(true)이다  
    문장;
```

- 무한 루프인 반복문은 내부에서 반복문을 종료할 수 있어야 함 (**break** 사용)

# 도입 예제

- 한 줄에 별표(\*) 다섯 개를 다섯 줄에 출력하는 Program 작성하여라

\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*

같은 작업을 반복적으로 수행할 필요가 있을 때  
반복문을 사용

# 도입 예제

```
public static void main(String[] args) {  
    System.out.println("*****");  
    System.out.println("*****");  
    System.out.println("*****");  
    System.out.println("*****");  
    System.out.println("*****");  
}
```

Program에서 반복적인 것은 작성하는 것은 ?

# 도입 예제

```
public static void main(String[] args) {  
    for (int i = 0; i < 5; i++)  
        System.out.println("*****");  
}
```

# 도입 예제

```
public static void main(String[] args) {  
    final int SIZE = 5;  
  
    for (int i = 0; i < SIZE; i++) {  
        System.out.println();  
        for (int j = 0; j < SIZE; j++)  
            System.out.print("*");  
        }  
    }
```

# 도입 예제 [심화]

- 다음과 같이 출력되는 Program을 작성하여라

\*\*\*\*\*

\*\*\*\*

\*\*\*

\*\*

\*

\*\*

\*\*\*

\*\*\*\*

\*\*\*\*\*

# 도입 예제 [심화]

```
public static void main(String[] args) {  
    final int SIZE = 5;  
  
    for (int i = SIZE; i > 0; i--) {  
        for (int j = 0; j < i; j++) {  
            System.out.print("*");  
        }  
        System.out.println();  
    }  
  
    for (int i = 0; i < SIZE; i++) {  
        for (int j = 0; j <= i + 1; j++) {  
            System.out.print("*");  
        }  
        System.out.println();  
    }  
}
```



# 도입 예제 [심화]

- 다음과 같이 출력되는 Program을 작성하여라

```
      *
     **
    ***
   ****
  *****
 *****
  *****
   ****
    ***
     **
      *
```

# 도입 예제 [심화]

```
public static void main(String[] args) {  
    for (int i = 5; i >= 1; i--) {  
        for (int j = 1; j < i; j++) {  
            System.out.print(" ");  
        }  
        for (int j = 0; j <= 5 - i; j++) {  
            System.out.print("*");  
        }  
        System.out.println();  
    }  
  
    for (int i = 1; i < 5; i++) {  
        for (int j = 0; j < i; j++) {  
            System.out.print(" ");  
        }  
        for (int j = 1; j <= 5 - i; j++) {  
            System.out.print("*");  
        }  
        System.out.println();  
    }  
}
```

# Mile을 Meter로 변환

- 1에서 10 마일(mile)을 까지 입력 받아 미터(meter)법으로 환산하는 Program을 작성하여라
- 단, 1 mile = 1609.344 m



# Mile을 Meter로 변환

- 문제 분석

- Data는 무엇이고, Information은 무엇인가 ?

- 입력

- 마일(mile) – int

- 1에서 10 Mile까지 연속으로 입력 받으려면 ?

- 출력

- 미터(meter) – int

- 계산 방법

- 1 mile = 1609.344 meter

# Mile을 Meter로 변환

- Program에서 반복되는 부분을 반복문(for, while, do~while)을 이용하여 수정하여라

```
public static void main(String[] args) {  
    final float MILE = 1609.344f;  
    int meter;
```

```
        meter = (int) (1 * MILE);
```

```
        System.out.printf("1 마일은 %,d 미터입니다\n", meter);
```

```
        meter = (int) (2 * MILE);
```

```
        System.out.printf("2 마일은 %,d 미터입니다\n", meter);
```

```
        meter = (int) (3 * MILE);
```

```
        System.out.printf("3 마일은 %,d 미터입니다\n", meter);
```

```
    }
```

# Mile을 Meter로 변환(for)

## ■ for 문을 사용

```
public static void main(String[] args) {  
    final float MILE = 1609.344f;  
    final int LAST = 10;  
    int meter;  
  
    for (int i = 1; i <= LAST; i++) {  
        meter = (int) (i * MILE);  
        System.out.printf("%2d 마일은 %,6d 미터입니다\n", i, meter);  
    }  
}
```

# 마일을 미터로 변환(while)

## ■ while 문을 사용

```
public static void main(String[] args) {  
    final float MILE = 1609.344f;  
    final int LAST = 10;  
    int meter;  
  
    int i = 1;  
    while (i <= LAST) {  
        meter = (int) (i * MILE);  
        System.out.printf("%2d 마일은 %,6d 미터입니다\n", i, meter);  
        i++;  
    }  
}
```

# Mile을 Meter로 변환(do~while)

## ■ do ~ while 문을 사용

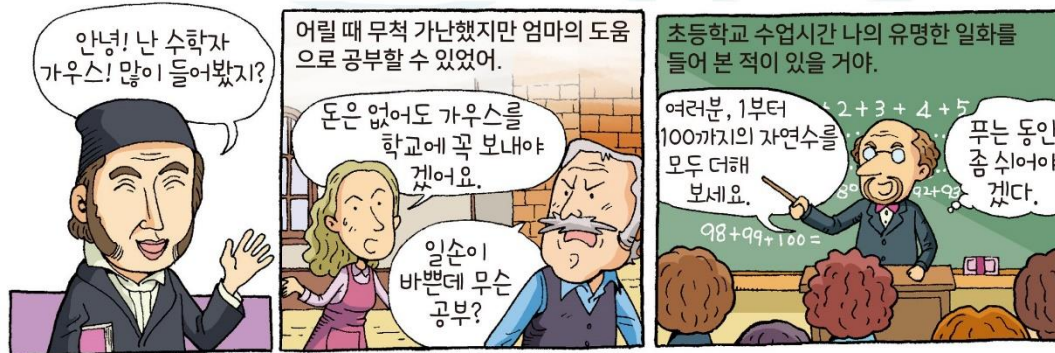
```
public static void main(String[] args) {  
    final float MILE = 1609.344f;  
    final int LAST = 10;  
    int meter;  
  
    int i = 1;  
    do {  
        meter = (int) (i * MILE);  
        System.out.printf("%2d 마일은 %,6d 미터입니다\n", i, meter);  
        i++;  
    } while (i <= LAST);  
}
```



# 1부터 100까지의 합

- for문을 이용하여 1부터 100까지 덧셈을 표시하고 합을 구하시오.

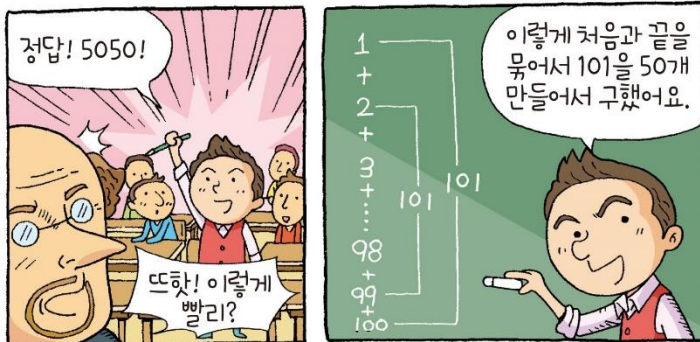
$$1 + 2 + \dots + 100 = 5,050$$



난 새로운 방식을 만들어 문제를 푸는 것을 좋아했어.

나만의 공식을 만들어 계산하면 사람들은 천재라며 무척 놀라곤 했지.

이런 천재성은 수학의 여러 분야에서 업적을 남겼어. 좀 어렵지? 앞으로 배울 수학에서 나올 거야.



# 1부터 100까지의 합

## ■ <방법 1>

■ 1 부터 100까지의 숫자를 열로 정렬한 후 더한다

$$\begin{array}{r} 1 \\ 2 \\ 3 \\ \vdots \\ \vdots \\ 98 \\ 99 \\ \hline + 100 \\ 5050 \end{array}$$

코딩 방법 생각

# 1부터 100까지의 합

## ■ <방법 2>

- 두 수의 합이 101이 되도록 숫자들을 그룹으로 정렬 한다. 그룹의 개수에 100을 곱한 후에 사용되지 않는 수를 앞에서 구한 총합에 더한다

$$\begin{array}{rcl} 1 + 100 & = & 101 \\ 2 + 99 & = & 101 \\ 3 + 98 & = & 101 \\ & \vdots & \\ & \vdots & \\ 50 + 51 & = & 101 \end{array} \quad \begin{array}{l} \text{50 그룹} \\ \downarrow \\ (50 \times 101) = 5050 \end{array}$$

가우스 생각

# 1부터 100까지의 합

## ■ <방법 3>

### ■ 공식을 사용하는 방법

$$\text{sum} = \frac{\text{item} \times (\text{first} + \text{last})}{2}$$

item = 더하는 항의 수 (100 개)

first = 더하는 처음 숫자 (1)

last = 더하는 수의 마지막 숫자 (100)

$$\text{sum} = \frac{100 \times (1 + 100)}{2} = 5050$$

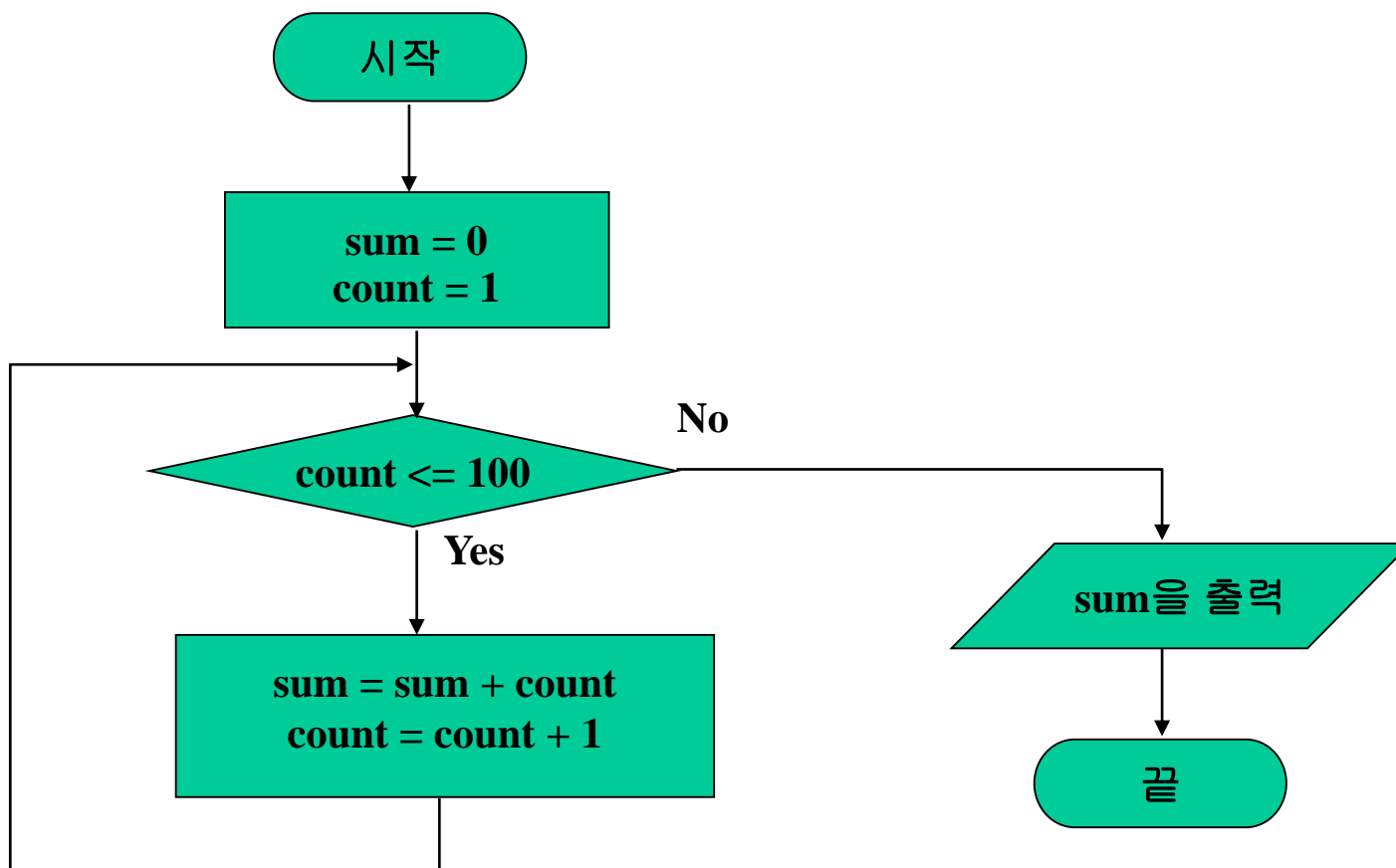
# 1부터 100까지의 합

■ <방법 1>을 가상 언어로 표현 (제 1, 2 세분화)

```
sum = 0
count = 1
while (count가 100보다 작은 값을 가지면)
    sum에 count 만큼 더한다.
    count에 1을 더한다.
end while
sum을 출력한다.
```

# 1부터 100까지의 합

## ■ 순서도 표현



# 1부터 100까지의 합(for)

```
public static void main(String[] args) {  
    final int LAST = 100;  
    int sum = 0;
```

while 문으로 작성해 보자

```
    for (int count = 1; count <= LAST; count++) {  
        sum += count;  
    }
```

```
    System.out.printf("1부터 %d까지의 합 : %,d\n", LAST, sum);  
}
```

# 1부터 100까지의 합(while)

```
public static void main(String[] args) {  
    final int LAST = 100;  
    int sum = 0;
```

```
    int count = 1;  
    while (count <= last) {  
        sum += count;  
        count++;  
    }
```

do ~ while 문으로 작성해 보자

```
    System.out.printf("1부터 %d까지의 합 : %,d\n", LAST, sum);
```

```
}
```



# 1부터 100까지의 합(do~while)

```
public static void main(String[] args) {  
    final int LAST = 100;  
    int sum = 0;  
  
    int count = 1;  
    do {  
        sum += count;  
        count++;  
    } while (count <= LAST);  
    System.out.printf("1부터 %d까지의 합 : %d\n", LAST, sum);  
}
```

# 1부터 100까지의 합(방법 2)

```
public static void main(String[] args) {  
    final int LAST = 101;  
    int sum = 0;  
  
    if (LAST % 2 == 0) {  
        int temp = 1 + LAST;  
        sum = temp * (LAST / 2);  
    } else {  
        int temp = 1 + LAST;  
        sum = temp * (LAST / 2);  
        sum += (LAST / 2 + 1);  
    }  
  
    System.out.printf("1부터 %d까지의 합 : %,d\n", LAST, sum);  
}
```

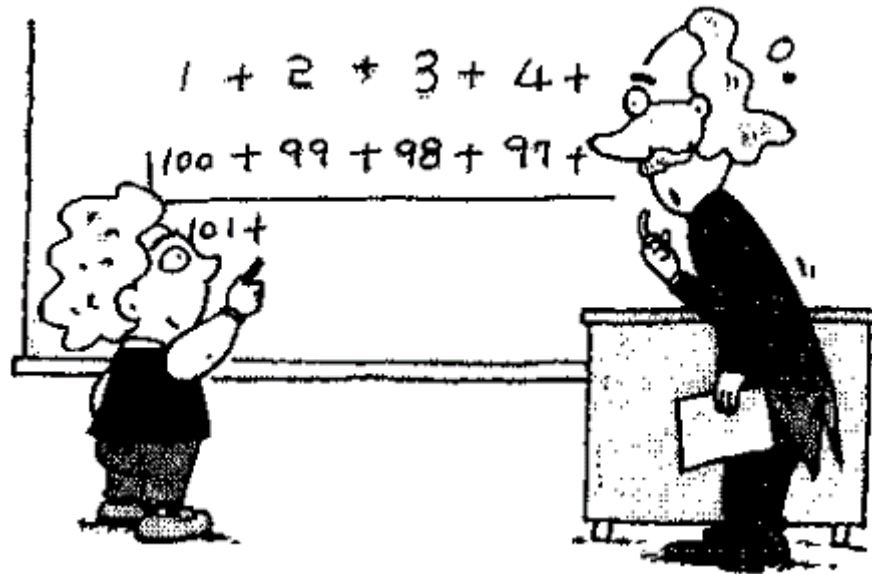
홀수일 때를 생각해야 함

# 1 부터 100까지의 합(방법 3)

```
public static void main(String[] args) {  
    final int LAST = 101;  
    int sum;  
  
    sum = LAST * (LAST + 1) / 2;  
  
    System.out.printf("1부터 %d까지의 합 : %,d\n", LAST, sum);  
}
```

# 1에서 N까지의 합[심화]

- 1에서 100까지의 합을 구하는 프로그램을 일반화하여 1부터 N까지의 합을 구하는 프로그램을 만들어보아라
  - try ~ catch문 사용



# 1에서 N까지의 합[심화]

```
public static void main(String[] args) throws IOException {  
    Scanner keyboard = new Scanner(System.in);  
    int last;  
    long sum = 0L;  
  
    do {  
        System.out.print("Wn 어디까지 더할까요 ? ");  
        last = keyboard.nextInt();  
        if (last >= 1)  
            break;  
        else {  
            System.err.print("Wn ERROR !!");  
            System.in.read();  
        }  
    } while (true);  
}
```

# 1에서 N까지의 합[심화]

```
for (int i = 1; i <= last; i++) {  
    sum += i;  
}
```

```
System.out.printf(" %d + %d + .... + %,d = %,d\n", 1, 2, last,  
sum);  
}
```

# X에서 Y까지의 합[심화]

```
public static void main(String[] args) throws IOException {  
    Scanner keyboard = new Scanner(System.in);  
    int start, last;  
    long sum = 0L;  
  
    do {  
        System.out.print("\n 어디부터 더할까요 ? ");  
        start = keyboard.nextInt();  
        if (start >= 1)  
            break;  
        else {  
            System.err.print(" ERROR !!");  
            System.in.read();  
        }  
    } while (true);  
}
```



Futuristic Innovator

京福大學校  
KYUNGBOK UNIVERSITY

# X에서 Y까지의 합[심화]

```
do {  
    System.out.print(" 어디까지 더할까요 ? ");  
    last = keyboard.nextInt();  
    if (last >= 1 && last > start)  
        break;  
    else {  
        System.err.print(" ERROR !!");  
        System.in.read();  
    }  
} while (true);  
  
for (int i = start; i <= last; i++) {  
    sum += i;  
}  
  
System.out.printf(" %d + %d + .... + %,d = %,d\n", start,  
    start + 1, last, sum);  
}
```

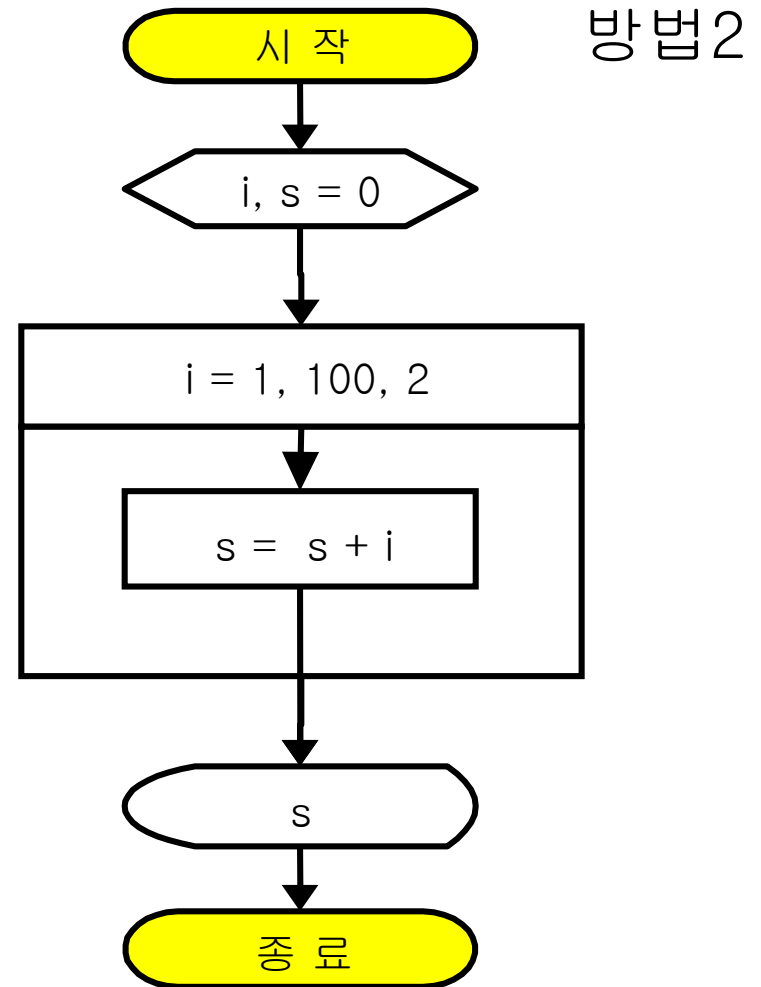
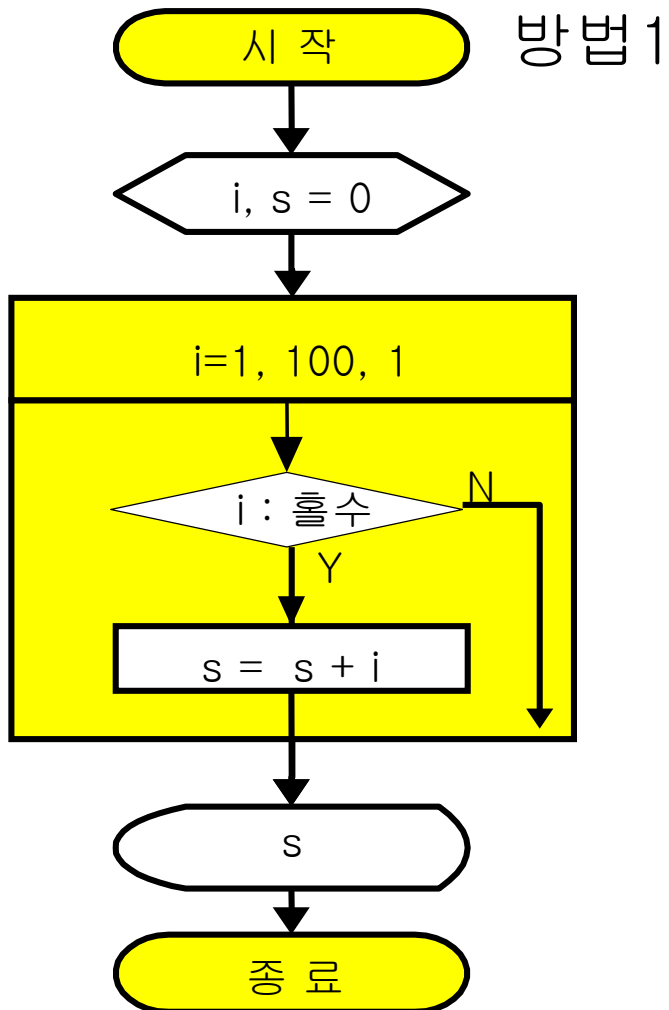


# 1부터 100까지 홀수 합

- 1부터 100까지의 홀수의 합을 구하여라



# 1부터 100까지 홀수 합



# 1부터 100까지 홀수 합

```
public static void main(String[] args) {  
    final int LAST = 100;  
    long sum = 0L;  
    int loop = 1;  
  
    while (loop <= LAST) {  
        if (loop % 2 == 1) /*홀수 */  
            sum += loop;  
        loop++;  
    }  
  
    System.out.printf("1부터 %d까지의 홀수의 합 = %,d", LAST, sum);  
}
```

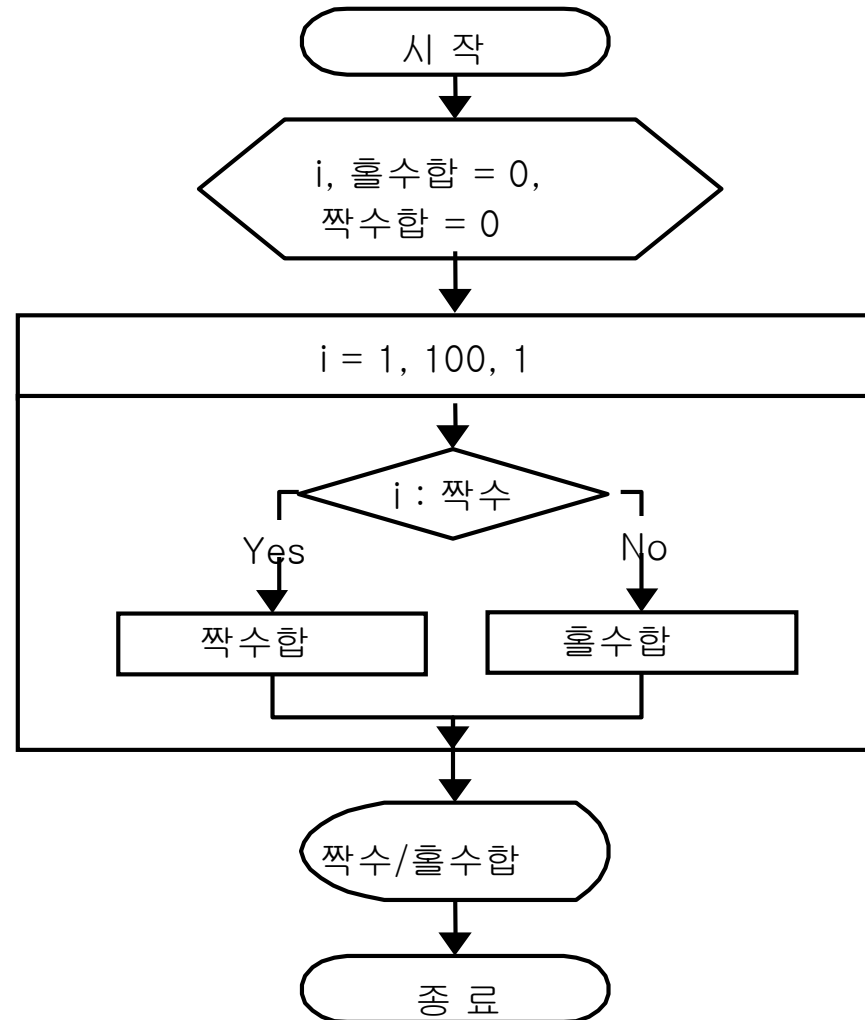
# 1부터 100까지 홀수 합

```
public static void main(String[] args) {  
    final int LAST = 100;  
    long sum = 0L;  
  
    for (int loop = 1; loop <= LAST; loop += 2) {  
        sum += loop;  
    }  
  
    System.out.printf("1부터 %d까지의 홀수의 합 = %,d", LAST, sum);  
}
```

# 홀수, 짝수 합 구하기

- 1부터 100까지의 정수 중에 이들 중 짝수와 홀수의 합을 구하는 순서도 및 프로그램
- 문제 해결 방안
  - 순서적으로 점검 : 짝수/홀수
  - 짝수/홀수 : 어떻게 검사하는가?
    - 2로 나누어서 나머지 0 - 짝수
    - 2로 나누어서 나머지 1 - 홀수

# 홀수, 짝수합 구하기



# 홀수, 짝수 합 구하기

```
public static void main(String[] args) {  
    final int LAST = 100;  
    long even = 0, odd = 0L;  
  
    for (int loop = 1; loop <= LAST; loop++) {  
        if (loop % 2 == 1) /*홀수 */  
            odd += loop;  
        else  
            even += loop;  
    }  
  
    System.out.printf("1부터 %d까지의 홀수의 합 = %,d\n", LAST, odd);  
    System.out.printf("1부터 %d까지의 짝수의 합 = %,d\n", LAST, even);  
}
```



Futuristic Innovator

京福大學校  
KYUNGBOK UNIVERSITY

# 입력 점수의 총점/평균 구하기

- while 문을 이용하여 Keyboard에서 JAVA 점수를 입력 받아 입력 받은 모든 점수의 총점과 평균을 출력하는 프로그램을 작성해보자.

몇명의 성적을 입력 받을 것입니까? 5

67

89

89

76

78

입력된 JAVA 점수는 5개이며 총점은 399이고, 평균은 79.80 입니다



# 입력 점수의 총점/평균 구하기

```
public static void main(String[] args) throws IOException {  
    Scanner keyboard = new Scanner(System.in);  
    int count = 0;  
    int sum = 0;  
    int value = 0;  
  
    do {  
        System.out.print("몇명의 성적을 입력 받을 것입니까? ");  
        count = keyboard.nextInt();  
        if (count >= 1)  
            break;  
        else {  
            System.err.print("입력 ERROR !!");  
            System.in.read();  
        }  
    } while (true);  
}
```

# 입력 점수의 총점/평균 구하기

```
int i = 1;
while (i <= count) {
    value = keyboard.nextInt();
    if (value >= 0 && value <= 100) {
        sum += value;
        i++;
    } else {
        System.err.print("입력 ERROR !!");
        System.in.read();
    }
}

System.out.print("입력된 JAVA 점수는 " + count);
System.out.printf("개이며 총점은 %d이고, 평균은 %.2f 입니다\n",
                  sum, (float) sum / count);
}
```



Futuristic Innovator

京福大學校  
KYUNGBOK UNIVERSITY

# 입력 점수의 총점/평균 구하기[심화]

- 음수를 입력하면 점수 입력 받는 것을 종료하도록 수정해보자

성적을 입력해주세요(음수 입력 종료)

56

78

87

86

57

-5

입력된 JAVA 점수는 5개이며 총점은 364이고, 평균은 72.80 입니다

# 입력 점수의 총점/평균 구하기[심화]

```
public static void main(String[] args) throws IOException {
    Scanner keyboard = new Scanner(System.in);
    int sum = 0;
    int value = 0;

    System.out.println("성적을 입력해주세요(음수 입력 종료) ");
    int count = 0;
    while (true) {
        value = keyboard.nextInt();
        if (value < 0)
            break;
        else if (value <= 100) {
            sum += value;
            count++;
        } else {
            System.err.print("입력 ERROR !!");
            System.in.read();
        }
    }
}
```

# 입력 점수의 총점/평균 구하기[심화]

```
System.out.print("입력된 JAVA 점수는 " + count);  
System.out.printf("개이며 총점은 %d이고, 평균은 %.2f 입니다\n",  
sum, (float) sum / count);  
}
```

# 입력 점수의 총점/평균 구하기[심화]

```
public static void main(String[] args) throws IOException {
    Scanner keyboard = new Scanner(System.in);
    int sum = 0;
    int value = 0;

    System.out.println("성적을 입력해주세요(음수 입력 종료) ");
    int count = 0;
    while ((value = keyboard.nextInt()) >= 0) {
        if (value <= 100) {
            sum += value;
            count++;
        } else {
            System.err.print("입력 ERROR !!");
            System.in.read();
        }
    }
    System.out.print("입력된 JAVA 점수는 " + count);
    System.out.printf("개이며 총점은 %d이고, 평균은 %.2f 입니다\n",
        sum, (float) sum / count);
}
```

# 달팽이 숫자 출력

- Keyboard에서 10보다 작은 자연수  $n$ 을 입력 받아 다음과 같이(예  $n = 5$ ) 출력하는 Program을 작성하여라.

1	2	3	4	5
10	9	8	7	6
11	12	13	14	15
20	19	18	17	16
21	22	23	24	25

# 달팽이 숫자 출력

```
public static void main(String[] args) throws IOException {  
    Scanner keyboard = new Scanner(System.in);  
    int num;  
  
    do {  
        System.out.print(" 정수 입력 : ");  
        num = keyboard.nextInt();  
        if (num <= 10)  
            break;  
        else {  
            System.err.print("입력 ERROR !!");  
            System.in.read();  
        }  
    } while(true);  
}
```



# 달팽이 숫자 출력

```
for (int i = 0; i < num; i++) {  
    System.out.println();  
    if (i % 2 == 0)  
        for (int j = i * num + 1; j <= i * num + num; j++)  
            System.out.printf("%4d", j);  
    else  
        for (int j = i * num + num; j > i * num; j--)  
            System.out.printf("%4d", j);  
}
```

# Multiple number 문제

- 1부터 100까지의 정수 중에서 3의 배수가 아닌 수의 총합을 구하시오.

1부터 100까지의 합 = 5,050

1부터 100까지 3의 배수의 합 = 1,683

1부터 100까지 3의 배수를 제외한 수의 합 = 3,367

# Multiple number 문제

## ■ 문제 분석

■ 반복 구간 : 1에서 100

■ 더해지는 숫자에 대한 정의

■ 3의 배수만 더함

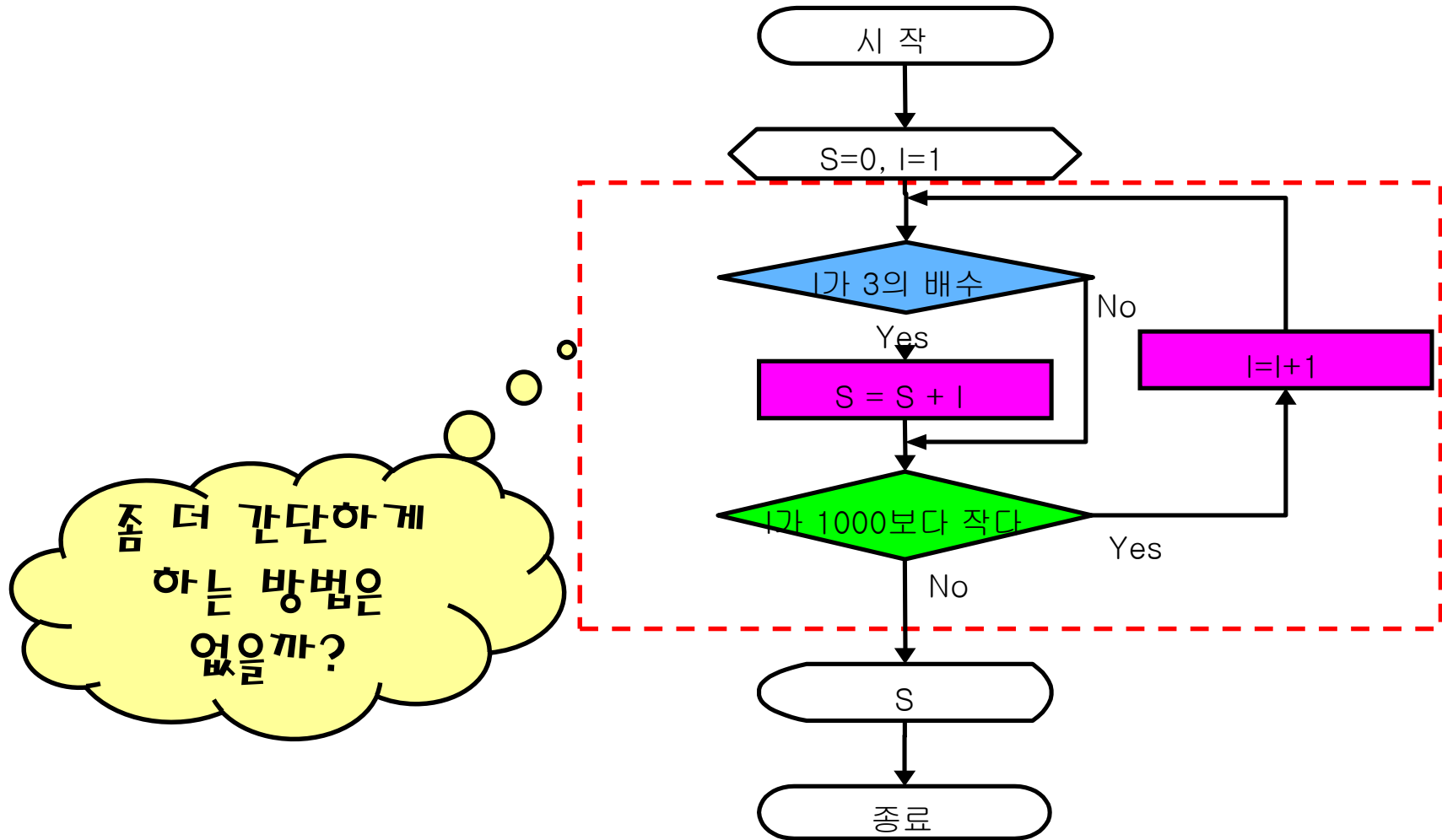
■ 아닌 경우에는 더하지 않고 다음 숫자로 넘어간다

■  $S = 3 + 6 + \dots + 999$



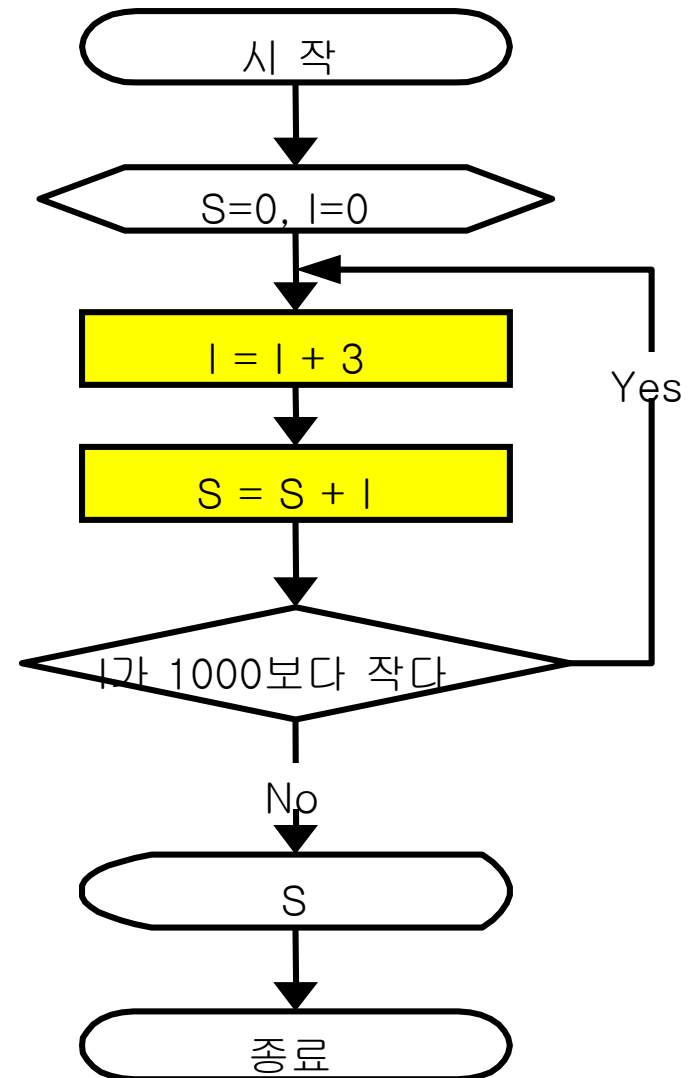
합의 공식?  
음... 아니군  
등차수열?  
등차수열 합의 공식이  
어떻게 되더라?  
등비수열... 음...

# Multiple number 문제

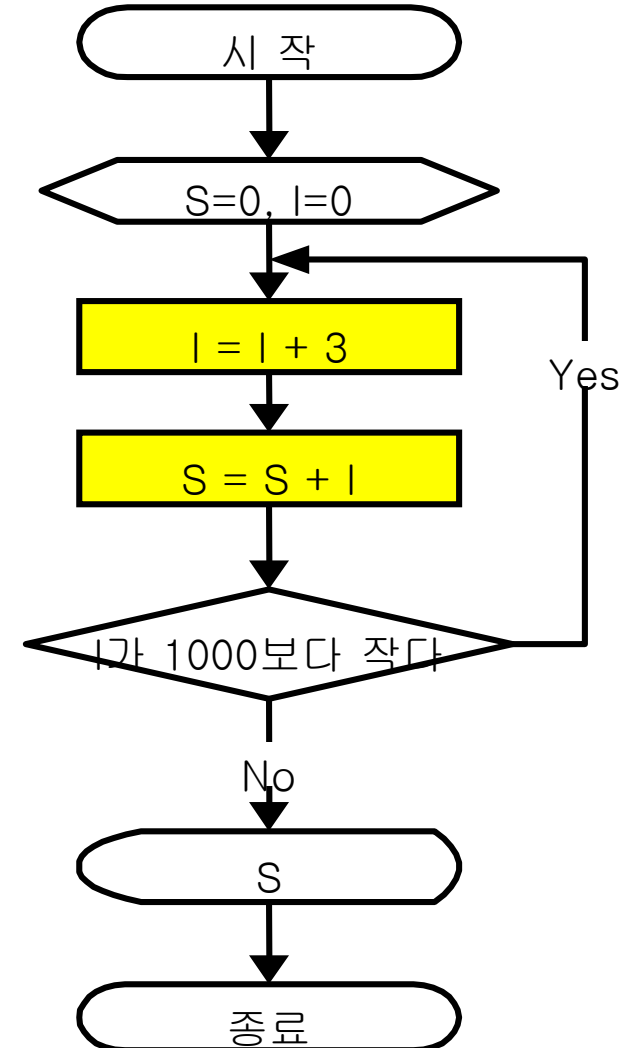
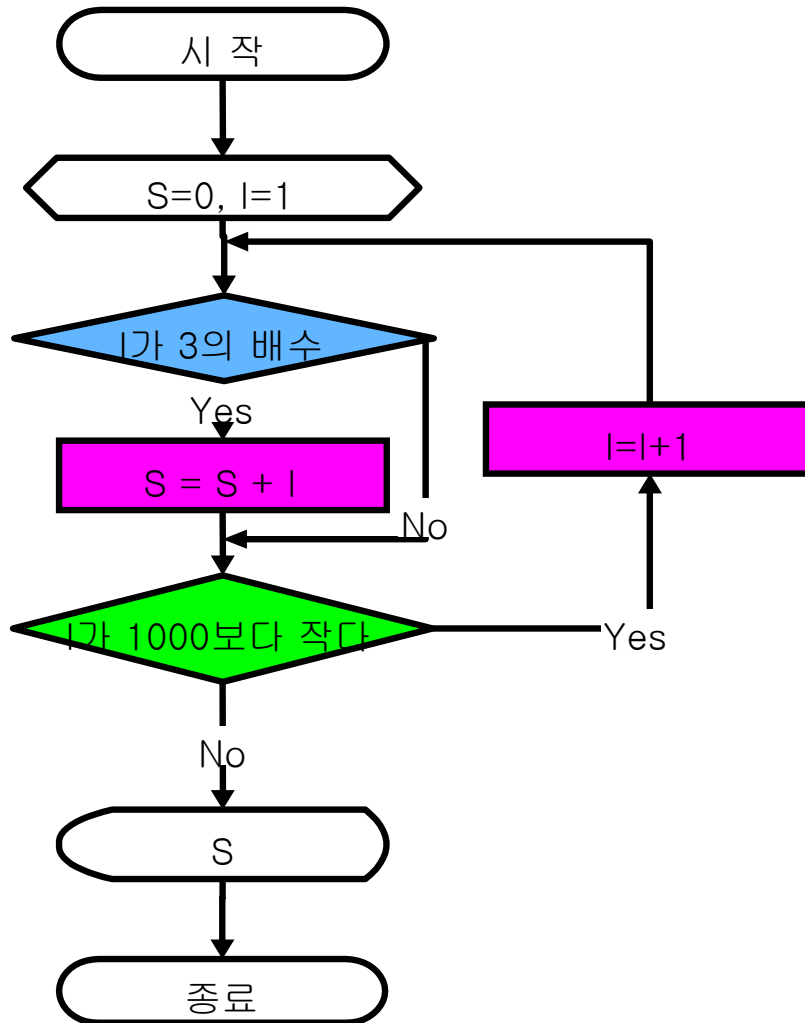


# Multiple number 문제

- 반복 구간 : 3에서 100
  - 더해지는 숫자에 대한 정의
    - 3의 배수만 더한다.
    - 다음 단계로 넘어갈 때, 3만큼 더해줌

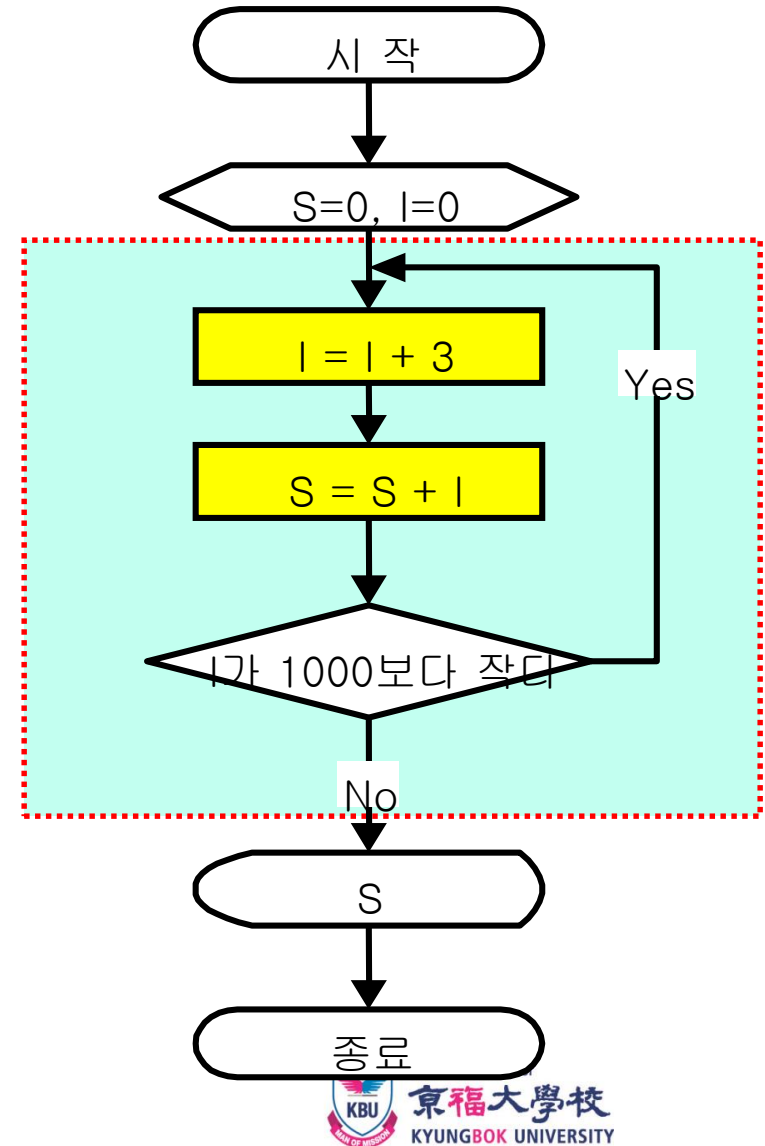
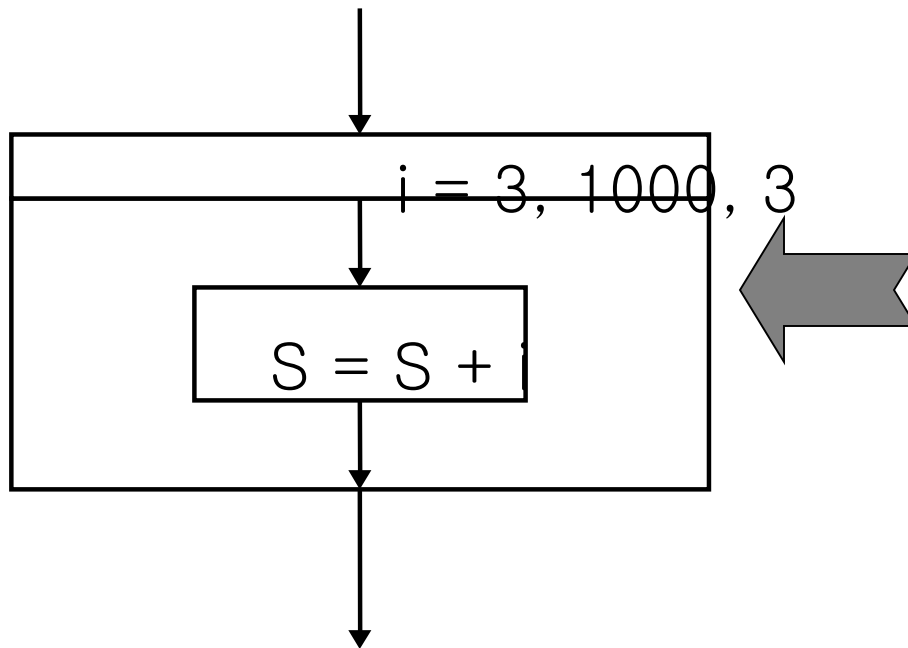


# Multiple number 문제



# Multiple number 문제

■ 반복문 부분 다음 기호로 대체



# Multiple number 문제

```
public static void main(String[] args) {  
    final int LAST = 100;  
    final int BAESU = 3;  
    int origin = 0;  
    int sum = 0;  
    int multiple = 0;  
  
    for (int loop = 1; loop <= LAST; loop++){  
        origin += loop;  
        if (loop % BAESU != 0)  
            sum += loop;  
        else  
            multiple += loop;  
    }  
    System.out.printf(" 1부터 %,d까지의 합 = %,d\n", LAST, origin);  
    System.out.printf(" 1부터 %,d까지 %d의 배수의 합 = %,d\n",  
        LAST, BAESU, multiple);  
    System.out.printf(" 1부터 %,d까지 %d의 배수를 제외한 수의 합 = %,d\n",  
        LAST, BAESU, sum);  
}
```



# Multiple number 문제

- 1부터 100까지의 정수 중에서 2와 3의 배수가 아닌 수의 총합을 구하시오.

1부터 100까지의 합 = 5,050

1부터 100까지 2의 배수의 합 = 2,550

1부터 100까지 3의 배수의 합 = 867

1부터 100까지 2와 3의 배수를 제외한 수의 합 = 1,633

# Multiple number 문제

```
public static void main(String[] args) {  
    final int LAST = 100;  
    final int BAESU1 = 2;  
    final int BAESU2 = 3;  
    int origin = 0;  
    int sum = 0;  
    int multiple2 = 0;  
    int multiple3 = 0;  
  
    for (int loop = 1; loop <= LAST; loop++) {  
        origin += loop;  
        if (loop % BAESU1 != 0 && loop % BAESU2 != 0)  
            sum += loop;  
        else if (loop % BAESU1 == 0)  
            multiple2 += loop;  
        else  
            multiple3 += loop;  
    }  
}
```

# Multiple number 문제

```
System.out.printf(" 1부터 %,d까지의 합 = %,d\n", LAST, origin);
System.out.printf(" 1부터 %,d까지 %d의 배수의 합 = %,d\n",
                  LAST, BAESU1, multiple2);
System.out.printf(" 1부터 %,d까지 %d의 배수의 합 = %,d\n",
                  LAST, BAESU2, multiple3);
System.out.printf(" 1부터 %,d까지 %d와 %d의 배수를 제외한 수의 합 = %,d\n",
                  LAST, BAESU1, BAESU2, sum);
}
```

# 순열 문제

- $1+(1+2)+(1+2+3)+(1+2+3+4)+\dots+(1+2+3+\dots+10)$ 의 결과를 계산하시오.

$$1 = 1$$

$$1 + 2 = 3$$

$$1 + 2 + 3 = 6$$

$$1 + 2 + 3 + 4 = 10$$

$$1 + 2 + 3 + 4 + 5 = 15$$

$$1 + 2 + 3 + 4 + 5 + 6 = 21$$

$$1 + 2 + 3 + 4 + 5 + 6 + 7 = 28$$

$$1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 = 36$$

$$1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 + 9 = 45$$

$$1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 + 9 + 10 = 55$$

$$\text{total} = 220$$

# 순열 문제

```
public static void main(String[] args) {  
    final int LAST = 10;  
    int temp = 0;  
    int total = 0;  
  
    for (int i = 1; i <= LAST; i++) {  
        temp = 0;  
        for (int j = 1; j <= i; j++) {  
            temp += j;  
            System.out.printf("%d", j);  
            System.out.printf(" + ");  
        }  
        System.out.printf("WbWbWb");  
        System.out.println(" = " + temp);  
        total += temp;  
    }  
  
    System.out.println("total = " + total);  
}
```

# 구구단

- 다음과 같이 구구단을 출력하는 Program을 작성하여보자.

출력하고자 하는 단을 입력하시오 : 3

3단 구구단

$$3 * 1 = 3$$

$$3 * 2 = 6$$

$$3 * 3 = 9$$

$$3 * 4 = 12$$

$$3 * 5 = 15$$

$$3 * 6 = 18$$

$$3 * 7 = 21$$

$$3 * 8 = 24$$

$$3 * 9 = 27$$

# 구구단

```
public static void main(String[] args) throws IOException {  
    Scanner keyboard = new Scanner(System.in);  
    int num;  
  
    while (true) {  
        System.out.print("출력하고자 하는 단을 입력하시오 : ");  
        num = keyboard.nextInt();  
        if (num >= 1 && num <= 9)  
            break;  
        else {  
            System.out.print("ERROR : 1 ~ 9사이의 숫자 입력 ");  
            System.in.read();  
        }  
    }  
}
```

# 구구단

```
System.out.printf("\n %d단 구구단 \n", num);  
for (int index = 1; index <= 9; index++)  
    System.out.printf("  %d * %d = %2d\n", num, index, num * index);  
}
```



# 구구단[심화]

- 다음과 같이 2단에서 9단까지 구구단을 출력하는 프로그램을 작성하시오

\*\*\*\* 구 구 단 \*\*\*\*

2*1=2	3*1=3	4*1=4	5*1=5	6*1=6	7*1=7	8*1=8	9*1=9
2*2=4	3*2=6	4*2=8	5*2=10	6*2=12	7*2=14	8*2=16	9*2=18
2*3=6	3*3=9	4*3=12	5*3=15	6*3=18	7*3=21	8*3=24	9*3=27
2*4=8	3*4=12	4*4=16	5*4=20	6*4=24	7*4=28	8*4=32	9*4=36
2*5=10	3*5=15	4*5=20	5*5=25	6*5=30	7*5=35	8*5=40	9*5=45
2*6=12	3*6=18	4*6=24	5*6=30	6*6=36	7*6=42	8*6=48	9*6=54
2*7=14	3*7=21	4*7=28	5*7=35	6*7=42	7*7=49	8*7=56	9*7=63
2*8=16	3*8=24	4*8=32	5*8=40	6*8=48	7*8=56	8*8=64	9*8=72
2*9=18	3*9=27	4*9=36	5*9=45	6*9=54	7*9=63	8*9=72	9*9=81

- 목적

- 중첩된 for문을 활용한 반복문 연습(이중 루프 사용)

# 구구단[심화]

## ■ 문제 분석

- 단순히 중첩된 for문을 이용하여 출력 형식을 맞추어 주기만 하면 됨
- 주의 해야 할 점은 반복문의 제어 변수가 변하는 값. 한 줄 단위로 출력이 되므로  $2*1=2$  다음에  $2*2=4$ 를 출력하는 것이 아니라  $3*1=3$ 을 줄 바꿈 없이 출력해야 함
- $9*1=9$ 를 출력하고 나면 줄을 바꾸어  $2*2=4$   $3*2=6$ ....  
 $9*2=18$ 을 출력한 후 다시 줄을 바꾸어 출력하도록 함

# 구구단[심화]

- 입출력 변수

- int out;

- 바깥쪽 for문의 제어 변수로서 1에서 9까지의 값을 가짐

- int in;

- 안쪽 for 문의 제어 변수

- 2단에서 9단까지의 값을 가지는 변수

# 구구단[심화]

## ■ 가상언어 표현

1. **title**을 출력한다.

2. **out = 1**에서 9까지

3. **in = 2**에서 9까지

4. **in \* out = in \* out**을 출력한다.

5. 한 줄을 바꾼다.

# 구구단[심화]

```
public static void main(String[] args) {  
    final int LAST = 9;  
    int value;  
  
    System.out.println("~~~~~ ***** 구 구 단 *****");  
    for (int row = 1; row <= LAST; row++) {  
        int col = 2;  
        do {  
            value = col + row;  
            System.out.printf("%d * %d = %2d ", col, row, value);  
        } while (col++ < LAST);  
        System.out.println();  
    }  
}
```

# 구구단[심화]

\*\*\*\* 구 구 단 \*\*\*\*

$2 * 1 = 2$	$2 * 2 = 4$	$2 * 3 = 6$	$2 * 4 = 8$	$2 * 5 = 10$	$2 * 6 = 12$	$2 * 7 = 14$	$2 * 8 = 16$	$2 * 9 = 18$
$3 * 1 = 3$	$3 * 2 = 6$	$3 * 3 = 9$	$3 * 4 = 12$	$3 * 5 = 15$	$3 * 6 = 18$	$3 * 7 = 21$	$3 * 8 = 24$	$3 * 9 = 27$
$4 * 1 = 4$	$4 * 2 = 8$	$4 * 3 = 12$	$4 * 4 = 16$	$4 * 5 = 20$	$4 * 6 = 24$	$4 * 7 = 28$	$4 * 8 = 32$	$4 * 9 = 36$
$5 * 1 = 5$	$5 * 2 = 10$	$5 * 3 = 15$	$5 * 4 = 20$	$5 * 5 = 25$	$5 * 6 = 30$	$5 * 7 = 35$	$5 * 8 = 40$	$5 * 9 = 45$
$6 * 1 = 6$	$6 * 2 = 12$	$6 * 3 = 18$	$6 * 4 = 24$	$6 * 5 = 30$	$6 * 6 = 36$	$6 * 7 = 42$	$6 * 8 = 48$	$6 * 9 = 54$
$7 * 1 = 7$	$7 * 2 = 14$	$7 * 3 = 21$	$7 * 4 = 28$	$7 * 5 = 35$	$7 * 6 = 42$	$7 * 7 = 49$	$7 * 8 = 56$	$7 * 9 = 63$
$8 * 1 = 8$	$8 * 2 = 16$	$8 * 3 = 24$	$8 * 4 = 32$	$8 * 5 = 40$	$8 * 6 = 48$	$8 * 7 = 56$	$8 * 8 = 64$	$8 * 9 = 72$
$9 * 1 = 9$	$9 * 2 = 18$	$9 * 3 = 27$	$9 * 4 = 36$	$9 * 5 = 45$	$9 * 6 = 54$	$9 * 7 = 63$	$9 * 8 = 72$	$9 * 9 = 81$



Futuristic Innovator

京福大學校  
KYUNGBOK UNIVERSITY

# 구구단[심화]

```
public static void main(String[] args) {  
    final int LAST = 9;  
  
    System.out.print("WtWtWtWtWtWtWtWtWtWtWt**** 구 구 단 ****");  
    for (int row = 2; row <= LAST; row++) {  
        System.out.println();  
        for (int col = 1; col <= LAST; col++) {  
            System.out.print('Wt');  
            System.out.print(row + " * " + col + " = " + row * col);  
        }  
    }  
}
```

# Pascal's triangle

- Keyboard에서 10보다 작은 숫자  $n$ 을 입력 받아 다음과 같이 출력하는 프로그램을 작성하라.

(예)  $n = 5$

```
      1
     2 3
    4 5 6
   7 8 9 10
  11 12 13 14 15
```



# Pascal's triangle

```
public static void main(String[] args) {
    Scanner keyboard = new Scanner(System.in);
    int i, j, k = 1, max, space;

    do {
        System.out.print("파스칼의 삼각형의 수 입력 : ");
        max = keyboard.nextInt();
    } while (max <= 0 || max > 10);
    for (i = 1; i <= max; i++) {
        for (space = 0; space < max - i; space++)
            System.out.print(" ");
        for (j = 1; j <= 2 * i - 1; j++) {
            if (j % 2 == 0)
                continue;
            System.out.printf("%2d ", k++);
        }
        System.out.println();
    }
}
```

# 거꾸로 수

- 9자리 이하의 정수를 입력받아 입력된 수의 거꾸로 수를 구하는 프로그램을 작성하여라.

9자리 이하의 숫자 입력 : 6789896785678

9자리 이하의 숫자 입력 : 678956789

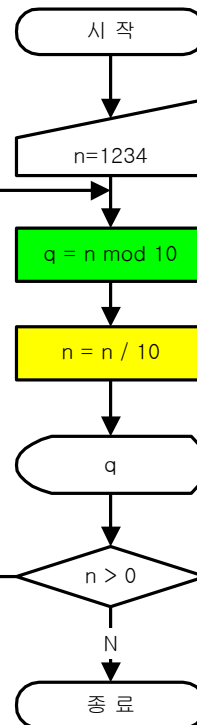
입력받은 678,956,789의 거꾸로 수는 987,659,876 입니다.

# 거꾸로 수

- 숫자 1234를 읽어서 4321인 역으로 출력하는 순서도
- 고려사항
  - 각각의 자리수를 어떻게 분리해 내는가?
    1. 마지막(일의 자리수)자리수를 간단하게 분리하는 방법?
      - 나머지 연산자(10)
      - 예)  $1234 \% 10 = 4$
    2. 마지막 자리수를 제외한 숫자를 추출할 수 있는 방법?
      - 10으로 나눈 몫을 취한다.
      - 예)  $1234 / 10 = 123$ (정수만)
  - 1, 2 단계 반복

# 거꾸로 수

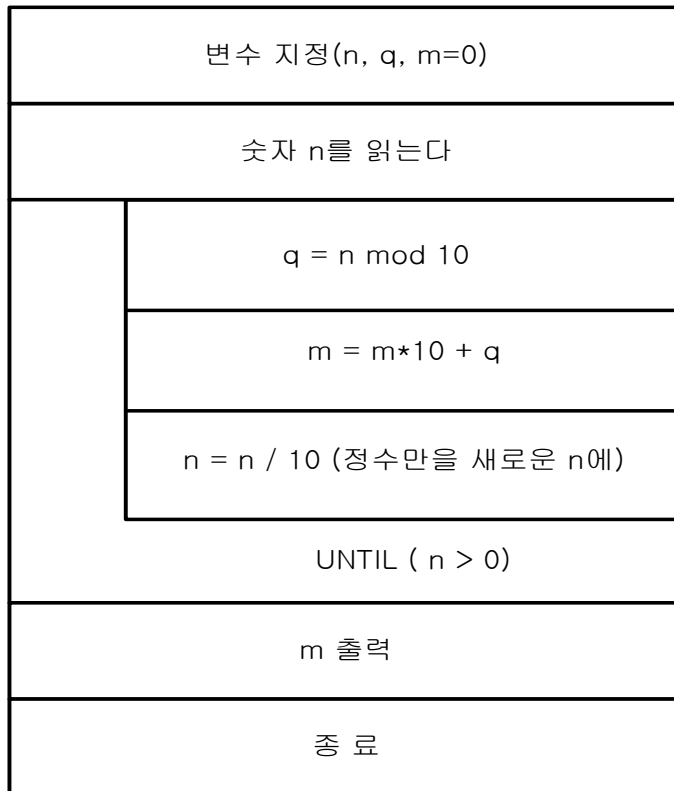
마지막 자리수  
분리하기  
예)  $1234 \% 10 = 4$



나머지 자리수 정리  
예)  $1234 / 10 = 123$

# 거꾸로 수

- 어떤 숫자  $n$ 를 읽어서 이를 역으로 만들어 출력하는 NS chart와 의사코드가 다음과 같다. 이를 C 프로그램으로 표현하시오. 예를 들어, 숫자 1234를 읽어서 4321인 값으로 변경하고 출력



```
m=0

READ  n

DO
    q = n mod 10
    m = m*10 + q
    n = n / 10 (정수만 새로운 n에 할당)
UNTIL (n > 0)

PRINT m
```

# 거꾸로 수

```
public static void main(String[] args) {  
    Scanner keyboard = new Scanner(System.in);  
    final int MAX = 999999999;  
    long number;  
    int value;  
    long reverse = 0L;  
  
    do {  
        System.out.print(" 9자리이하의 숫자 입력 : ");  
        number = keyboard.nextInt();  
    } while (MAX < number || number < -MAX) ;  
}
```



Futuristic Innovator

京福大學校  
KYUNGBOK UNIVERSITY

# 거꾸로 수

```
System.out.printf(" 입력 받은 %,d의 거꾸로 수는", number);
do {
    reverse *= 10;
    value = (int) (number % 10);
    reverse += value;
    number = number / 10;
} while (number != 0);
System.out.printf(" %,d 입니다.\n", reverse);
}
```



Futuristic Innovator

京福大學校  
KYUNGBOK UNIVERSITY

# 숫자의 합

- Keyboard로부터 양의 정수를 입력 받아 각 자리수의 합을 구하는 프로그램을 작성하여라.

## Sample

양의 정수 입력 : 12345

입력한 정수는 12,345 입니다.

숫자의 개수는 5개 입니다.

숫자의 합은 15 입니다.



# 숫자의 합

```
public static void main(String[] args) throws IOException {  
    Scanner keyboard = new Scanner(System.in);  
    int num;  
    int count = 0;  
    int sum = 0;  
  
    do {  
        System.out.printf(" 양의 정수 입력 : ");  
        num = keyboard.nextInt();  
        if (num > 0)  
            break;  
        else {  
            System.out.print("오류\n");  
            System.in.read();  
        }  
    } while (true);  
}
```

# 숫자의 합

```
System.out.printf("\n 입력한 정수는 %,d 입니다.", num);
while (num != 0) {
    count++;
    sum += num % 10;
    num /= 10;
}
System.out.printf("\n 숫자의 개수는 %d개 입니다.", count);
System.out.printf("\n 숫자의 합은 %d 입니다.\n", sum);
}
```



Futuristic Innovator

京福大學校  
KYUNGBOK UNIVERSITY

# 약수 구하기

- 양의 정수를 입력으로 받아서 이 수의 모든 약수를 구하는 프로그램을 작성하시오
- 0또는 음수가 입력되면 프로그램을 종료한다

24의 약수 : 1, 2, 3, 4, 6, 12, 24

- 목적
  - while문과 for문을 이용한 중첩된 반복문 활용 연습

# 약수 구하기

## ■ 문제 분석

- 약수란 1외의 수로 나누어 나머지가 0인 양의 정수

## ■ 입력

- 정수(num) - int

## ■ 출력

- 약수 (1 ~ num 사이의 정수)

## ■ 계산 방법

- 1부터 num까지를 젓수(나누는 수)로 나누어 나머지가 0이면 약수

$$8 \div 1 = 8$$

$$8 \div 2 = 4$$

$$8 \div 3 = 2 \cdots 2$$

$$8 \div 4 = 2$$

$$8 \div 5 = 1 \cdots 3$$

$$8 \div 6 = 1 \cdots 2$$

$$8 \div 7 = 1 \cdots 1$$

$$8 \div 8 = 1$$

→ 8의 약수 : 1, 2, 4, 8

# 약수 구하기

- 문제 해결 방법
  - 입력 데이터 처리 방법
    - 하나의 데이터를 입력하여
      - 양수이면 이 수의 약수를 구해서 출력하고 다시 입력을 받는다. (while 문 사용)
      - 0이나 음수가 입력되면 종료

# 약수 구하기

## ■ 실행 시 화면

- 입력형태

- 프로그램이 실행될 때 다음과 같은 메시지를 내보내어 num 값을 입력 받는다.
- “수를 입력하시오 ?”

- 출력형태

- 약수를 구한 후 다음과 같이 출력한다.
- “24의 약수 : 1, 2, 3, 4, 6, 8, 12, 24”

# 약수 구하기

## ■ 가상 언어 표현

num을 입력한다.

num > 0 동안 다음을 반복

num을 출력.

div = 1에서 num까지 변할 동안 다음을 반복

(num % div) = 0이면 div를 출력

다시 num값을 입력.

0이나 음수가 입력되면 종료 된다.

# 약수 구하기

```
public static void main(String[] args) {  
    Scanner keyboard = new Scanner(System.in);  
    int div;  
    int number;  
    String result = "";  
  
    System.out.print("약수를 원하는 정수를 입력 : ");  
    number = keyboard.nextInt();  
  
    result = String.format("\n %d의 약수 = ", number);  
    for (int count = 1; count <= number / 2 + 1; count++) {  
        div = number % count;  
        if (div == 0)  
            result += String.format(" %3d, ", count);  
    }  
    result += String.format(" %3d", number);  
  
    System.out.println(result);  
}
```



# Perfect number

- 양의 정수를 입력 받아 완전수인지를 검사해보자
- 완전수
  - 자기 자신을 제외한 양의 약수(진 약수)를 더했을 때 자기 자신이 되는 양의 정수를 말함
  - 최초 다섯 개의 완전수는 6, 28, 496, 8128, 33550336

# Perfect number

```
public static void main(String[] args) {  
    Scanner keyboard = new Scanner(System.in);  
  
    System.out.print("정수 입력 : ");  
    int num = keyboard.nextInt();  
  
    System.out.print(num + "의 약수는 : " + 1 + ", ");  
    int sum = 1;  
    for (int i = 2; i <= num / 2; i++) {  
        if ((num % i) == 0) {  
            System.out.print(i + ", ");  
            sum += i;  
        }  
    }  
    System.out.println(num + "입니다.");  
    if (sum == num)  
        System.out.println(num + "은 완전수입니다.");  
    else  
        System.out.println(num + "은 완전수가 아닙니다.");  
}
```

# 최대 공약수/최소 공배수

- 두수를 입력 받아 최대 공약수와 최소 공배수를 구하여라.

# palindrome 구하기

- 회문수(palindrome)를 구하는 프로그램을 작성하여라  
(나머지 연산자를 이용)
- 회문수
  - 숫자를 거꾸로 읽어도 앞으로 읽는 것과 같은 수를 말한다.
  - 예) '12321'이나 '13531'같은 수를 말한다

# palindrome 구하기

```
public static void main(String[] args) {  
    Scanner keyboard = new Scanner(System.in);  
    int number;  
    int temp;  
    int result = 0;  
  
    System.out.print("정수 입력 : ");  
    number = keyboard.nextInt();  
  
    temp = number;  
    while (temp != 0) {  
        result = result * 10 + temp % 10;  
        temp /= 10;  
    }  
}
```

# palindrome 구하기

```
if (number == result)
    System.out.println( number + "는 회문수 입니다.");
else
    System.out.println( number + "는 회문수가 아닙니다.");
}
```

# Alphabet Show

- do-while문을 이용하여 'a'부터 'z'까지 출력하는 프로그램을 작성하시오.

abcdefghijklmnopqrstuvwxyz

# Alphabet Show

```
public static void main(String[] args) {  
    int ch = 'a';  
  
    do {  
        System.out.print((char) ch);  
        ch++;  
    } while (ch <= 'z');  
}
```



# Alphabet Show[심화]

- 알파벳 'A' ~ 'Z'까지 한 칸씩 움직이는 알파벳 테이블을 만들어라.

ABCDEFGHIJKLMNOPQRSTUVWXYZ  
BCDEFGHIJKLMNOPQRSTUVWXYZA  
CDEFGHIJKLMNOPQRSTUVWXYZAB  
DEFGHIJKLMNOPQRSTUVWXYZABC  
EFGHIJKLMNOPQRSTUVWXYZABCD  
FGHIJKLMNOPQRSTUVWXYZABCDE  
GHIJKLMNOPQRSTUVWXYZABCDEF  
HIJKLMNOPQRSTUVWXYZABCDEFG  
IJKLMNOPQRSTUVWXYZABCDEFGH  
JKLMNOPQRSTUVWXYZABCDEFGHI  
LMNOPQRSTUVWXYZABCDEFGHIJ  
MNOPQRSTUVWXYZABCDEFGHIJK  
NOPQRSTUVWXYZABCDEFGHIJKL  
OPQRSTUVWXYZABCDEFGHIJKLM  
PQRSTUVWXYZABCDEFGHIJKLMNO  
QRSTUVWXYZABCDEFGHIJKLMNOP  
RSTUVWXYZABCDEFGHIJKLMNOPQ  
STUVWXYZABCDEFGHIJKLMNOPQR  
TUVWXYZABCDEFGHIJKLMNOPQRS  
UVWXYZABCDEFGHIJKLMNOPQRST  
VWXYZABCDEFGHIJKLMNOPQRSTU  
WXYZABCDEFGHIJKLMNOPQRSTUV  
XYZABCDEFGHIJKLMNOPQRSTUVW  
YZABCDEFGHIJKLMNOPQRSTUVWX  
ZABCDEFGHIJKLMNOPQRSTUVWXY



Futuristic Innovator

京福大學校  
KYUNGBOK UNIVERSITY

# Alphabet Show [심화]

```
public static void main(String[] args) {  
    for (int row = 0; row <= 'Z' - 'A'; row++) {  
        int ch = 'A' + row;  
        for (int loop = 'A'; loop <= 'Z'; loop++) {  
            if (ch > 'Z')  
                ch = 'A';  
            System.out.printf("%c", ch++);  
        }  
        System.out.println();  
    }  
}
```



Futuristic Innovator

京福大學校  
KYUNGBOK UNIVERSITY

# 온도계산 문제

- $0^{\circ}\text{F}$  에서  $100^{\circ}\text{F}$  사이의 화씨 온도를  $10^{\circ}\text{F}$  간격으로 섭씨온도( $^{\circ}\text{C}$ )로 변환하는 프로그램을 작성하시오.

단, 섭씨온도 =  $5 / 9$  (화씨온도 - 32)

- 목적
  - do - while문을 활용한 반복문 연습

# 온도계산 문제

## ■ 문제 해결 방법

- 섭씨온도 =  $5 / 9$  (화씨온도 - 32)이다
- 화씨온도 값을 식에 대입하여 섭씨온도를 구한다
- 10씩 증가 시켜 가면서 섭씨온도를 구하여 출력
- lower, upper, step을 각각 변수로 사용하여 다른 구간의 화씨온도도 쉽게 구할 수 있도록 한다

# 온도계산 문제 : 입출력 설계

## ■ 입출력 변수선정

- float fahr ;

- 화씨 온도 값을 저장할 변수

- float celsius;

- 섭씨 온도 값을 저장할 변수 (출력 변수)

- int lower;

- 화씨온도의 최저 값을 저장할 변수 (0을 저장)

- int upper;

- 화씨온도의 최고 값을 저장할 변수로 사용(100을 저장)

- int step;

- 온도 간격을 저장할 변수 (10을 저장)

# 온도계산 문제

## ■ 문제 파악

$$\text{섭씨온도} = 5 / 9 (\text{화씨온도} - 32)$$

## ■ 출력 형태

화씨온도	섭씨온도
0	?
10	?
100	?

# 온도 계산 문제

```
public static void main(String[] args) {  
    float celsius;  
  
    System.out.printf("%4s %6s\n", "화씨", "섭씨");  
    for (int fahr = 0; fahr <= 100; fahr += 10) {  
        celsius = (fahr - 32.0f) * 5.0f / 9.0f;  
        System.out.printf("%4d %10.1f\n", fahr, celsius);  
    }  
}
```

# 이자 저축

- 어떤 사람이  $x$  만원을 연리  $y\%$ 의 단리이자와 복리이자로 10년간 예금하면 10년 후의 원리금 합계는 ?





# 이자 저축

## ■ 문제 분석

- 단리는 원금에만 이자가 붙는 것이고, 복리는 처음에는 원금에 이자가 붙지만, 그 다음부터는 원금+이자에 다시 이자가 붙는 방식이다. 1년 만기 저축 상품에 가입한다면 1년 후에 원금에 더해 이자를 받게 된다. 단리 상품과 복리 상품이 있다면 당연히 복리 상품에 가입을 하는 것이 이익이다

- 단리 만기 금액 = 원금 + 원금 × 이자율 × 기간

- 복리 만기 금액 = 원금 × (1 + 이자율)<sup>기간</sup>

# 이자 저축

원금 입력 : 1000000

이율(%) 입력 : 4

1년 후의 단리 원리금 합계 = 1,040,000 원, 복리 원리금 합계 = 1,040,000 원
2년 후의 단리 원리금 합계 = 1,080,000 원, 복리 원리금 합계 = 1,081,600 원
3년 후의 단리 원리금 합계 = 1,120,000 원, 복리 원리금 합계 = 1,124,864 원
4년 후의 단리 원리금 합계 = 1,160,000 원, 복리 원리금 합계 = 1,169,859 원
5년 후의 단리 원리금 합계 = 1,200,000 원, 복리 원리금 합계 = 1,216,653 원
6년 후의 단리 원리금 합계 = 1,240,000 원, 복리 원리금 합계 = 1,265,319 원
7년 후의 단리 원리금 합계 = 1,280,000 원, 복리 원리금 합계 = 1,315,932 원
8년 후의 단리 원리금 합계 = 1,320,000 원, 복리 원리금 합계 = 1,368,569 원
9년 후의 단리 원리금 합계 = 1,360,000 원, 복리 원리금 합계 = 1,423,312 원
10년 후의 단리 원리금 합계 = 1,400,000 원, 복리 원리금 합계 = 1,480,244 원



Futuristic Innovator

京福大學校

KYUNGBOK UNIVERSITY

# 이자 저축

■ <https://wisenomics.com/compound-interest-calculator/>

wisenomics

오늘 시작하시고 최대 10만원의 쿠폰을 받으세요

원금(원)\* 원금  
₩0

연 이자율(%)\*

기간(년)\*

복리 계산 단위  
1년(연복리) ▼  
연복리, 월복리, 6개월 복리, 분기 복리 중 하나를 선택하세요.

위에서 입력한 내용의 계산 결과가 아래에 자동으로 표시됩니다.

단리 이자 단리:원금+이자  
₩0 ₩0

복리 이자 복리:원금+이자  
₩0 ₩0

와이즈노믹스에 올라오는 새 글 소식을 이메일로 알려 드립니다.  
이메일 주소... 구독

많이 찾는 글과 페이지

- 주택 청약 1순위 조건 및 2순위 조건
- 자동차 할부 구매 vs. 리스: 현명한 선택은?
- 단리·복리 계산법
- 신용 카드 할부 이자(수수료) 계산기
- 복리 계산기
- 청구할인 이란? 환급할인과의 차이와 확인 방법

# 이자 저축

```
public static void main(String[] args) {  
    Scanner keyboard = new Scanner(System.in);  
    final int TERM = 10;  
    int investment;  
    float rate;  
    float simple, compound;  
  
    System.out.print("원금 입력 : ");  
    investment = keyboard.nextInt();  
    System.out.print("이율(%) 입력 : ");  
    rate = keyboard.nextFloat();  
}
```



Futuristic Innovator

京福大學校  
KYUNGBOK UNIVERSITY

# 이자 저축

```
rate /= 100.0f;
compound = investment;
for (int year = 1; year <= TERM; year++) {
    simple = investment * (1 + rate * year);
    compound += (rate * compound);
    System.out.printf(" %2d년 후의 단리 원리금 합계 = %,8.0f 원,
                        복리 원리금 합계 = %,8.0f 원\n",
                        year, simple, compound);
}
```

# 1억원 만들기

- 첫날은 1원부터 시작하여 다음날은 전날의 2배의 저축을 하면 언제 1억원에 도달할 수 있을까요?

오늘 저축 금액 =	1 원,	1 일 =	1 원
오늘 저축 금액 =	2 원,	2 일 =	3 원
오늘 저축 금액 =	4 원,	3 일 =	7 원
오늘 저축 금액 =	8 원,	4 일 =	15 원

오늘 저축 금액 =	524,288 원,	20 일 =	1,048,575 원
오늘 저축 금액 =	1,048,576 원,	21 일 =	2,097,151 원
오늘 저축 금액 =	2,097,152 원,	22 일 =	4,194,303 원
오늘 저축 금액 =	4,194,304 원,	23 일 =	8,388,607 원
오늘 저축 금액 =	8,388,608 원,	24 일 =	16,777,215 원
오늘 저축 금액 =	16,777,216 원,	25 일 =	33,554,431 원
오늘 저축 금액 =	33,554,432 원,	26 일 =	67,108,863 원
오늘 저축 금액 =	67,108,864 원,	27 일 =	134,217,727 원
1억원 ....27일 만에 달성			

# 1억원 만들기

```
public static void main(String[] args) {  
    int day;  
    long pay = 0L;  
    long temp = 0L;  
  
    for (day = 1; true; day++) {  
        temp = (day == 1) ? 1 : temp * 2;  
        pay += temp;  
        System.out.printf("오늘 저축 금액 = %,10d 원, %3d 일 = %,11d 원\n",  
                           temp, day, pay);  
  
        if (pay > 1000000000L)  
            break;  
    }  
    System.out.printf(" 1억원 ....%d일 만에 달성\n", day);  
}
```

# 1억 원으로 살기

- 어떤 퇴직자가 1억 원을 년 5%의 이자를 주는 보통예금을 하고 매월 2백 만원씩 생활비를 쓰면 얼마나 살아갈 수 있을까 ?





# 1억원으로 살기

```
public static void main(String[] args) {  
    final int MONEY = 100000000;  
    final int MONTH = 2000000;  
    final float RATE = 5.0f / 100;  
    float deposit = (float) MONEY;  
    int month = 0;  
  
    do {  
        if (month != 0 && month % 12 == 0)  
            deposit *= (1 + rate);  
        if (deposit <= MONTH)  
            break;  
        deposit -= MONTH;  
        System.out.printf("Wn %2d월 은행 잔고 = %,10.0f 원",  
                           month + 1, deposit);  
  
        month++;  
    } while (true);  
}
```



Futuristic Innovator

京福大學校  
KYUNGBOK UNIVERSITY

# 1억원으로 살기

```
System.out.printf("Wn %2d월 최종 잔고 = %,10.0f 원",  
                                                           month, deposit);  
System.out.printf("Wn %d년 %d개월 살았다",  
                                                           month / 12, month % 12);  
}
```

# “I love you !” 를 만난 날만큼 보여 주기

```
public static void main(String[] args) throws IOException {  
    Scanner keyboard = new Scanner(System.in);  
    int day;  
  
    do {  
        System.out.print(" 우리 만남이 몇일지났나요 ? ");  
        day = keyboard.nextInt();  
  
        if (day <= 0) {  
            System.err.print("오류");  
            System.in.read();  
        } else  
            break;  
    } while (true);  
  
    for (int loop = 1; loop <= day; loop++)  
        System.out.println("I Love You !!");  
}
```

# 추의 무게

- 2g, 3g, 5g의 추가 각각 10개씩 있다. 이 추를 가지고 전체 무게가 test g이 되는 추의 조합을 찾는 프로그램을 만들어라.



# 추의 무게

```
public static void main(String[] args) {  
    Scanner keyboard = new Scanner(System.in);  
    final int MAX = 10;  
    int two, three, five;  
    int weight, total;  
  
    System.out.print("몇 Gram의 추의 조합이 필요합니까 ?");  
    weight = keyboard.nextInt();  
}
```

# 추의 무게

```
System.out.printf("   *** %d gram 추의 조합 ***\n", weight);
for (two = 1; two <= MAX; two++)
    for (three = 1; three <= MAX; three++)
        for (five = 1; five <= MAX; five++) {
            total = two * 2 + three * 3 + five * 5;
            if (total == weight)
                System.out.printf(" 2g = %2d개, 3g = %2d개,
                                   5g = %2d개\n", two, three, five);
        }
}
```

# 생선 판매 금액

- 생선 가게에서 마리당 500원하는 생선을 100마리 구입하여 판매하고자 한다. 당일 판매하지 못하는 생선은 버려야 한다고 할 때, 전체 이윤이 최소한 10,000원이 되고, 구매 생선의 80%는 판매하고자 한다면 판매 단가를 얼마로 하면 되겠는가?
- 단가를 높여 판매하면 마리당 이윤은 높을지 모르지만 판매량은 감소하여 전체 이윤이 줄어든 것이고, 마리당 단가를 낮추면 판매량은 늘어날지 모르지만 마리당 이윤이 적어 역시 전체 이윤은 줄어든 것이다

# 생선 판매 금액

## ■ 문제 분석

- 판매 단가를  $x$ , 판매 마리 수를  $y$ 로 놓으면
- 구매 총액 = 구매 단가 \* 구매 마리수
- 판매 총액 = 판매 단가( $x$ ) \* 판매 마리 수( $y$ )
- 이윤 = 판매 총액 - 구매 총액
- 즉,  $x * y - 500 * 100 \geq 10000$
- 판매를 80%수준으로 유지
- 800원, 75개정도



# 생선 판매 금액

## ■ 문제 분석

- Data는 무엇이고, Information는 무엇인가?

## ■ 입력

- 이윤

- 구매 단가

- 구매 마리수

- 구매 마리의 80%는 판매

## ■ 출력

- 판매 단가

- 판매 마리수

# 생선 판매 금액

```
public static void main(String[] args) {  
    int fish = 100;  
    int price = 500;  
    int i, sale = 0;  
    int total = 100 * price;  
  
    for (i = 1; i <= fish; i++) {  
        for (sale = price; sale >= price; sale += 10) {  
            int test = (i * sale) - total;  
            if (test >= 10000)  
                break;  
        }  
        if (i >= 80)  
            break;  
    }  
    System.out.printf("판매 수량 : %d, 판매가격 : %d\n", i, sale);  
}
```



# 로마 숫자

- 1부터 10,000미만의 아라비아 숫자를 입력 받아 로마 숫자로 변환하는 프로그램을 작성하여라



# 로마숫자

■ <https://latina.bab2min.pe.kr/x/numconverter/4772>

살아있는 라틴어 / 공지사항 / 라틴어 명언들 / 도구 / 라틴어 텍스트 / 라틴어 Q&A / 링크

도구 / 로마 숫자 변환기

### 로마 숫자 변환기

아라비아 숫자를 입력하면 로마 숫자로, 로마 숫자를 입력하면 아라비아 숫자로 변환해줍니다. 10000 이상의 숫자에 대해서는 Vinculum 표기법을 이용한 로마 숫자로 변환합니다.

변환결과

## 도구

통합검색

사전

- 라틴어-한국어 사전
- 한국어-라틴어 사전
- 라틴어 단어 색인

# 로마 숫자

아라비아	로마 수	아라비아	로마 수	아라비아	로마 수
1	I	14	XIV	90	XC
2	II	15	XV	100	C
3	III	16	XVI	200	CC
4	IV	17	XVII	300	CCC
5	V	18	XVIII	400	CD
6	VI	19	XIX	500	D
7	VII	20	XX	600	DC
8	VIII	30	XXX	700	DCC
9	IX	40	XL	800	DCCC
10	X	50	L	900	CM
11	XI	60	LX	1000	M
12	XII	70	LXX	4000	MF
13	XIII	80	LXXX	5000	F

# 로마 숫자

## ■ 변환 규칙

- $1 = I, 5 = V, 10 = X, 50 = L, 100 = C, 500 = D, 1000 = M, 5000 = F$  (이 숫자들은 기준이 되는 숫자)
- 이 숫자를 기준으로 왼쪽에 있으면 적은 수, 오른쪽에 있으면 많은 수
  - 예)  $24 = XXIV = XX + IV = 20 + 4 = 24$   
 $26 = XXVI = XX + VI = 20 + 6 = 26$
  - 4의 경우는 5를 의미하는 V앞에 일을 의미하는 I가 있다. 즉, 5에서 1을 빼는 것이다.
  - 6의 경우는 5를 의미하는 V뒤에 일을 의미하는 I가 있다. 즉, 5에서 1을 더하는 것이다
- 숫자 왼쪽에 다른 숫자를 두면 빼지고, 오른쪽에 두면 더해진다.
  - 예)  $4 - IV, 999 - IM, 564 - DLXIV$

# 로마 숫자

```
public static void main(String[] args) throws IOException {  
    Scanner keyboard = new Scanner(System.in);  
    int number, temp;  
    String roma = "";  
  
    while (true) {  
        System.out.printf("\n 숫자 입력 : ");  
        number = keyboard.nextInt();  
        if (number >= 1 && number <= 3000)  
            break;  
        else {  
            System.out.println("Error : 1 ~ 3000 사이의 숫자 입력");  
            System.in.read();  
        }  
    }  
}
```

# 로마 숫자

```
temp = number;
while (temp >= 1000) {
    roma += 'M';
    temp -= 1000;
}
if (temp >= 900) {
    roma += "CM";
    temp -= 900;
}
while (temp >= 500) {
    roma += "D";
    temp -= 500;
}
if (temp >= 400){
    roma += "CD";
    temp -= 400;
}
```



# 로마 숫자

```
while (temp >= 100){  
    roma += "C";  
    temp -= 100;  
}  
if (temp >= 90){  
    roma += "XC";  
    temp -= 90;  
}  
while(temp >= 50){  
    roma += "L";  
    temp -= 50;  
}  
if(temp >= 40){  
    roma += "XL";  
    temp -= 40;  
}
```

# 로마 숫자

```
while(temp >= 10){  
    roma += "X";  
    temp -= 10;  
}  
if(temp >= 9){  
    roma += "IX";  
    temp -= 9;  
}  
while(temp >= 5){  
    roma += "V";  
    temp -= 5;  
}  
if(temp >= 4){  
    roma += "IV";  
    temp -= 4;  
}
```

# 로마 숫자

```
while(temp >= 1){  
    roma += "|";  
    temp--;  
}
```

```
System.out.printf( "   %d = %s\n", number, roma);  
}
```

# 호수

- 호수의 고요한 수면에 조그마한 돌을 던졌을 때, 생기는 파문의 반지름이 매초 15cm의 속도로 커진다고 한다. 돌을 던진 후 5초간의 원형 파문의 넓이를 계산하는 프로그램을 작성하여라.



# 호수

```
public static void main(String[] args) {  
    final float PI = 3.141592f;  
    float area;  
    int radius = 15, second;  
  
    for (second = 1; second <= 5; second++){  
        area = PI * radius * radius;  
        System.out.printf("Wn %d 초 경과후 면적 = %8.2f", second, area);  
        radius += 15;  
    }  
}
```

1 초 경과후 면적 =	706.86
2 초 경과후 면적 =	2827.43
3 초 경과후 면적 =	6361.72
4 초 경과후 면적 =	11309.73
5 초 경과후 면적 =	17671.46



Futuristic Innovator

京福大學校  
KYUNGBOK UNIVERSITY

# 주사위

- 두 개의 주사위를 던졌을 때, 눈의 합이 6이 되는 모든 경우의 수를 출력하는 프로그램을 작성하시오



# 주사위

```
public static void main(String[] args) {  
    for(int i=1;i<=6;i++)  
        for(int j=1;j<=6;j++)  
            if(i+j==6)  
                System.out.println(i+" "+j+"="+ (i+j));  
}
```

# 2019년 오늘이 무슨 요일일까?

- 2019년 1월 1일은 화요일이다. 2019년 A월 B일은 무슨 요일일까요?





# 2019년 오늘이 무슨 요일일까?

- 문제 분석

- 입력

- 월(month) – int (1 ~ 12)

- 일(day) – int (1 ~ 28, 29, 30, 31)

- 출력

- 요일명(weekName) – String

- 월요일, 화요일, 수요일, 목요일, 금요일, 토요일,  
일요일

- 계산 방법

# 2019년 오늘이 무슨 요일일까?

- 1년은 12개월이고 1, 3, 5, 7, 8, 10, 12월은 총 31일까지 있고 4, 5, 9, 11월은 30일, 2월은 28일까지 있다.
- 윤년인 경우는 2월이 29일 이다.
- 요일을 구하려면 '1년1월1일 부터~ 내가 구하고자하는 년 월일 까지의 모든 날짜수' 나누기 7을 한 후, 그 나머지가 1이면 월요일 , 2이면 화요일... 이 특징을 이용하면 됨

# 2019년 오늘이 무슨 요일일까?

```
public class Main {  
    static int days[] = {31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31};  
  
    public static void main(String[] args) {  
        Scanner scanner = new Scanner(System.in);  
        int year = 2019, day, month;  
        int count = 0;  
        String weekName;  
  
        if (((year % 4 == 0) && (year % 100 != 0))  
            || year % 400 == 0)  
            days[1] = 29;  
        do {  
            System.out.print("월을 입력하세요 (1 ~ 12) : ");  
            month = scanner.nextInt();  
        } while (month < 1 || month > 12);  
    }  
}
```

# 2019년 오늘이 무슨 요일일까?

```
do {
    System.out.printf("일을 입력하세요(1 ~ %d) : ",
                      days[month-1]);

    day = scanner.nextInt();
} while (day < 1 || day > days[month-1]);

if (month > 1) {
    for (int i = 1; i < month; i++) {
        count += days[i];
    }
}
count += day;
```

# 2019년 오늘이 무슨 요일일까?

```
switch (count % 7) {  
    case 1:  
        weekName = "화요일";  
        break;  
    case 2:  
        weekName = "수요일";  
        break;  
    case 3:  
        weekName = "목요일";  
        break;  
    case 4:  
        weekName = "금요일";  
        break;  
    case 5:  
        weekName = "토요일";  
        break;  
}
```

# 2019년 오늘이 무슨 요일일까?

```
case 6:
    weekName = "일요일";
default:
    weekName = "월요일";
}
System.out.println(year + "년 " + month + "월 " +
                    day + "일은 " + weekName + "입니다.");
}
```

# 오늘이 무슨 요일일까?

- 년, 월, 일을 입력 받아 무슨 요일인지를 출력하는 프로그램을 만들어보자
  - 단, 달력(Calendar) 클래스는 사용하지 않는다

월 화 수 목 금 토 일

# 오늘이 무슨 요일일까?

- 문제 분석

- 입력

- 년(year) – int

- 월(month) – int

- 일(day) – int

- 출력

- 요일(weekday) – String

- 계산 방법

- 1900년 1월 1일 (월요일)부터 시작하여 입력한 날짜 사이의 얼마나 많은 날이 있는지 계산하여 이 날수를 7로 나누어 나머지를 가지고 요일을 판단 함



# 오늘이 무슨 요일일까?

## ■ 문제 분석

### ■ 계산 방법

- 1900부터 시작해서 년(year) - 1까지 윤년이면 366일을 그렇지 않으면 365일을 더함
- 해당 해(year)이면 월(month) - 1까지 날수를 더함
- 해당 월(month)이면 일(day)을 더함
- 계산된 날수를 7로 나누어 나머지를 구함
- 0이면 일요일
- 1이면 월요일
- 2이면 화요일
- 3이면 수요일
- 4이면 목요일
- 5이면 금요일
- 6이면 토요일

# 오늘이 무슨 요일일까?

```
public class Main {  
    static int[] days = {31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31};  
  
    public static void main(String[] args) {  
        Scanner scanner = new Scanner(System.in);  
        int year, day, month;  
        char dayOfWeek = ' ';  
        int totalDay = 0;  
  
        do {  
            System.out.print("연도를 입력하세요 (1900 ~ ? ) : ");  
            year = scanner.nextInt();  
        } while (year < 1900);  
    }  
}
```

# 오늘이 무슨 요일일까?

```
do {  
    System.out.print("월을 입력하세요 (1 ~ 12) : ");  
    month = scanner.nextInt();  
} while (month < 1 || month > 12);  
  
do {  
    System.out.printf("일을 입력하세요 (1 ~ %d) : ",  
                      days[month-1]);  
    day = scanner.nextInt();  
} while (day < 1 || day > days[month-1]);
```



Futuristic Innovator

京福大學校  
KYUNGBOK UNIVERSITY

# 오늘이 무슨 요일일까?

```
for(int i = 1900; i <= year; i++) {  
    if (i < year) {  
        if (((i % 4 == 0) && (i % 100 != 0)) ||  
            i % 400 == 0)  
            totalDay += 366;  
        else  
            totalDay += 365;  
    } else {  
        if (month > 1) {  
            for (int j = 1; j < month; j++) {  
                totalDay += days[j];  
            }  
        }  
    }  
}
```

# 오늘이 무슨 요일일까?

```
totalDay += day;
switch (totalDay % 7) {
    case 0:
        dayOfWeek = '일';
        break;
    case 1:
        dayOfWeek = '월';
        break;
    case 2:
        dayOfWeek = '화';
        break;
    case 3:
        dayOfWeek = '수';
        break;
```



Futuristic Innovator

京福大學校  
KYUNGBOK UNIVERSITY

# 오늘이 무슨 요일일까?

```
case 4:
    dayOfWeek = '목';
    break;
case 5:
    dayOfWeek = '금';
    break;
case 6:
    dayOfWeek = '토';
    break;
}
System.out.println(year + "년 " + month + "월 " +
    day + "일은 " + dayOfWeek + "요일입니다." );
}
```



Futuristic Innovator

京福大學校  
KYUNGBOK UNIVERSITY

# 숫자 맞추기 게임

- 1과 100사이의 값을 반복적으로 입력해서 컴퓨터가 생각한 값을 맞추면 게임이 끝난다 사용자가 값을 입력하면 컴퓨터는 자신이 생각한 값과 비교해서 결과를 알려준다 사용자가 컴퓨터가 생각한 숫자를 맞추면 게임이 끝나고 몇 번 만에 숫자를 맞췄는지 알려준다

# 숫자 맞추기 게임

```
public class Main {  
    public static void main(String[] args) throws IOException {  
        Random random = new Random();  
        Scanner keyboard = new Scanner(System.in);  
  
        int num = random.nextInt(100) + 1;  
        int guess;  
        int count = 0;  
        do {  
            count++;  
            do {  
                System.out.print("당신이 생각하는 숫자 입력 : ");  
                guess = keyboard.nextInt();  
                if (guess >= 1 && guess <= 100)  
                    break;  
                else {  
                    System.err.println("입력 오류");  
                    System.in.read();  
                }  
            } while (true);  
        }
```



# 숫자 맞추기 게임

```
if (guess > num) {  
    System.out.printf("%d보다 작은 수를 입력하세요\n", guess);  
} else if (guess < num) {  
    System.out.printf("%d보다 큰 수를 입력하세요\n", guess);  
} else {  
    System.out.printf("%d번에 맞았습니다\n\n", count);  
    count = 0;  
}  
} while (true);  
}
```

# 통신 요금

- 동호는 T 통신사의 새 핸드폰을 구입했다. T 통신사는 동호에게 다음 두 가지 요금제 중 하나를 선택하라고 했다.
- Type1 요금제
  - Type1 요금제는 30초마다 10원씩 청구된다. 이 말은 만약 29초 또는 그 보다 적은 시간 통화를 했으면 10원이 청구된다. 만약 30초부터 59초 사이로 통화를 했으면 20원이 청구된다.
- Type2 요금제
  - Type2 요금제는 60초마다 15원씩 청구된다. 이 말은 만약 59초 또는 그 보다 적은 시간 통화를 했으면 15원이 청구된다. 만약 60초부터 119초 사이로 통화를 했으면 30원이 청구된다.

# 통신 요금

- 동호가 지난 달에 T 통신사를 이용할 때 통화 시간 목록이 주어지면 어느 요금제를 사용하는 것이 더 저렴한지 출력하는 프로그램을 작성하시오.