



FOR문과 소인수 분해 알고리즘

> For문이란?

- FOR 문 : 반복문

- While 문과는 달리 초기식, 조건식, 증감식 모두 포함하고 있는 반복문

그래서 while 문보다는 좀 더 간결하게 표현 할 수 있다.

```
For ( 1초가식; 2조건식; 4증감식 ) {  
    3조건식의 결과가 참인 동안 반복적으로 실행하고자 하는 명령문;  
}
```

> For문이란?

- FOR 문 : 예제

```
int num = 0;
for ( int i = 0; i < 5; i ++ ) {
    num ++;
    System.out.println ( num );
}
```

출력) 1

2

3

4

5

> For문이란?

- FOR 문 :

//초기화식 생략

```
int i = 0;
```

```
for ( ; i < 10; i ++ ) {
```

```
    printf( "%d\n", i );
```

```
}
```

//조건식 생략(무한루프)

```
for ( int i = 0; ; i ++ ) {
```

```
    printf( "%d\n", i );
```

```
}
```

> For문이란?

- FOR 문 :

//증감식 생략(무한루프)

```
for ( int i = 0 ; i < 10 ; ) {  
    printf ( "%d\n", i );  
}
```

//실행문이 한라인일 경우 {} 대괄호 생략 가능

```
for ( int i = 0 ; i < 10 ; i ++ )  
    printf ( "%d\n", i );
```

> For문이란?

- FOR 문 : 범위 기반 for문

기본 문법

```
for ( 데이터 타입 elem : 데이터 리스트 ) {  
  
}
```

C++ 11에서 도입됨

> For문이란?

- FOR 문 : 범위 기반 for문 예제
- 기본 for문으로 작성

```
int arr [5] = {1, 2, 3, 4, 5};  
for ( int i = 0 ; i < 5 ; i ++ ) {  
    cout << arr[i] << endl;  
}
```

출력) 1

2

3

4

5

> For문이란?

- FOR 문 : 범위 기반 for문 예제
- 범위 기반 for문 사용시

```
int arr [5] = {1, 2, 3, 4, 5};  
for ( int i : arr) {  
    cout << i << endl;  
}
```

출력) 1

2

3

4

5

> For문이란?

- FOR 문 : 범위 기반 for문의 단점

배열에 있는 인자를 새로운 변수에 복사를 하기 때문에

배열의 원래 값을 변경하지 못 함

복사비용이 발생함

> For문이란?

- FOR 문 : 범위 기반 for문 단점
- 범위기반 반복문 내부에서 배열의 요소를 변경할 수 없다.

```
int arr [5] = {1, 2, 3, 4, 5};
```

```
for ( int i : arr ) {
```

```
    i = 10;
```

```
}
```

```
for ( int i : arr) {
```

```
    cout << i << endl;
```

```
}
```

출력) 1

2

3

4

5

> For문이란?

- FOR 문 : 범위 기반 for문 단점을 보완하는 방법
- & 레퍼런스 사용

```
int arr [5] = {1, 2, 3, 4, 5};  
for ( int& i : arr ) {  
    i = 10;  
}  
for ( int& i : arr) {  
    cout << i << endl;  
}
```

출력) 10

10

10

10

10

> For문이란?

- FOR 문 : 범위 기반 for문 단점을 보완하는 방법
- 변경이 일어나지 않아야 하는 경우 `const`와 `&`같이 사용

```
int arr [5] = {1, 2, 3, 4, 5};  
for ( const int& i : arr ) {  
    i = 10; //불가능  
}  
for ( const int& i : arr) {  
    cout << i << endl;  
}
```

> 소인수분해란?

- 소인수분해 :

어떤 정수를 소수인 인수들로 분해하는 것

EX) $25 > 1 * 25 > 1 * 5 * 5$

$70 > 2 * 5 * 7$

> 소인수분해란?

- 소수 :

- 1과 자기 자신만을 약수로 가지는 수

EX) 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31.....

> 소인수분해란?

- 소인수 분해 알고리즘 : N을 모든 숫자와 나누기

```
int k = 2;
```

```
While ( num != 1) {  
    if ( num & k == 0 ) {  
        System.out.println( k + " ");  
        num /= k;  
    } else {  
        k++;  
    }  
}
```

> 소인수분해란?

- 소인수 분해 알고리즘 : 제곱근을 이용

```
int k = 2;
double sqrtNum = Math.sqrt(num);

While ( k <= sqrtNum && num != 1) {
    if ( num2 % k == 0 ) {
        System.out.println( k + " ");
        num2 /= k;
    } else {
        k++;
    }
}

if ( num2 > 1 ) {
    System.out.print( k + " ");
}
```




끝!