

## 예제 1

Scanner 클래스로 사용자로부터 양의 정수를 입력받는 프로그램입니다.

입력을 계속 받되, -1이 입력되는 순간 프로그램을 종료합니다.

입력받은 수는 모두 **ArrayList**에 저장하고, 프로그램이 종료되면, 저장된 **List** 에서 가장 큰 수를 출력하려고 합니다.

**ArrayList<Integer>**

단, 가장 큰 수를 선택할 때는 함수를 사용하지 않고, 반복문을 사용해주세요.

출력 예시

```
양의 정수를 입력해주세요 >>> 3
양의 정수를 입력해주세요 >>> 5
양의 정수를 입력해주세요 >>> 6
양의 정수를 입력해주세요 >>> -1
프로그램을 종료합니다.
저장된 리스트 : [3, 5, 6, -1]
가장 큰 수 : 6
```

## 예제 2

이름과 학점(4.5만점)을 5개를 입력받아 해시맵에 저장하고, 장학생 선발 기준을 입력 받아 장학생 명단을 출력하라.

HashMap<String, Double>

출력 예시

```
미래 장학금 관리 시스템입니다.
이름 >>학생1
학점 >>3.4
=====
이름 >>학생2
학점 >>3.9
=====
이름 >>학생3
학점 >>4.5
=====
이름 >>학생4
학점 >>2.8
=====
이름 >>학생5
학점 >>4.23
=====
정보 입력 종료
장학생 선발 학점 기준 입력 >> 3.7
장학생 명단 장학생 명단 : 학생2 학생3 학생5
```

// 여기서 stu는 학생의 이름과 학점을 모아둔 hashmap

```
Set<String> key =stu.keySet(); // 학생의 키를 Set 형식으로 모아둔 거
Iterator<String> it = key.iterator(); // key의 index를 반복 가능하도록 한 거
```

```
while(it.hasNext()){
    String name = it.next();
    double grade = stu.get(name);
```

// 학점 기준보다 큰 경우를 찾아 출력하는 코드를 작성하면 됩니다.

```
}
```

🍌 깨알 정보....

iterator란 자바의 컬렉션(Collection)에 저장되어 있는 요소들을 순회하는 인터페이스

### 예제 3 - HashMap

고객의 이름과 포인트 점수를 관리하는 프로그램을 HashMap 을 이용하여 작성하려 합니다.

입출력 예시는 다음과 같습니다.

고객의 이름이 이미 존재하면, 포인트가 누적되는 방식입니다.

===== 출력 예시 =====

```
고객 포인트 관리 프로그램입니다.
```

```
고객 이름 >>홍길동
```

```
홍길동님 포인트 입력 >>30
```

```
{홍길동=30}
```

```
고객 이름 >>김삼순
```

```
김삼순님 포인트 입력 >>60
```

```
{홍길동=30, 김삼순=60}
```

```
고객 이름 >>이순신
```

```
이순신님 포인트 입력 >>45
```

```
{홍길동=30, 김삼순=60, 이순신=45}
```

```
고객 이름 >>황진이
```

```
황진이님 포인트 입력 >>15
```

```
{홍길동=30, 김삼순=60, 황진이=15, 이순신=45}
```

```
고객 이름 >>김삼순
```

```
김삼순님 포인트 입력 >>50
```

```
{홍길동=30, 김삼순=110, 황진이=15, 이순신=45}
```

```
고객 이름 >>종료
```

```
{홍길동=30, 김삼순=110, 황진이=15, 이순신=45}
```

기본 toString () 메서드를 사용 하여 출력했습니다.

고객 정보를 입력할 때마다 나왔던 정보출력도 마찬가지입니다.

위 출력 예시는 출력시 toString() 기본 메서드를 사용한 예시입니다.

가능하다면 예제2의 Set<String> 과 Iterator<String> 을 이용하여 다음과 같이 출력하는 연습도 같이 해보세요

```
고객 포인트 관리 프로그램입니다.
```

```
고객 이름 >>홍길동
```

```
홍길동님 포인트 입력 >>30
```

```
[홍길동, 30]
```

```
고객 이름 >>김삼순
```

```
김삼순님 포인트 입력 >>60
```

```
[홍길동, 30] [김삼순, 60]
```

```
고객 이름 >>이순신
```

```
이순신님 포인트 입력 >>45
```

```
[홍길동, 30] [김삼순, 60] [이순신, 45]
```

```
고객 이름 >>황진이
```

```
황진이님 포인트 입력 >>15
```

```
[홍길동, 30] [김삼순, 60] [황진이, 15] [이순신, 45]
```

```
고객 이름 >>김삼순
```

```
김삼순님 포인트 입력 >>50
```

```
[홍길동, 30] [김삼순, 110] [황진이, 15] [이순신, 45]
```

```
고객 이름 >>종료
```

```
고객 포인트 관리 정보입니다.
```

```
[홍길동, 30]  
[김삼순, 110]  
[황진이, 15]  
[이순신, 45]
```

## 예제 4 - HashSet

로또 번호를 중복없이 생성하는 프로그램입니다. 사용자로 부터 1) 로또 생성, 2) 종료 를 입력받아

1)을 입력받을 시 6줄의 로또를 생성하고, 2)를 입력받을 시 프로그램을 종료합니다.

로또 번호는 다음과 같이 생성되고, 한줄의 로또 번호는 HashSet을 사용하십시오.

1. 한 줄 내에서는 중복이 없습니다.
2. 총 6줄이 생성됩니다.
3. 줄과 줄 끼리는 중복이 허용됩니다.

===== 출력예시 =====

```
로또 생성 프로그램을 시작합니다.  
[1] > lotto 생성, [2] > 종료  
번호를 선택해주세요 >> 1  
지금부터 로또를 생성합니다.  
추첨된 로또번호는 다음과 같습니다.  
[20, 4, 37, 22, 44, 31]  
[35, 36, 7, 44, 14, 31]  
[1, 20, 10, 27, 28, 31]  
[1, 17, 20, 4, 5, 26]  
[36, 5, 39, 41, 10, 14]  
[3, 5, 37, 41, 42, 14]  
  
[1] > lotto 생성, [2] > 종료  
번호를 선택해주세요 >> 1  
지금부터 로또를 생성합니다.  
추첨된 로또번호는 다음과 같습니다.  
[18, 34, 36, 23, 25, 44]  
[33, 2, 18, 35, 23, 31]  
[4, 38, 41, 42, 27, 29]  
[36, 20, 5, 38, 24, 41]  
[17, 18, 3, 20, 9, 28]  
[16, 1, 2, 19, 5, 8]  
  
[1] > lotto 생성, [2] > 종료  
번호를 선택해주세요 >> 2  
프로그램을 종료합니다.
```