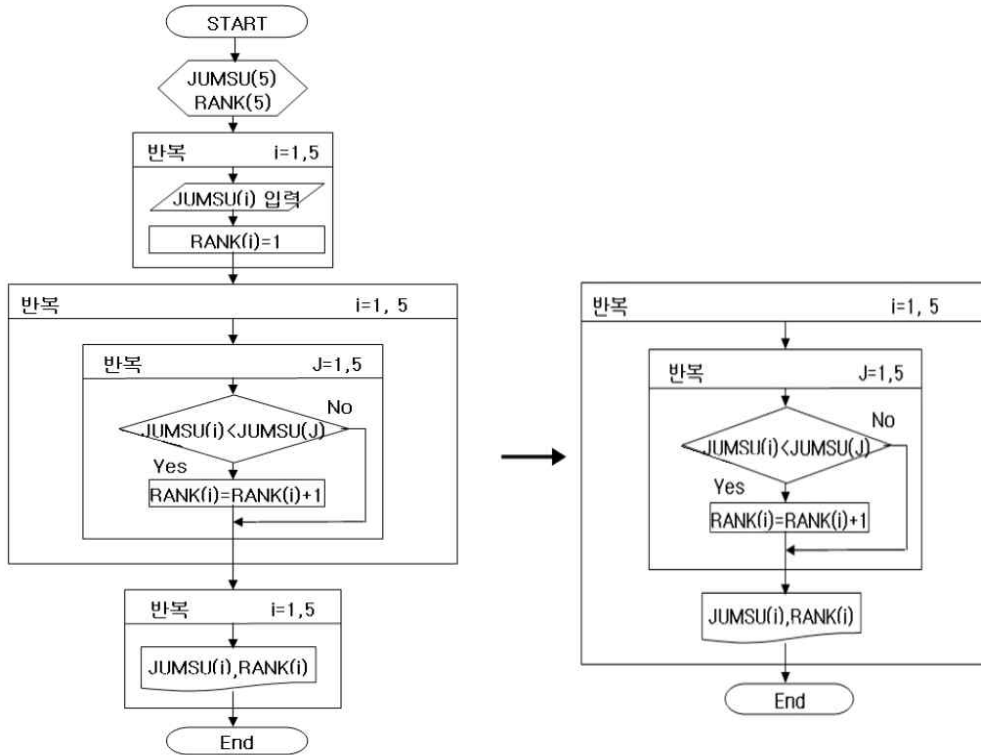


[고급유형 01] 석차

1) 문제 : 5명 학생의 성적에 따른 석차를 구하는 순서도를 작성하시오.

2) 조건처리 :

- JUMSU(5) : 점수를 저장할 배열, RANK(5) : 석차를 저장할 배열, i, j: 인덱스 변수



i=1,2,3,4,5

j=1,2,3,4,5

JUMSU(i) < JUMSU(j)

JUMSU(1)

JUMSU(2)

JUMSU(3)

JUMSU(4)

JUMSU(5)

```

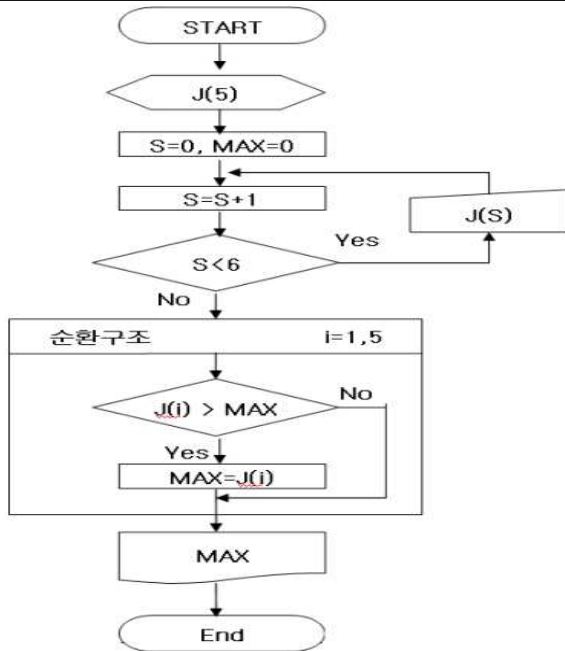
2
3 import java.util.Scanner;
4 public class 석차 {
5     public static void main(String[] args) {
6         int jumsu[]=new int[5];
7         int rank[]=new int[5];
8         int i,j;
9         Scanner sc=new Scanner(System.in);
10        System.out.println("5개의 성적을 입력하세요");
11        for(i=0;i<5;i++) {
12            jumsu[i]=sc.nextInt();
13            rank[i]=1;
14        }
15        for(i=0;i<5;i++) {
16            for(j=0;j<5;j++) {
17                if(jumsu[i]< jumsu[j]) rank[i]=rank[i]+1;
18            }
19        }
20        for(i=0;i<5;i++) {
21            System.out.println(jumsu[i]+"," +rank[i]);
22        }
23    }
24 }
25
26 }
    
```

[고급유형 02] 최대값과 최소값

1) 문제 : 5개의 성적을 입력받아 배열에 저장한 후 저장된 자료 중 최대값을 찾는 순서도를 작성하시오.

2) 처리조건 :

- J(5) : 입력 받은 점수를 저장할 배열
- S : 배열의 위치, MAX : 최대값 저장, i :인덱스 변수



i=1,2,3,4,5	MAX=0
-------------	-------

J(i)

 $J(1)$ 

J(2)

J(3)

J(4)

J(5)

---

[illegible][illegible][illegible]

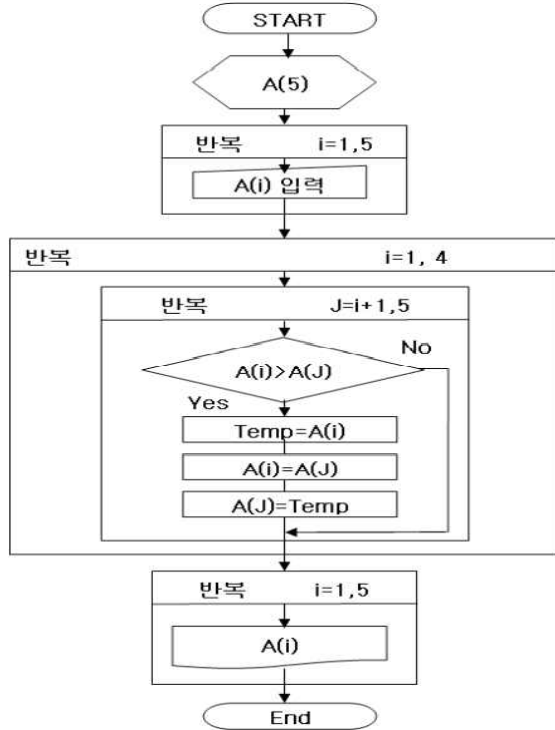
---

[고급유형 03] : 선택정렬(selection sort)

1) 문제 : 5명의 학생 성적을 입력 받아 배열에 저장한 후 저장된 자료를 오름차순으로 정렬하는 순서도를 작성하시오. (선택정렬 이용)

2) 조건처리 :

- A(5) : 정렬할 숫자가 저장될 배열, i, j : 인덱스 변수, Temp : 자료 교환에 필요한 임시 변수



i=1                      i=2                      i=3                      i=4  
j=2,3,4,5              j=3,4,5                  j=4,5                  j=5

if(A(i) > A(j)) 이면 교체법

A(1)

A(2)

A(3)

A(4)

A(5)

A(1)	A(2)	A(3)	A(4)	A(5)
10	30	15	20	70



A(1)	A(2)	A(3)	A(4)	A(5)
10	30	15	20	70

※선택소트 암기코드

1) A(i) 와 A(j) 처럼 i, j 다른 첨자를 사용함

2) > 일때 교체한다면 오름차순

< 일때 교체한다면 내림차순

3) j=i+1 부터 5까지 앞쪽에 변화가 있음

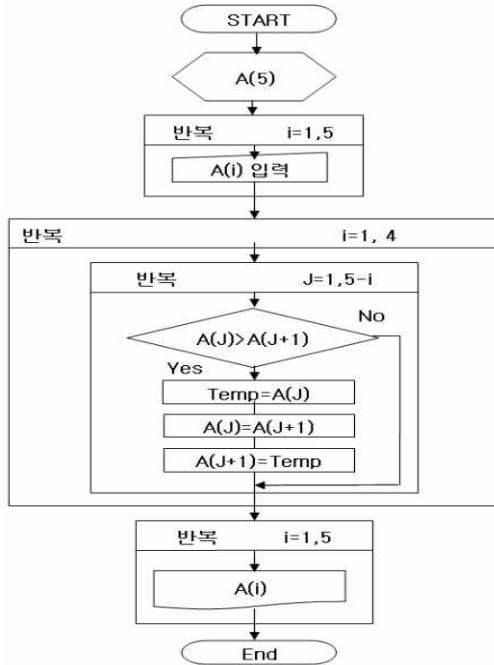
```
1 package 알골;
2
3 public class 선택소트 {
4     public static void main(String[] args) {
5         int a[]={10,30,15,20,70};
6         int i,j,temp;
7         for(i=0;i<4;i++) {
8             for(j=i+1;j<5;j++) {
9                 if(a[i]< a[j]) {
10                     temp=a[i];
11                     a[i]=a[j];
12                     a[j]=temp;
13                 }
14             }
15         }
16
17         for(i=0;i<5;i++) {
18             System.out.print(a[i]+" ");
19         }
20     }
21 }
22
```

[고급유형 04] : 버블정렬(bubble sort)

1) 문제 : 5명의 학생 성적을 입력 받아 배열에 저장한 후 저장된 자료를 오름차순으로 정렬하는 순서도를 작성하시오. (버블정렬 이용)

2) 조건처리 :

- A(5) : 정렬할 숫자가 저장될 배열, i, J : 인덱스 변수, Temp : 자료 교환에 필요한 임시 변수



i=1	i=2	i=3	i=4
J=1,2,3,4	J=1,2,3	J=1,2	J=1

if(A(J) > A(J+1)) 교체법

A(1)

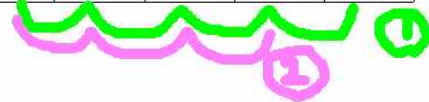
A(2)

A(3)

A(4)

A(5)

A(1)	A(2)	A(3)	A(4)	A(5)
10	30	15	20	70



A(1)	A(2)	A(3)	A(4)	A(5)
10	30	15	20	70

※버블소트 암기코드

1) A(J) 와 A(J+1) 처럼 첨자가 인접하다

2) > 일때 교체한다면 오름차순

< 일때 교체한다면 내림차순

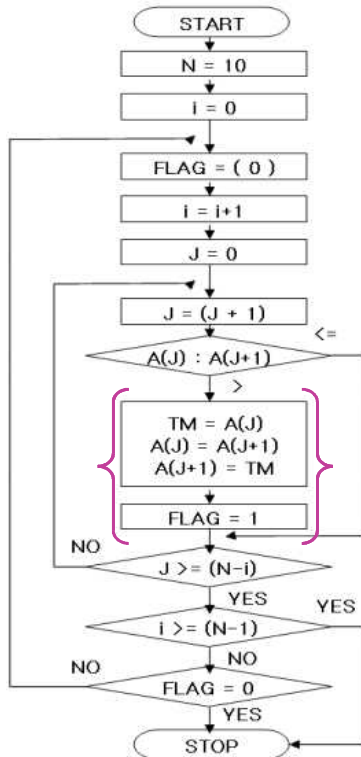
3) 1부터 5-i 까지 뒤쪽에 변화가 있음

[고급유형 04] : 버블정렬(bubble sort) - 2

1) 문제 : 배열 A(10)에 기억된 10개의 수치 데이터에 대하여 버블정렬(Bubble sort)을 이용하여 오름차순으로 정렬하는 순서도이다.

## 2) 조건처리 :

- N : 정렬하고자 하는 수치 데이터의 갯수, i : 정렬의 회전수를 계산하기 위한 변수, J : 배열의 첨자 등을 위한 변수, TM : 주 변수 간의 값을 서로 바꾸기 위한 변수, FLAG : 임의의 회전 작업시 데이터의 교환이 발생하지 않을 경우 비교가 반복되는 것을 방지하기 위한 변수



i=1	i=2	i=3	i=4
J=1,2,3,4	J=1,2,3	J=1,2	J=1

if(A(J) > A(J+1)) 교체법  
A(1)  
A(2)  
A(3)  
A(4)  
A(5)

A(1)	A(2)	A(3)	A(4)	A(5)	FLAG=0
10	20	30	40	50	

A(1)	A(2)	A(3)	A(4)	A(5)	FLAG=0
10	30	15	20	70	

### ※개선된 버블소트 암기코드

1) FLAG=0 초기값 할당한 경우 조건은 반드시 FLAG=0 이  
나고 판단한다. 즉, if(flag==0)

2) FLAG=1을 대입했다는 의미는 한번이라도 교체작업을 했  
다는 뜻으로 해석되어 계속 비교작업을 해야 한다.

3) FLAG=1을 대입하지 않았다는 의미는 FLAG에 초기값으로 0이 할당되어 있다는 의미이므로 한번도 교체작업을 하지 않았기 때문에 이미 오름차순으로 정렬되어 있다는 뜻으로 해석되어 더 이상 비교작업을 하지 않고 빠져나와서 끝낸다

4)  $A(J)$  와  $A(J+1)$  처럼 첨자가 인접하다

5) > 일때 교체한다면 오름차순

< 일때 교체한다면 내림차순

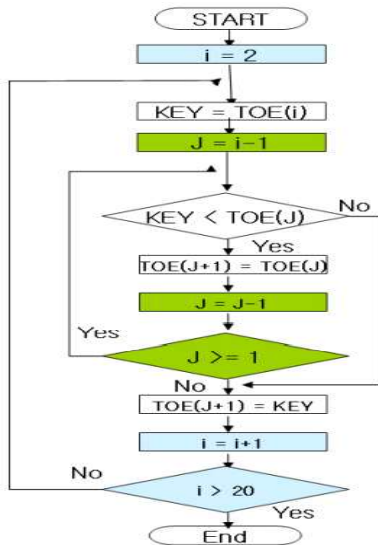
6) 1부터 5-i 까지 뒤쪽에 변화가 있음

[고급유형 05] : 삽입 정렬(insertion sort)

1) 문제 : 삽입정렬(insertion sort) 기법을 사용하여 전산실 직원 20명의 토익 점수가 저장되어 있는 배열 TOE(20)의 데이터를 오름차순으로 정렬하고자 한다.

2) 처리조건

- KEY : 비교 기준이 되는 키 값 저장 변수, i, J : 인덱스 변수



TOE(1)	TOE(2)	TOE(3)	TOE(4)	TOE(5)
60	10	5	40	7

i=2 J=1,0	i=3 J=2,1,0	i=4 J=3,2,1,0	i=5 J=4,3,2,1,0
--------------	----------------	------------------	--------------------

KEY :

※삽입소트 암기코드

1) key=TOE(i) 배열의 두번째 값을 key로 저장

2) 배열에서 앞쪽으로 전진하는 첨자는 J 이며,

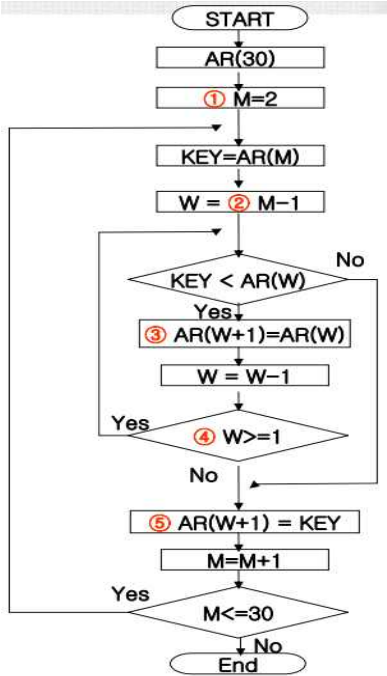
key 보다 **배열앞쪽**이 크면, 배열앞쪽을 배열뒤쪽으로 저장하면서 전진하며 반복한다.

3) 배열앞쪽이 더 없거나, **key**가 배열앞쪽보다 크면, key를 배열에 저장한다

4) 1)부터 반복 즉, 배열의 세번째 값을 key에 저장

[고급유형 05] : 삽입 정렬(insertion sort) 연습하세요

1) 문제 : 삽입정렬(insertion sort)

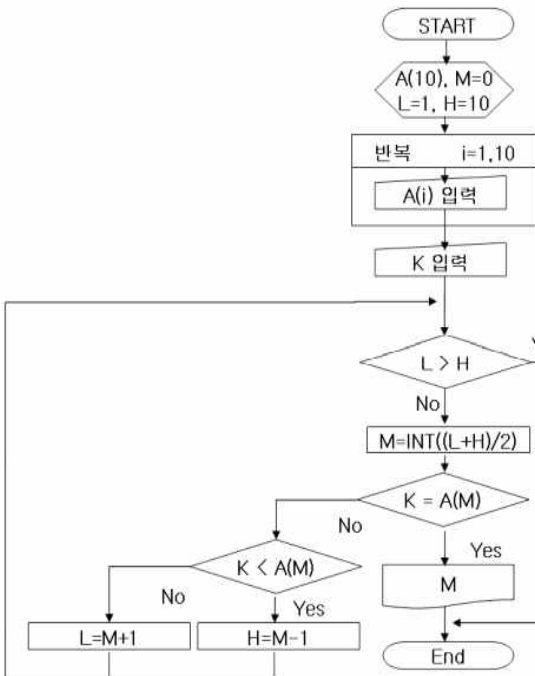


[고급유형 06] : 이분 검색

1) 문제 : 입력 받은 값이 배열의 몇 번째에 기억되어 있는지 알아보기 위해 출력하는 순서도를 작성하시오.

(단, 배열에 저장된 값은 오름차순 정렬된 10개 데이터이며, 이분검색을 이용하라.)

2) 조건처리 : A(10) : 숫자가 저장된 배열, K : 검색할 값, L : 배열첨자이며 시작위치, H : 배열첨자이며 마지막 위치, M : 배열첨자이며 중간위치, i : 인덱스 변수



A(1)	A(2)	A(3)	A(4)	A(5)	A(6)	A(7)	A(8)	A(9)	A(10)
10	20	30	40	50	60	70	80	90	100

K : 30

L : 1

H : 10

M :

K(키)가 작으면 앞에 위치하니 H=(M-1)

K(키)가 크면 뒤에 위치하니 L=(M+1)



--	--