이터레이터 지식

1. sort()

```
template <class RandomAccessIterator, class Compare>
  void sort (RandomAccessIterator first, RandomAccessIterator last, Compare comp;
```

1-1 first, last

1. 처음부터 끝까지 정렬

```
    sort(arr, arr + Size);
    sort(V.begin(), V.end(), compare)
```

2. 특정 구간까지만 정렬

```
    sort(arr + N, arr + M);
    sort(V.begin() + N, V.begin() + M, compare)
```

- 따지면 V[N]부터 V[M]까지
- M-N 개 정렬

1-2 compare는 그럼 뭐임?

1. 오름차순(default)
   less<자료형>

```
//1-1 처음부터 끝까지 정렬을 기준으로
    sort(V.begin(), V.end(), less<자료형>()); //괄호까지 꼭!!
    sort(V.begin(), V.end());
```

2. 내림차순
   greater<자료형>

```
    sort(V.begin(), V.end(), greater<자료형>()); //괄호까지 꼭!!
```

3. 내 ㅈ대로 할꺼야 -> 왓?
   pair<int,int>,구조체 사용하는경우

```
typedef struct pair{
    int first;
    int second;
}pair;
```

```
typedef struct triple{
    int one;
    int two;
    int three;
}triple;
```

엥??? 이런것도 배열로 만들수 있나? -> 맞다 게이야...

```
typedef struct pair{
    int first;
    int second;
}pair;

pair arr[10] = {
    {1,10}, {2,20}, ... {10,100}
    };
혹은..
for(int i = 0 ; i <= 10 ; i++){
    int a,b;
    cin >> a >> b;
    arr[i] = {a,b};
}
벡터는
vector<pair> V;
V.push_back({1,100});
```

```
typedef struct triple{
    int first;
    int second;
}triple;

pair arr[10] = {
    {1,10, 500},
    {2,20, 1000},
    ...
    {10, 100,5000}
    };

혹은..
for(int i = 0 ; i <= 10 ; i++){
    int a,b,c;
    cin >> a >> b >> c;
    arr[i] = {a,b,c};
}
vector<triple> V;
V.push_back({1,100, 500});
```