4주차

배열이란?

- 요소의 집합
- 기본값은 쓰레기값

```cpp
#include <iostream>
using namespace::std;

int main() {
    ios::sync_with_stdio(false);
    cin.tie(nullptr);

    int arr1[5];

    int arr2[4] = {0, 0, -1, 1};

    int arr3[] = {0, 0, 1, -1};

    char arr4[] = {'a', 'b'};

    return 0;
}

```

백터란?

- 배열의 현대판
- 가변적
- 기본값은 0, false

```cpp
#include <iostream>
#include <vector>
using namespace::std;

int main() {
    ios::sync_with_stdio(false);
    cin.tie(nullptr);

    vector<int> vec1(4);

    vector<int> vec2 = {0, 0, -1, 1};

    vector<char> vec3 = {'a', 'b', 'c'};

    return 0;
}
```

반복자(iterator)란?

- 컨테이너 탐색에 사용되는 객체

```cpp
#include <iostream>
#include <vector>
using namespace::std;

int main() {
    ios::sync_with_stdio(false);
    cin.tie(nullptr);

    vector<int> list = {0, 1, 2, 3};

    vector<int>::iterator it = list.begin();
    auto it2 = list.begin();

    while (it != list.end()) {
        cout << *it << '\n';
        it++;
    }


    return 0;
}
```

- vec[i] : 백터의 i번째 원소를 반환
- vec.push_back(a) : 백터 뒤쪽에 a를 넣는다
- vec.pop_back() : 백터 마지막 요소 삭제
- vec.erase(pos) : pos 위치의 원소 삭제
- vec.assign(n, value) : 기존 데이터 삭제 후 n개의 value로 채움
- vec.size() : 백터의 사이즈 반환
- vec.empty() : 백터가 비었으면 true, 아니면 false
- vec.resize(n) : 백터의 사이즈를 수정
- vec.front() : 백터의 첫 요소 반환
- vec.back() : 백터의 마지막 요소 반환
- vec.insert(pos, value) : pos 위치에 value 삽입
- vec.clear() : 백터 클리어

- sort(vec.begin(), vec.end()) : 백터를 올림차순으로 정렬
- reverse(vec.begin(), vec.end()) : 백터를 뒤집기
- find(vec.begin(), vec.end(), value) : 특정 값 찾기 (중복된다면 앞에 있는 값의 인덱스 반환)
- count(vec.begin(), vec.end(), value) : 특정 값의 개수 세기