프로그래밍 언어 – STL (1)

최백준 choi@startlink.io

STL

STL

C++

- Standard Template Library
- 알고리즘
- 컨테이너
- 함수
- 이터레이터
- 로 이루어져 있다.

pair

pair

STL

- pair를 사용하면 두 자료형 T1과 T2를 묶을 수 있다
- 항상 두 개를 묶는다.
- 첫 번째 자료는 first
- 두 번째 자료는 second로 접근할 수 있다.

pair

STL

- #include <utility>
- 에 있는데,
- algorithm, vector와 같은 헤더파일에서 이미 include 하고 있기 때문에, 따로 include 하는 경우는 없다.
- make_pair를 이용하거나, 생성자를 이용해서 만들 수 있다.

pair

STL


```
pair<int, int> p1;
cout << p1.first << ' ' << p1.second << '\n';

p1 = make_pair(10, 20);
cout << p1.first << ' ' << p1.second << '\n';

p1 = pair<int, int>(30, 40);
cout << p1.first << ' ' << p1.second << '\n';

pair<int, int> p2(50, 60);
cout << p2.first << ' ' << p2.second << '\n';
```

pair

STL


```
0 0
10 20
30 40
50 60
```

pair

STL


```
pair<pair<int,int>, pair<int,int>>  p =
make_pair(make_pair(10,20), make_pair(30,40));


cout << p.first.first << ' ' << p.first.second << ' ';
cout << p.second.first << ' ' << p.second.second << '\n';
```

```
10 20 30 40
```

tuple

tuple

STL

- tuple은 pair와 같지만 여러 개를 묶을 수 있다
- .first, .second, .third, .fourth …. 가 아니고
- get을 이용해서 인덱스로 접근해야 한다
- tuple은 #include <tuple>에 정의되어 있다

tuple

STL


```
tuple<int, int, int> t1 = make_tuple(1, 2, 3);

cout << get<0>(t1) << ' ';
cout << get<1>(t1) << ' ';
cout << get<2>(t1) << '\n';

/*
for (int i=0; i<3; i++) {
    cout << get<i>(t1) << '\n';
}
*/
```

tuple

STL

- get<> 사이에 변수를 넣을 수는 없다

tie

tie

STL


```
auto t = make_tuple(10, 20, 30);

int x = get<0>(t);
int y = get<1>(t);
int z = get<2>(t);

cout << x << ' ' << y << ' ' << z << '\n';

x = y = z = 0;

tie(x,y,z) = t;

cout << x << ' ' << y << ' ' << z << '\n';
```

tie

STL

- tie는 pair에도 사용할 수 있다
- 변수값을 무시해야 하는 경우에는 ignore를 사용한다

```
auto t = make_tuple(1, 2, 3);

int x, y;
tie(x, y, ignore) = t;

cout << x << ' ' << y << '\n';
```

vector

vector

STL

- vector는 배열이다
- 길이를 변경할 수 있는 배열이다
- #include <vector>

vector

STL


```cpp
#include <iostream>
#include <vector>
using namespace std;
int main() {
    vector<int> v1;
    vector<int> v2(10);
    vector<int> v3(15, 1);
    vector<int> v4 = {1, 2, 3, 4, 5};

    return 0;
}
```

vector

STL


```cpp
#include <iostream>
#include <vector>
using namespace std;
int main() {
    vector<pair<int,int>> v5;
    vector<pair<int,int>> v6 = {{1, 2}, {3, 4}};
    vector<vector<int>> v7;

    int n = 10, m = 20;
    vector<vector<int>> v8(n, vector<int>(m));

    return 0;
}
```

vector

STL

```
vector<int> a = {1, 2, 3, 4, 5};

a.push_back(6); // [1, 2, 3, 4, 5, 6]
a.push_back(7); // [1, 2, 3, 4, 5, 6, 7]

a.pop_back(); // [1, 2, 3, 4, 5, 6]
a.pop_back(); // [1, 2, 3, 4, 5]
a.pop_back(); // [1, 2, 3, 4]

a.clear(); // []

a.resize(5); // [0, 0, 0, 0, 0]
```

vector

STL


```
a.clear(); // []

a.push_back(1); // [1]
a.push_back(2); // [1, 2]

a.resize(5); // [1, 2, 0, 0, 0]
a.resize(3); // [1, 2, 0]
a.clear(); // []
```

vector

STL


```
vector<int> a = {1, 2, 3, 4};
cout << "size = " << a.size() << '\n';

a.push_back(5);
cout << "size = " << a.size() << '\n';
cout << "empty = " << a.empty() << '\n';

a.clear();
cout << "size = " << a.size() << '\n';
cout << "empty = " << a.empty() << '\n';
```

vector

STL


```
vector<int> a = {1, 2, 3};
cout << "front = " << a.front() << ‘\n’;
cout << "a[1] = " << a[1] << ‘\n’;
cout << "back = " << a.back() << ‘\n’;

a.push_back(4);

for (int i=0; i<a.size(); i++) {
    cout << a[i] << ' ';
}
cout << '\n';
```

vector

STL


```
front = 1
a[1] = 2
back = 3
1 2 3 4
```

vector

STL

```
vector<int> a = {1, 2, 3, 4, 5};

for (int i=0; i<a.size(); i++) {
    cout << a[i] << ' ';
}
cout << '\n';

for (int &x : a) {
    cout << x << ' ';
}
cout << '\n';
```

vector

STL

```
vector<int> a = {1, 2, 3, 4, 5};

for (vector<int>::iterator it = a.begin(); it != a.end(); ++it) {
    cout << *it << ' ';
}
cout << '\n';

for (auto it = a.begin(); it != a.end(); ++it) {
    cout << "a[" << (it - a.begin()) << "] = " << *it << '\n';
}
```

vector

STL

```
vector<pair<int,int>> a;
a.emplace_back(1, 2);
a.push_back({3, 4});
a.push_back(make_pair(5,6));

for (auto &x : a) {
    cout << x.first << ' ' <<  x.second << '\n';
}

for (auto it = a.begin(); it != a.end(); ++it) {
    cout << it->first << ' ' << it->second << '\n';
}
```

vector

STL

```
vector<int> a = {1, 2, 3};
print(a);

auto it = a.begin();
a.insert(it, 4); print(a);

it = a.begin() + 1;
a.insert(it, 5, 0); print(a);

it = a.begin() + 2;
vector<int> b = {10, 20};
a.insert(it, b.begin(), b.end()); print(a);
```

vector

STL


```
void print(vector<int> &a) {
    for (int x : a) {
        cout << x << ' ';
    }
    cout << '\n';
}
```

vector

STL

```
1 2 3
4 1 2 3
4 0 0 0 0 0 1 2 3
4 0 10 20 0 0 0 0 1 2 3
```

vector

STL


```
vector<int> a = {1, 2, 3, 4, 5};
print(a);

a.erase(a.begin()+2);
print(a);

a.erase(a.begin()+1, a.begin()+3);
print(a);
```

```
1 2 3 4 5
1 2 4 5
1 5
```

deque

deque

STL


```
deque<int> d;

d.push_back(1); print(d);

d.push_front(2); print(d);

d.push_back(3); print(d);

d.pop_back(); print(d);

d.pop_front(); print(d);
```

deque

STL


```
1
2  1
2  1  3
2  1
1
```

덱

https://www.acmicpc.net/problem/10866

- https://gist.github.com/Baekjoon/3e348d29fa84fc256782

list

list

STL


```
list<int> l = {2, 1, -5, 4, -3, 6, -7}; print(l);

l.sort(); print(l);

l.sort(greater<int>()); print(l);

l.sort([](int &u, int &v) {
    return abs(u) < abs(v);
});
print(l);

l.reverse(); print(l);
```

list

STL


```
2 1 -5 4 -3 6 -7
-7 -5 -3 1 2 4 6
6 4 2 1 -3 -5 -7
1 2 -3 4 -5 6 -7
-7 6 -5 4 -3 2 1
```

풍선 터뜨리기

https://www.acmicpc.net/problem/2346

- https://gist.github.com/Baekjoon/34f32467072792589cc1
- https://gist.github.com/Baekjoon/d4483e012a28248ed5cd

에디터

https://www.acmicpc.net/problem/1406

- https://gist.github.com/Baekjoon/a2028fb9878c7bf82e35

set

set

STL


```
set<int> s1;
set<int> s2 = {1, 2, 3, 4, 5};
set<int> s3 = {1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 3, 3, 3};

cout << "s1.size() = " << s1.size() << '\n';
cout << "s2.size() = " << s2.size() << '\n';
cout << "s3.size() = " << s3.size() << '\n';

set<int, greater<int>> s4;
```

set

STL


```
set<int> s;
s.insert(1);  s.insert(3); s.insert(2);

cout << "s.size() = " << s.size() << '\n';

pair<set<int>::iterator, bool> result = s.insert(4);
cout << "result iterator = " << *result.first << '\n';
cout << "result bool = " << result.second << '\n';

auto result2 = s.insert(3);
cout << "result2 iterator = " << *result2.first << '\n';
cout << "result2 bool = " << result2.second << '\n';
```

set

STL


```
s.size() = 3
result iterator = 4
result bool = 1
result2 iterator = 3
result2 bool = 0
```

set

STL


```
set<int> s = {1, 2, 3, 4, 5};
s.erase(s.begin());
cout << "s.size() = " << s.size() << '\n';

auto it = s.begin();
++it;
cout << "*it = " << *it << '\n';
it = s.erase(it);
cout << "*it = " << *it << '\n';
cout << "s.size() = " << s.size() << '\n';
```

set

STL


```
s.size() = 4
*it = 3
*it = 4
s.size() = 3
```

set

STL


```
set<int> s = {5, 2, 4, 1, 3, 7, 6};

for (auto it = s.begin(); it != s.end(); ++it) {
    cout << *it << ' ';
}
cout << '\n';

for (auto x : s) {
    cout << x << ' ';
}
cout << '\n';
```

중복 빼고 정렬하기

https://www.acmicpc.net/problem/10867

- https://gist.github.com/Baekjoon/7b42ea52c76c026bbbf9

set

STL


```
set<int> s = {7, 5, 3, 1};

auto it = s.find(1);
print(s, it);

it = s.find(2);
print(s, it);

it = s.find(3);
print(s, it);

it = s.find(4);
print(s, it);
```

set

STL


```
void print(set<int> &s, set<int>::iterator it) {
    if (it == s.end()) {
        cout << "end\n";
    } else {
        cout << *it << '\n';
    }
}
```

set

STL


```
set<int> s = {7, 1, 5, 3};

for (int i=1; i<=9; i++) {
    cout << "s.count(" << i << ") = " << s.count(i) << '\n';
}
```

숫자 카드

https://www.acmicpc.net/problem/10815

- https://gist.github.com/Baekjoon/39dc82f84e81ddc2b905

숫자 카드 2

https://www.acmicpc.net/problem/10816

- https://gist.github.com/Baekjoon/7a9151ba2beefaaf5944

map

map

STL


```
map<int, int> d1;
map<int, int> d2 = {{1, 2}, {3, 4}, {5, 6}};

cout << "d1.size() = " << d1.size() << '\n';
cout << "d2.size() = " << d2.size() << '\n';

d1[10] = 20;
cout << "d1[10] = " << d1[10] << '\n';
cout << "d2[1] = " << d2[1] << '\n';
cout << "d2[2] = " << d2[2] << '\n';
cout << "d2[3] = " << d2[3] << '\n';
cout << "d2[4] = " << d2[4] << '\n';
cout << "d2[5] = " << d2[5] << '\n';
```

map

STL


```
map<int, int> d1;
map<int, int> d2;

for (int i=1; i<=9; i+=2) {
    d1[i] = i*i;
    d2[i] = i*i;
}

cout << "d1.size() = " << d1.size() << '\n';
cout << "d2.size() = " << d2.size() << '\n';
```

map

STL

```
for (int i=1; i<=10; i++) {
    if (d1[i]) {
        cout << i << ' ';
    }
}
cout << '\n';
for (int i=1; i<=10; i++) {
    if (d2.count(i)) {
        cout << i << ' ';
    }
}
cout << '\n';
cout << "d1.size() = " << d1.size() << '\n';
cout << "d2.size() = " << d2.size() << '\n';
```

map

STL


```
d1.size() = 5
d2.size() = 5
1 3 5 7 9
1 3 5 7 9
d1.size() = 10
d2.size() = 5
```

map

STL


```
map<int, int> d = {{1, 2}, {3, 4}, {5, 6}};

for (auto it = d.begin(); it != d.end(); ++it) {
    cout << (it->first) << ' ' << (it->second) << '\n';
}

for (auto p : d) {
    cout << p.first << ' ' << p.second << '\n';
}
```

저항

https://www.acmicpc.net/problem/1076

- https://gist.github.com/Baekjoon/9a5d86db0452703ec384

듣보잡

https://www.acmicpc.net/problem/1764

- https://gist.github.com/Baekjoon/d6ee3fbcda9555e1103b

stack

stack

STL


```
stack<int> s1;

stack<int, list<int>> s2;

deque<int> d = {1, 2, 3, 4, 5};
stack<int> s3(d);
```

stack

STL


```
stack<int> s;

for (int i=1; i<=5; i++) {
    s.push(i);
}

for (int i=0; i<2; i++) {
    cout << s.top() << '\n';
    s.pop();
}

cout << "size = " << s.size() << '\n';
```

stack

STL


```
for (int i=10; i>=6; i--) {
    s.push(i);
}

cout << "size = " << s.size() << '\n';
cout << "empty = " << s.empty() << '\n';

while (!s.empty()) {
    cout << s.top() << '\n';
    s.pop();
}

cout << "size = " << s.size() << '\n';
cout << "empty = " << s.empty() << '\n';
```

stack

STL


```
stack<pair<int,int>> s;
s.push(make_pair(1,2));
s.push({3, 4});
s.emplace(5, 6);

while (!s.empty()) {
    auto p = s.top();
    cout << p.first << ' ' << p.second << '\n';
    s.pop();
}
```

스택

https://www.acmicpc.net/problem/10828

- https://gist.github.com/Baekjoon/1f45c9069e527209fdc0

queue

queue

STL


```
queue<int> q1;

queue<int, list<int>> q2;

deque<int> d = {1, 2, 3, 4, 5};
queue<int> q3(d);
```

queue

STL


```
queue<int> q;

for (int i=1; i<=5; i++) {
    q.push(i);
}

for (int i=0; i<2; i++) {
    cout << q.front() << ' ' << q.back() << '\n';
    q.pop();
}

cout << "size = " << q.size() << '\n';
cout << "empty = " << q.empty() << '\n';
```

queue

STL

```
for (int i=6; i<=10; i++) {
    q.push(i);
    cout << "back = " << q.back() << '\n';
}

while (!q.empty()) {
    cout << q.front() << ' ' << q.back() << '\n';
    q.pop();
}

cout << "size = " << q.size() << '\n';
cout << "empty = " << q.empty() << '\n';
```

queue

STL


```
queue<pair<int,int>> q;
q.push(make_pair(1,2));
q.push({3,4});
q.emplace(5,6);

while (!q.empty()) {
    auto p = q.front();
    cout << p.first << ' ' << p.second << '\n';
    q.pop();
}
```

https://www.acmicpc.net/problem/10845

- https://gist.github.com/Baekjoon/275f19126d5d0f54641f

조세퍼스 문제

https://www.acmicpc.net/problem/1158

- https://gist.github.com/Baekjoon/8b4b4a815349c97b369d

priority_queue

priority_queue

STL


```
vector<int> a = {5, 2, 4, 1, 3};

priority_queue<int> q1;

for (int x : a) {
    q1.push(x);
}

while (!q1.empty()) {
    cout << q1.top() << ' ';
    q1.pop();
}
cout << '\n';
```

priority_queue

STL


```
vector<int> a = {5, 2, 4, 1, 3};

priority_queue<int> q2;
for (int x : a) {
    q2.push(-x);
}

while (!q2.empty()) {
    cout << -q2.top() << ' ';
    q2.pop();
}
cout << '\n';
```

priority_queue

STL


```
vector<int> a = {5, 2, 4, 1, 3};

priority_queue<int, vector<int>, greater<int>> q3;
for (int x : a) {
    q3.push(x);
}
while (!q3.empty()) {
    cout << q3.top() << ' ';
    q3.pop();
}
cout << '\n';
```

priority_queue

STL


```
priority_queue<int> q;

for (int x : {2, 1, 4, 3, 5}) {
    cout << "x = " << x << '\n';
    q.push(x);
    cout << "top = " << q.top() << '\n';
}

cout << "size = " << q.size() << '\n';
cout << "empty = " << q.empty() << '\n';
```

priority_queue

STL


```
while (!q.empty()) {
    cout << "top = " << q.top() << '\n';
    q.pop();
}

cout << "size = " << q.size() << '\n';
cout << "empty = " << q.empty() << '\n';
```

최소 힙

https://www.acmicpc.net/problem/1927

- https://gist.github.com/Baekjoon/a5de8034d60ad0466a24
- https://gist.github.com/Baekjoon/9ad24438f9124c26a461

bitset

bitset

STL

```
bitset<8> b1; // 0,0,0,0,0,0,0,0

bitset<10> b2(88); // 0,0,0,1,0,1,1,0,0,0

bitset<8> b3("10110"); // 0,0,0,1,0,1,1,0
```

bitset

STL

```
bitset<10> b(88); // 0,0,0,1,0,1,1,0,0,0

for (int i=0; i<b.size(); i++) {
    cout << i << ": " << b[i] << '\n';
}
```

bitset

STL


```
bitset<10> b(88); // 0,0,0,1,0,1,1,0,0,0
cout << b << '\n'; //0001011000

cout << b.test(4) << '\n'; // 1
cout << b.test(5) << '\n'; // 0

b.set(0);
cout << b << '\n'; // 0001011001

b.reset(3);
cout << b << '\n'; // 0001010001

b.flip(2);
cout << b << '\n'; // 0001010101
```

bitset

STL

```
bitset<10> b(88); // 0,0,0,1,0,1,1,0,0,0
cout << b << '\n'; //0001011000

b.flip();
cout << b << '\n'; //1110100111

b.set();
cout << b << '\n'; //1111111111

b.reset();
cout << b << '\n'; //0000000000
```

bitset

STL

```
bitset<10> b(88); // 0,0,0,1,0,1,1,0,0,0
cout << b << '\n'; //0001011000

cout << b.all() << '\n'; // false
cout << b.any() << '\n'; // true
cout << b.none() << '\n'; // false

cout << b.count() << '\n'; // 3
```

bitset

STL


```
bitset<10> b1(88); // 0,0,0,1,0,1,1,0,0,0
bitset<10> b2(47); // 0,0,0,0,1,0,1,1,1,1

cout << (b1 & b2) << '\n'; //0000001000
cout << (b1 | b2) << '\n'; //0001111111
cout << (b1 ^ b2) << '\n'; //0001110111
cout << ~(b1) << '\n'; //1110100111

cout << (b1 << 2) << '\n'; //0101100000
cout << (b2 >> 2) << '\n'; //0000001011
```

이진수 연산

https://www.acmicpc.net/problem/12813

- https://gist.github.com/Baekjoon/1a2d791133196df6b0daaa9739aa05fc