Day-21

Control Flow

if Statement

- if문은 지정된 조건이 true로 평가되면 첫 번째 분기를 실행
- 그렇지 않다면 두 번째 분기를 실행함
- 삼항 연산자
  - 형식
    - <cond> ? <표현식1> : <표현식2>
  - <표현식1>과 <표현식2>는 동일하거나 변환 가능한 타입의 값을 반환해야 함

```
int value = (a==b) ? a : b; // a와 b가 같다면 value = a
                            // 그렇지 않다면 value = b
```

for and while Loops

- for
  - 반복 횟수를 알고 있을 때 사용
- while
  - 반복 횟수를 알 수 없을 때 사용
- do-while
  - 반복 횟수를 알 수 없지만 적어도 한번 이상의 반복이 있을 때 사용
- C++은 “in loop” 정의를 허용

```
for(int i = 0, k = 0; i < 10; i++, k += 2)
```

- 무한 루프

```
for(;;) // while(true)와 동일
```

- 점프 문(break, continue, return)

```
for(int i = 0; i < 10; i++){
        if(<condition>
            break;        // 반복문 빠져나옴
        if(<condition>)
            continue;     // 다음 반복문 시작, i++
        return;          // 함수 종료
}
```

Range-based for Loop

- C++11에는 범위 기반 for 루프가 도입되어 기존 for 루프 구조의 복잡성을 간소화
- 값 범위에서 작동하는 for 루프와 동일하지만 더 안전함
- 범위 기반 for 루프는 사용자가 루프의 시작, 끝, 증분값을 지정할 필요가 없음

```
for (int v : { 3, 2, 1 }) // INITIALIZER LIST
    cout << v << " "; // print: 3 2 1

int values[] = { 3, 2, 1 };
for (int v : values) // ARRAY OF VALUES
    cout << v << " "; // print: 3 2 1

for (auto c : "abcd") // RAW STRING
    cout << c << " "; // print: a b c d
```

- 범위 기반 for 루프는 세 가지 경우에 적용 가능함
  1. 고정 크기 배열 `int array[3]`, `“abcd”`
  2. 분기 초기화 리스트 `{1, 2, 3}`
  3. `begin()`, `end()` 메서드가 있는 모든 객체 : vector 등

```
vector vec{1, 2, 3, 4};

for(auto x : vec){
    cout << x << ",";
}
// print : "1, 2, 3, 4"
```

- C++17은 구조체 바인딩을 위한 범위 기반 루프의 개념을 확장

```
struct A {
    int x;
    int y;
};

A array[] = { {1,2}, {5,6}, {7,1} };
for (auto [x1, y1] : array){
    cout << x1 << "," << y1 << " "; // print: 1,2 5,6 7,1
}
```