

2022 D.Com 신입생 C++ 스터디 4차시 과제

- 함수, 매개변수, 오버로딩, 템플릿

1, 2번 과제에서 알고 가기 : Vector - STL(Standard Template Library) Container

Vector는 C++ 표준 라이브러리(Standard Template Library)에 있는 컨테이너로 사용자가 손쉽게 사용하기 위해 정의된 class입니다. Vector의 가장 큰 장점은 동적으로 원소를 추가할 수 있으며 크기가 자동으로 늘어난다는 점입니다. 쉽게 말해 크기가 가변적으로 변하는 배열이라고 할 수 있습니다.

- <https://coding-factory.tistory.com/596> 참고하기.

- 1, 2번 과제

아래 두 프로그램을 각각 작성하시오

작성 예시	<pre>#include<iostream> #include<vector> using namespace std; template</* 구현 */> bool less_than(/* 구현 */) { return a < b; } template<class T> T sum(const vector<T>& v) { /* 구현 */} int main() { cout << less_than(2, 3) << endl; cout << less_than(2.1, 2.9) << endl; cout << less_than(2, 2.5) << endl; vector<int> v1{ 1, 2, 3, 4 }; vector<double> v2{ 10.1, 20.2, 30.3, 40.4 }; cout << sum(v: v1) << endl; cout << sum(v: v2) << endl; return 0; }</pre>	출력 결과 <pre>1 1 1 10 101 Process finished with exit code 0</pre>
작성 예시	<pre>#include<iostream> #include <vector> using namespace std; int main() { int ary[] = { [0]: 1, [1]: 2, [2]: 3, [3]: 4 }; int* pBegin, *pEnd; pBegin = ary; pEnd = ary + 4; for (/* 구현 */) cout << /* 구현 */ << "\t"; cout << endl; //Hint : auto, begin(), end() vector<int> v{ 10, 20, 30, 40 }; auto iter_begin = /* 구현 */; auto iter_end = /* 구현 */; for (/* 구현 */) cout << /* 구현 */ << "\t"; cout << endl; return 0; }</pre>	출력 결과 <pre>1 2 3 4 10 20 30 40 Process finished with exit code 0</pre>

- 3번 과제

정수 n 과, n 개의 정수가 입력될 때, 함수를 구현하여 n 개 정수의 합을 출력하는 프로그램을 작성하시오. (단, $1 \leq n \leq 100$)

입력 예시	<pre>7 3 3 3 5 5 5 7</pre>	출력 예시	<pre>31 Process finished with exit code 0</pre>
----------	----------------------------	----------	---

- 4번 과제

정수 n 이 입력될 때, '참고'의 규칙에 따라 n 개의 원판의 이동 횟수 및 이동 경로를 아래의 예시와 같이 출력하는 프로그램을 작성하시오. (단, $1 \leq n \leq 20$)

참고 : 하노이 탑 문제

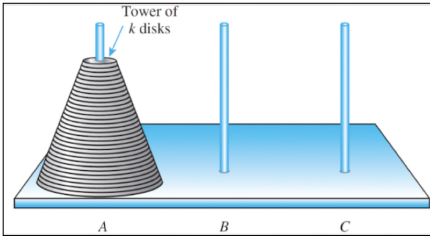
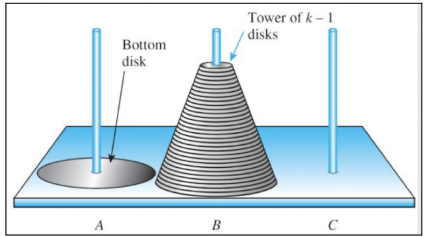
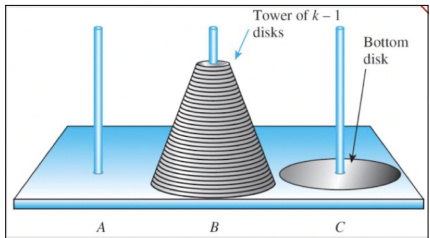
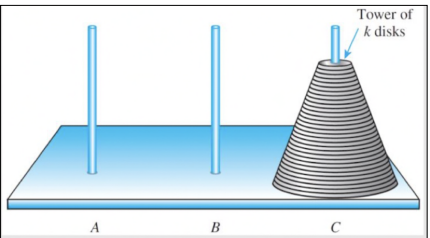
세 개의 장대(A,B,C)가 있고 첫 번째 장대(A)에는 반경이 서로 다른 k 개의 원판이 쌓여 있다. 각 원판은 반경이 큰 순서대로 쌓여있다.

다음 규칙에 따라 원판을 첫 번째 장대에서 세 번째 장대로 옮기려 한다.

규칙 1. 한 번에 한 개의 원판만을 다른 장대로 옮길 수 있다.

규칙 2. 쌓아 놓은 원판은 항상 위의 것이 아래의 것보다 작아야 한다.

힌트 : 재귀적으로 생각하기

1		2	
3		4	

입력 예시	<pre>3</pre>	출력 예시	<pre>7 1 3 1 2 3 2 1 3 2 1 2 3 1 3</pre>
----------	--------------	----------	--

- 5번 과제

연산자 오버로딩을 이용하여, 두 Complex 클래스의 덧셈이 가능하도록 프로그램을 작성하시오.

작성
예시

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

class Complex
{
public:
    int real;
    int img;

    Complex(int real, int img)
    {
        this->real = real;
        this->img = img;
    }

    /** 구현 */
};

int main()
{
    Complex a = Complex( real: 1,  img: 2);
    Complex b = Complex( real: 3,  img: 4);

    Complex c = a + b;
    cout << c.real << " + " << c.img << "i" << endl;

    return 0;
}
```

출력
결과

4 + 6i

Process finished with exit code 0