

1. 커스텀 메시지 파일

```
int64 a
int32[] b
```

2. 헤더파일

```
#include "rclcpp/rclcpp.hpp"
#include "custom_interfaces/msg/add_two_ints.hpp"
#include <vector>
#include <string>

class CmdNode : public rclcpp::Node
{
public:
    CmdNode();
    void run_once(); // CMD 입력 + 퍼블리시

private:
    void topic_callback(const custom_interfaces::msg::AddTwoInts::SharedPtr msg);

    rclcpp::Publisher<custom_interfaces::msg::AddTwoInts>::SharedPtr publisher_;
    rclcpp::Subscription<custom_interfaces::msg::AddTwoInts>::SharedPtr subscription_;
};
```

통신 하기 위한 클래스를 생성하였다.

3. 메인 윈도우 파일

```
#include "custom_interfaces/mainwindow.h"
#include <iostream>
#include <sstream>

CmdNode::CmdNode() : Node("cmd_node")
{
    publisher_ = this->create_publisher<custom_interfaces::msg::AddTwoInts>(
        "add_two_ints_topic", 10);

    subscription_ = this->create_subscription<custom_interfaces::msg::AddTwoInts>(
        "add_two_ints_topic", 10,
        std::bind(&CmdNode::topic_callback, this, std::placeholders::_1));

    RCLCPP_INFO(this->get_logger(), "cmd_node started!");
}
```

Node("cmd_node")로 부모 클래스를 초기화하면서 노드 이름을 "cmd_node"로 지정한다. create_publisher를 이용해 퍼블리셔를 "add_two_ints_topic" 토픽에 만들고, 큐 사이즈는 10으로 설정한다.

create_subscription으로 같은 토픽을 구독하는 서브스크립션을 만든다. 메시지를 수신하면 topic_callback()이 호출되도록 연결한다.

마지막으로 RCLCPP_INFO를 사용해 "cmd_node started!"라는 로그를 출력해 노드가 정상적으로 시작했음을 알린다.

```
void CmdNode::run_once()
{
    int64_t a_input;
    std::string line;
    std::vector<int32_t> b_input;

    // a 입력
    while (true)
    {
        std::cout << "Enter a (int64): ";
        std::getline(std::cin, line);
        std::istringstream iss(line);
        if (iss >> a_input)
        {
            break;
        }
        else
        {
            std::cout << "Invalid input. Please enter an integer." << std::endl;
        }
    }

    // b 벡터 입력
    while (true)
    {
        std::cout << "Enter b vector (space-separated ints, end with ENTER): ";
        std::getline(std::cin, line);
        std::istringstream iss(line);
        b_input.clear();
        int num;
        bool valid = true;

        while (iss >> num)
        {
            b_input.push_back(num);
        }

        if (b_input.empty())
        {
            std::cout << "Vector cannot be empty. Try again." << std::endl;
            continue;
        }

        if (iss.fail() && !iss.eof())
        {
            std::cout << "Invalid input detected. Only integers are allowed." << std::endl;
            continue;
        }

        break;
    }

    custom_interfaces::msg::AddTwoInts msg;
    msg.a = a_input;
    msg.b = b_input;

    publisher_ -> publish(msg);
    RCLCPP_INFO(this->get_logger(), "Published a=%ld, b_size=%zu", msg.a, msg.b.size());
}
```

먼저 사용자에게 정수 a를 입력받는다. std::getline으로 문자열을 받고 변환을 시도한다. 올바른 정수(int64_t)가 들어올 때까지 반복한다.

다음으로 b 벡터를 입력받는다. 한 줄에 공백으로 구분된 여러 개의 정수를 읽어서 std::vector<int32_t>에 넣는다. 만약 벡터가 비어 있으면 다시 입력을 요구한다.

정수가 아닌 값이 들어오면 에러 메시지를 출력하고 다시 입력을 받는다. 이후 msg.a에 앞서 입력한 정수를, msg.b에 벡터 전체를 저장하고 퍼블리셔(publisher_)를 통해 메시지를 발행한다.

마지막으로 RCLCPP_INFO를 사용해 터미널에 로그를 출력한다. 여기서는 발행된 a 값과 벡터의 크기를 표시한다.

```
void CmdNode::topic_callback(const custom_interfaces::msg::AddTwoInts::SharedPtr msg)
{
    std::cout << "Received: a=" << msg->a << ", b=[";
    for (size_t i = 0; i < msg->b.size(); ++i)
    {
        std::cout << msg->b[i];
        if (i < msg->b.size() - 1)
            std::cout << ", ";
    }
    std::cout << "]" << std::endl;
}
```

구독한 토픽으로부터 custom_interfaces::msg::AddTwoInts 메시지를 받으면 호출된다.

먼저 msg->a 값을 출력한다.

이어서 msg->b 벡터의 내용을 출력한다. 벡터 크기만큼 반복문을 돌면서 각 원소를 표시하고, 마지막 원소가 아닐 경우 , 를 붙여 가독성을 높인다.

벡터 출력이 끝나면]로 닫고 줄바꿈을 한다.