```cpp
#include <iostream>
#include <fstream>
#include <filesystem>
#include <string>
#include "myfile.hpp"

OutMyFile::OutMyFile(const int n, const std::filesystem::path &path) : N(n), file_path(path) {
  return;
}

void OutMyFile::out(const double * const val) const {

  std::ofstream outfile(file_path);
  outfile << std::setprecision(std::numeric_limits<double>::max_digits10)
          << std::showpoint;

  for (int i = 0; i < N; ++i) {
    outfile << val[i] << std::endl;
  }

  return;
}

void OutMyFile::out(const double * const xval, const double * const yval) const {

  std::ofstream outfile(file_path);
  outfile << std::setprecision(std::numeric_limits<double>::max_digits10)
          << std::showpoint;

  for (int i = 0; i < N; ++i) {
    outfile << xval[i] << " " << yval[i] << std::endl;
  }

  return;
}

InMyFile::InMyFile(const std::filesystem::path &path) :file_path(path), N(getNum()) {

  std::ifstream infile(file_path);

  if (!infile) {
    std::cout << "cannot open" << std::endl;
    return;
  }

  arr = new double[N];
  for (int i = 0; i < N; ++i) {
    infile >> arr[i];
  }
  infile.close();

  return;
}

InMyFile::~InMyFile() {
  if (arr != nullptr) {
    delete[] arr;
  }
  return;
}

inline bool InMyFile::check_index(const int idx) const {
  if ((idx < 0) || (N <= idx)) return false;
  return true;
}

int InMyFile::getNum() const{

  std::ifstream infile(file_path);

  int count = 0;
  std::string line;

  while (std::getline(infile, line)) ++count;

  infile.close();

  return count;
}

double InMyFile::operator() (const int idx) const {
  if(!check_index(idx)) {
    std::cout << "error: out of index" << std::endl;
    return 0;
  }
  return arr[idx];
}

```