

## HW4

### 1. 소스코드

- myvector.h

```
#ifndef MYVECTOR_H
#define MYVECTOR_H

#include <iostream>
#include <stdexcept>

template <typename T>
class MyVector {
private:
    T* data; // 동적 배열
    int capacity; // 배열의 용량
    int size; // 현재 원소의 개수

    void resize(); // 배열 크기 늘리는 함수

public:
    MyVector(); // 기본 생성자
    MyVector(const MyVector& other); // 복사 생성자
    ~MyVector(); // 소멸자

    void push_back(const T& element);
    T& at(int index);
    const T& operator[](int index) const;
    MyVector& operator=(const MyVector& other); // 복사 대입 연산자
    MyVector operator+(const MyVector& other); // 벡터 덧셈 연산자

    bool empty() const;
    int sizeGet() const;
    void erase(int index);
    void insert(int index, const T& element);

    T* begin();
```

```
T*T*end();
```

```
};
```

```
// 기본 생성자
```

```
template<typename T>
```

```
MyVector<T>::MyVector() : data(nullptr), capacity(0), size(0) {}
```

```
// 복사 생성자
```

```
template<typename T>
```

```
MyVector<T>::MyVector(const MyVector&other) : capacity(other.capacity),  
size(other.size) {
```

```
    data = new T[capacity];
```

```
    std::copy(other.data, other.data+size, data);
```

```
}
```

```
// 소멸자
```

```
template<typename T>
```

```
MyVector<T>::~~MyVector() {
```

```
    delete[] data;
```

```
}
```

```
// 원소 추가
```

```
template<typename T>
```

```
void MyVector<T>::push_back(const T&element) {
```

```
    if (size == capacity) {
```

```
        resize();
```

```
    }
```

```
    data[size++] = element;
```

```
}
```

```
// 특정 인덱스의 원소 반환
```

```
template<typename T>
```

```
T MyVector<T>::at(int index) {
```

```
    return data[index];
```

```
}
```

```

// [] 연산자 오버로딩
template<typename T>
TMyVector<T>::operator[](int index) {
    return at(index);
}

// = 연산자 오버로딩
template<typename T>
MyVector<T>& MyVector<T>::operator=(const MyVector& other) {
    if (this == &other) return *this; // 자기 자신을 대입하는 경우, 그대로 반환

    delete[] data; // 기존 데이터를 해제

    // 새로운 데이터 할당 및 크기 설정 k
    size = other.size;
    capacity = other.capacity;
    data = new T[capacity];

    // 다른 벡터의 데이터를 복사
    for (int i = 0; i < size; i++) {
        data[i] = other.data[i];
    }

    return *this; // 자기 자신 반환
}

// + 연산자 오버로딩 (두 벡터 더하기)
template<typename T>
MyVector<T> MyVector<T>::operator+(const MyVector& other) {
    MyVector<T> result;

    for (int i = 0; i < this->size; ++i) {
        result.push_back(this->data[i]); // 왼쪽 벡터의 원소 추가
    }

    for (int i = 0; i < other.size; ++i) {
        result.push_back(other.data[i]); // 오른쪽 벡터의 원소 추가
    }
}

```

```

    }

    rreturnresult;
}

// 내부적으로 배열 크기 늘리기
template<typenameT>
voidMyVector<T>::resize() {
    inintincreaseAmount =5;// 증가량 설정
    capacity +=+=increaseAmount;// 현재 용량에 증가량 추가
    T*n*newData =newT[capacity];// 새로운 배열 생성

    // 기존 데이터를 새 배열로 복사
    ststd::copy(data, data +size, newData);

    dedelete[]data;// 이전 배열 메모리 해제
    data =n=newData;// 새로운 배열로 데이터 포인터 업데이트
}

// 벡터가 비어있는지 확인
template<typenameT>
boolMyVector<T>::empty() const{
    rreturnsize ==0;
}

// 벡터의 원소 개수 반환
template<typenameT>
intMyVector<T>::sizeGet() const{
    rreturnsize;
}

// 특정 인덱스의 원소 삭제
template<typenameT>
voidMyVector<T>::erase(intindex) {
    ifif(index <0||index >=size) return;// 인덱스 범위 체크 후 무시
    fofor(inti =index; i <size -1; i++) {

```

```

        data[data[i] = data[i + 1];
    }

    size--;
}

// 특정 인덱스에 원소 삽입
template<typename T>
void MyVector<T>::insert(int index, const T& element) {
    if (index < 0 || index > size) return; // 인덱스 범위 체크 후 무시
    if (size == capacity) {
        resize();
    }
    for (int i = size; i > index; i--) {
        data[i] = data[i - 1];
    }
    data[index] = element;
    size++;
}

// 시작 원소에 대한 포인터 반환
template<typename T>
T* MyVector<T>::begin() {
    return data; // 첫 번째 원소의 포인터 반환
}

// 끝 원소 다음에 대한 포인터 반환
template<typename T>
T* MyVector<T>::end() {
    return data + size; // 마지막 원소 다음의 포인터 반환
}

#include "myvector.h"

#endif // MYVECTOR_H

```

```
#include "myvector.h"

#include <iostream>

using namespace std;

int main() {
    MyVector<i<int>>v;

    int number;

    while(1) {
        // 기능을 선택하기 위해 어떤 기능이 있는지 보여준다.
        std::cout << "1. push_back()" << std::endl;
        std::cout << "2. erase()" << std::endl;
        std::cout << "3. insert()" << std::endl;
        std::cout << "4. show" << std::endl;
        std::cout << "5. empty()" << std::endl;
        std::cout << "6. size()" << std::endl;
        std::cout << "7. exit" << std::endl;

        // 기능을 선택한다.
        std::cin >> number;
        std::cout << std::endl;

        int n;
        int i;

        // 7이 입력 되었을 때 프로그램은 종료 된다.
        if(number == 7) break;

        // 스위치문을 통해서 선택한 기능을 실행시킨다.
        switch(number) {
            case 1:
            case 2:
            case 3:
            case 4:
            case 5:
            case 6:
            case 7:
            case 8:
            case 9:
            case 10:
            case 11:
            case 12:
            case 13:
            case 14:
            case 15:
            case 16:
            case 17:
            case 18:
            case 19:
            case 20:
            case 21:
            case 22:
            case 23:
            case 24:
            case 25:
            case 26:
            case 27:
            case 28:
            case 29:
            case 30:
            case 31:
            case 32:
            case 33:
            case 34:
            case 35:
            case 36:
            case 37:
            case 38:
            case 39:
            case 40:
            case 41:
            case 42:
            case 43:
            case 44:
            case 45:
            case 46:
            case 47:
            case 48:
            case 49:
            case 50:
            case 51:
            case 52:
            case 53:
            case 54:
            case 55:
            case 56:
            case 57:
            case 58:
            case 59:
            case 60:
            case 61:
            case 62:
            case 63:
            case 64:
            case 65:
            case 66:
            case 67:
            case 68:
            case 69:
            case 70:
            case 71:
            case 72:
            case 73:
            case 74:
            case 75:
            case 76:
            case 77:
            case 78:
            case 79:
            case 80:
            case 81:
            case 82:
            case 83:
            case 84:
            case 85:
            case 86:
            case 87:
            case 88:
            case 89:
            case 90:
            case 91:
            case 92:
            case 93:
            case 94:
            case 95:
            case 96:
            case 97:
            case 98:
            case 99:
            case 100:
            case 101:
            case 102:
            case 103:
            case 104:
            case 105:
            case 106:
            case 107:
            case 108:
            case 109:
            case 110:
            case 111:
            case 112:
            case 113:
            case 114:
            case 115:
            case 116:
            case 117:
            case 118:
            case 119:
            case 120:
            case 121:
            case 122:
            case 123:
            case 124:
            case 125:
            case 126:
            case 127:
            case 128:
            case 129:
            case 130:
            case 131:
            case 132:
            case 133:
            case 134:
            case 135:
            case 136:
            case 137:
            case 138:
            case 139:
            case 140:
            case 141:
            case 142:
            case 143:
            case 144:
            case 145:
            case 146:
            case 147:
            case 148:
            case 149:
            case 150:
            case 151:
            case 152:
            case 153:
            case 154:
            case 155:
            case 156:
            case 157:
            case 158:
            case 159:
            case 160:
            case 161:
            case 162:
            case 163:
            case 164:
            case 165:
            case 166:
            case 167:
            case 168:
            case 169:
            case 170:
            case 171:
            case 172:
            case 173:
            case 174:
            case 175:
            case 176:
            case 177:
            case 178:
            case 179:
            case 180:
            case 181:
            case 182:
            case 183:
            case 184:
            case 185:
            case 186:
            case 187:
            case 188:
            case 189:
            case 190:
            case 191:
            case 192:
            case 193:
            case 194:
            case 195:
            case 196:
            case 197:
            case 198:
            case 199:
            case 200:
            case 201:
            case 202:
            case 203:
            case 204:
            case 205:
            case 206:
            case 207:
            case 208:
            case 209:
            case 210:
            case 211:
            case 212:
            case 213:
            case 214:
            case 215:
            case 216:
            case 217:
            case 218:
            case 219:
            case 220:
            case 221:
            case 222:
            case 223:
            case 224:
            case 225:
            case 226:
            case 227:
            case 228:
            case 229:
            case 230:
            case 231:
            case 232:
            case 233:
            case 234:
            case 235:
            case 236:
            case 237:
            case 238:
            case 239:
            case 240:
            case 241:
            case 242:
            case 243:
            case 244:
            case 245:
            case 246:
            case 247:
            case 248:
            case 249:
            case 250:
            case 251:
            case 252:
            case 253:
            case 254:
            case 255:
            case 256:
            case 257:
            case 258:
            case 259:
            case 260:
            case 261:
            case 262:
            case 263:
            case 264:
            case 265:
            case 266:
            case 267:
            case 268:
            case 269:
            case 270:
            case 271:
            case 272:
            case 273:
            case 274:
            case 275:
            case 276:
            case 277:
            case 278:
            case 279:
            case 280:
            case 281:
            case 282:
            case 283:
            case 284:
            case 285:
            case 286:
            case 287:
            case 288:
            case 289:
            case 290:
            case 291:
            case 292:
            case 293:
            case 294:
            case 295:
            case 296:
            case 297:
            case 298:
            case 299:
            case 300:
            case 301:
            case 302:
            case 303:
            case 304:
            case 305:
            case 306:
            case 307:
            case 308:
            case 309:
            case 310:
            case 311:
            case 312:
            case 313:
            case 314:
            case 315:
            case 316:
            case 317:
            case 318:
            case 319:
            case 320:
            case 321:
            case 322:
            case 323:
            case 324:
            case 325:
            case 326:
            case 327:
            case 328:
            case 329:
            case 330:
            case 331:
            case 332:
            case 333:
            case 334:
            case 335:
            case 336:
            case 337:
            case 338:
            case 339:
            case 340:
            case 341:
            case 342:
            case 343:
            case 344:
            case 345:
            case 346:
            case 347:
            case 348:
            case 349:
            case 350:
            case 351:
            case 352:
            case 353:
            case 354:
            case 355:
            case 356:
            case 357:
            case 358:
            case 359:
            case 360:
            case 361:
            case 362:
            case 363:
            case 364:
            case 365:
            case 366:
            case 367:
            case 368:
            case 369:
            case 370:
            case 371:
            case 372:
            case 373:
            case 374:
            case 375:
            case 376:
            case 377:
            case 378:
            case 379:
            case 380:
            case 381:
            case 382:
            case 383:
            case 384:
            case 385:
            case 386:
            case 387:
            case 388:
            case 389:
            case 390:
            case 391:
            case 392:
            case 393:
            case 394:
            case 395:
            case 396:
            case 397:
            case 398:
            case 399:
            case 400:
            case 401:
            case 402:
            case 403:
            case 404:
            case 405:
            case 406:
            case 407:
            case 408:
            case 409:
            case 410:
            case 411:
            case 412:
            case 413:
            case 414:
            case 415:
            case 416:
            case 417:
            case 418:
            case 419:
            case 420:
            case 421:
            case 422:
            case 423:
            case 424:
            case 425:
            case 426:
            case 427:
            case 428:
            case 429:
            case 430:
            case 431:
            case 432:
            case 433:
            case 434:
            case 435:
            case 436:
            case 437:
            case 438:
            case 439:
            case 440:
            case 441:
            case 442:
            case 443:
            case 444:
            case 445:
            case 446:
            case 447:
            case 448:
            case 449:
            case 450:
            case 451:
            case 452:
            case 453:
            case 454:
            case 455:
            case 456:
            case 457:
            case 458:
            case 459:
            case 460:
            case 461:
            case 462:
            case 463:
            case 464:
            case 465:
            case 466:
            case 467:
            case 468:
            case 469:
            case 470:
            case 471:
            case 472:
            case 473:
            case 474:
            case 475:
            case 476:
            case 477:
            case 478:
            case 479:
            case 480:
            case 481:
            case 482:
            case 483:
            case 484:

```

```

std::cout<<std::cout <<"저장할 숫자를 입력하세요."<<std::endl;

std::cin>>std::cin >>num;

v.push_back(num);

std::cout<<std::cout <<std::endl;

break; break;

}

case2:case2: {

std::cout<<std::cout <<"지울 index를 입력하세요."<<std::endl;

std::cin>>std::cin >>index;

v.erase(v.erase(index));

std::cout<<std::cout <<std::endl;

break; break;

}

case3:case3: {

std::cout<<std::cout <<"index를 입력하세요."<<std::endl;

std::cin>>std::cin >>index;

std::cout<<std::cout <<"숫자를 입력하세요."<<std::endl;

std::cin>>std::cin >>num;

v.insert(v.insert(index, num));

std::cout<<std::cout <<std::endl;

break; break;

}

case4:case4: {

if(v.empty()) std::cout <<"비어있습니다."<<std::endl;

else{ else{

for(int i = 0; i < v.size(); i++) {

std::cout <<std::cout <<v.at(i) <<std::endl;

}

}

}

}

```

```

        std::cout<<std::endl;

        break; break;
    }

    case5:case5: {
        if(v.empty()) std::cout <<"비어있습니다."<<std::endl;
        elsestd::cout <<"비어있지 않습니다."<<std::endl;

        std::cout<<std::endl;

        break; break;
    }

    case6:case6: {
        std::cout<<v.sizeGet() <<std::endl;

        std::cout<<std::endl;

        break; break;
    }

    // 잘못된 숫자가 들어왔을 경우 다시 입력하게 한다. 뻔 톤
    defaultdefault: {
        std::cout<<"다시 입력하세요"<<std::endl;

        std::cout<<std::endl;

        break; break;
    }
}

}

} ㄹ

return0;

}

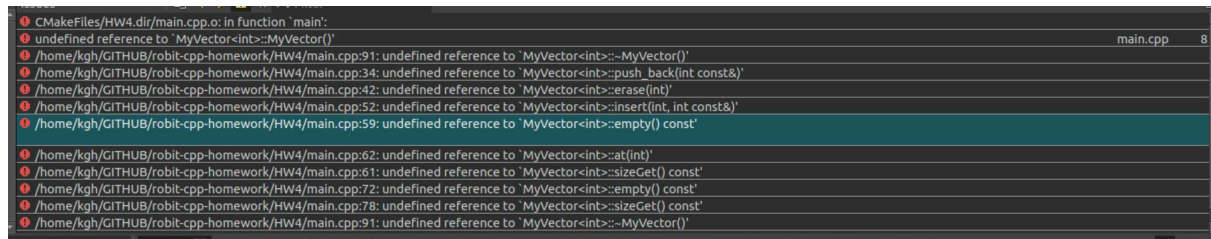
```



## 2. 코드를 작성할 때 가지고 한 생각

디버깅을 할 수 있는 순으로 코드를 작성해 나아갔다.

## 3. 고찰



```
CMakeFiles/HW4.dir/main.cpp.o: In function 'main':
main.cpp:8: undefined reference to 'MyVector<int>::MyVector()'
/home/kgh/GITHUB/robit-cpp-homework/HW4/main.cpp:91: undefined reference to 'MyVector<int>::~MyVector()'
/home/kgh/GITHUB/robit-cpp-homework/HW4/main.cpp:34: undefined reference to 'MyVector<int>::push_back(int const&)'
/home/kgh/GITHUB/robit-cpp-homework/HW4/main.cpp:42: undefined reference to 'MyVector<int>::erase(int)'
/home/kgh/GITHUB/robit-cpp-homework/HW4/main.cpp:52: undefined reference to 'MyVector<int>::insert(int, int const&)'
/home/kgh/GITHUB/robit-cpp-homework/HW4/main.cpp:59: undefined reference to 'MyVector<int>::empty() const'
/home/kgh/GITHUB/robit-cpp-homework/HW4/main.cpp:62: undefined reference to 'MyVector<int>::at(int)'
/home/kgh/GITHUB/robit-cpp-homework/HW4/main.cpp:61: undefined reference to 'MyVector<int>::sizeGet() const'
/home/kgh/GITHUB/robit-cpp-homework/HW4/main.cpp:72: undefined reference to 'MyVector<int>::empty() const'
/home/kgh/GITHUB/robit-cpp-homework/HW4/main.cpp:78: undefined reference to 'MyVector<int>::sizeGet() const'
/home/kgh/GITHUB/robit-cpp-homework/HW4/main.cpp:91: undefined reference to 'MyVector<int>::~MyVector()'
```

myvector.h와 myvector.cpp를 분리하지 못하였다. 분리를 여러 번 시도해 보았지만 수많은 오류가 발생하였다. 오류에 대해서 검색도 해보고 1번에서 .h와 .cpp분리 한 것을 따라해보는 등 많은 시도를 하였다. 하지만 끝내 저 오류를 해결하지 못한 것에 대해서 아쉬움을 느끼고 있다. 또한 과제로 주신 기능 이외에 다양한 기능에 대해 코드를 작성해 보지 못하여 아쉽다.