

객체지향프로그래밍 LAB #14

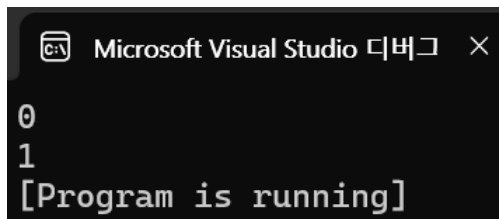
<기초문제>

1.

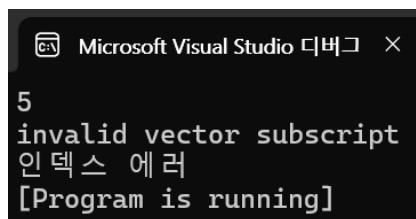
```
#include <iostream>
#include <vector>
using namespace std;
int main() {
    vector<int> v{ 1, 2, 3, 4 };

    int index;
    cin >> index;
    try { cout << v.at(index) << endl; } // v.at(index) VS v[index]
    catch (exception& e) {
        cout << e.what() << endl;
        cout << "인덱스 에러" << endl;
    }

    cout << "[Program is running]" << endl;
    return 0;
}
```



0
1
[Program is running]



5
invalid vector subscript
인덱스 에러
[Program is running]

2.

```
#include <iostream>
#include <vector>
#include <string>
#include <fstream>
using namespace std;

class FileNotFoundException : public exception {
    string message;
public:
    FileNotFoundException(string _m) :
        message("File not found: " + _m) {
    }
}
```

```

        virtual const char* what() const throw() {
            return message.c_str();
        }

};

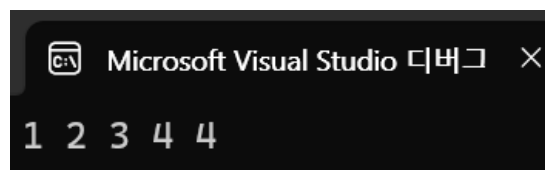
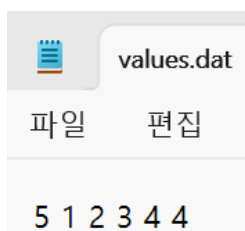
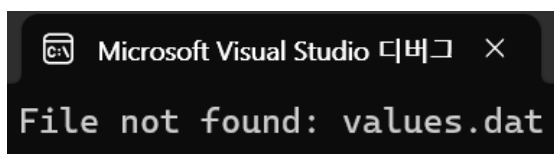
vector<int> load_vector(string filename) {
    ifstream fin(filename);

    // 파일이 열리지 않으면 (파일이 존재하지x)
    if (!fin) {
        // 예외처리 (throw)
        throw FileNotFoundException(filename);
    }

    vector<int> result;
    int num, value;
    // 파일로부터 값을 result에 저장
    // 파일의 form: size, elements (5 1 2 3 4 4)
    fin >> num;
    for (int i = 0; i < num; i++) {
        fin >> value;
        result.push_back(value);
    }
    return result;
}

int main() {
    try {
        vector<int> v = load_vector("values.dat");
        for (int elem : v)
            cout << elem << ' ';
        cout << endl;
    }
    catch (exception& e) {
        cout << e.what() << endl;
    }
    return 0;
}

```

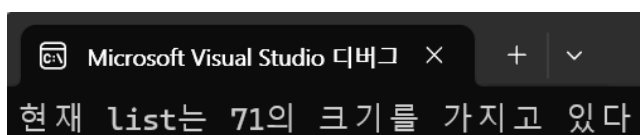


<응용문제>

1.

```
#include <iostream>
#include <vector>
#include <cstdlib>
#include <ctime>
using namespace std;

int main() {
    vector<int> list;
    // vector를 1~100 사이의 random한 크기로 만들고 채우는 코드 구현
    srand((unsigned int)time(0));
    int b = rand() % 100;
    int c;
    for (int i = 0; i <= b; i++)
        list.push_back(i);
    int cnt = -1;
    while (1) {
        cnt++;
        try {
            c = list.at(cnt);
        }
        catch (exception& e) {
            cout << "현재 list는 " << cnt << "의 크기를 가지고 있다";
            break;
        }
    }
    return 0;
}
```



2.

```
#include <iostream>
#include <vector>
#include <fstream>
#include <string>
using namespace std;

class FileNotFoundException : public exception {
    string message; // Identifies the exception and filename
public:
    FileNotFoundException(const string& fname) :
```

```

        message("File W" + fname + "W" not found") {
    }

    virtual const char* what() const throw () {
        return message.c_str();
    }
};

class CStudent {
private:
    string m_Name;
    int m_Number;
    string m_Major;
public:
    CStudent() {}
    ~CStudent() {}
    void setAll(string _name, int _num, string _maj) {
        m_Name = _name;
        m_Number = _num;
        m_Major = _maj;
    }
    void Display() {
        cout << "이름: " << m_Name << endl;
        cout << "학번: " << m_Number << endl;
        cout << "전공: " << m_Major << endl << endl;
    }
};

vector<CStudent> read_file(string& filename) {
    vector<CStudent> a;
    ifstream fin(filename);
    if (!fin)
        throw FileNotFoundException(filename);
    int num, b;
    string c, d;
    fin >> num;
    for (int i = 0; i < num; i++) {
        CStudent value = CStudent();
        fin >> c >> b >> d;
        value.setAll(c, b, d);
        a.push_back(value);
    }
    return a;
}

int main() {
    string str;

    cout << "파일 이름 : ";
    cin >> str;

    try {
        vector<CStudent> numbers = read_file(str);
        for (CStudent value : numbers)

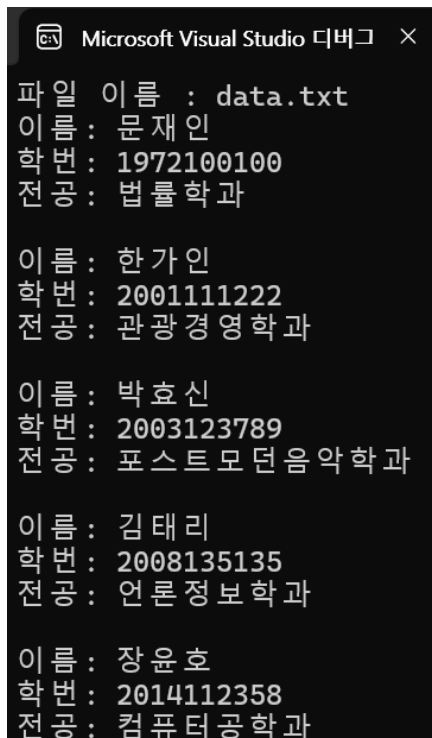
```

```

        value.Display();
    }
    catch (exception& e) {
        cout << e.what() << '\n';
    }

    return 0;
}

```



Microsoft Visual Studio 디버그

```

파일 이름 : data.txt
이름 : 문재인
학번 : 1972100100
전공 : 법률학과

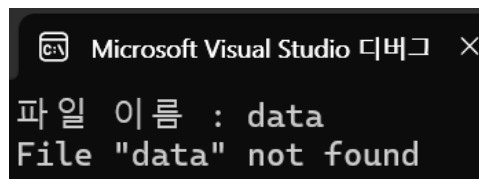
이름 : 한가인
학번 : 2001111222
전공 : 관광경영학과

이름 : 박호신
학번 : 2003123789
전공 : 포스트모던음악학과

이름 : 김태리
학번 : 2008135135
전공 : 언론정보학과

이름 : 장운호
학번 : 2014112358
전공 : 컴퓨터공학과

```



Microsoft Visual Studio 디버그

```

파일 이름 : data
File "data" not found

```

3.

```

#include <iostream>
#include <vector>
#include <fstream>
#include <string>
#include <cstdlib>
#include <ctime>
using namespace std;

class FileNotFoundException : public exception {
    string message;
public:
    FileNotFoundException(const string& fname) :
        message("File W" + fname + "W not found") {
    }

    virtual const char* what() const throw () {
        return message.c_str();
    }
}

```

```

};

int main(){
    ofstream ofs;
    ofs.open("temp.txt");
    // 임의의 10x10 행렬 저장 구현
    int b;
    srand((unsigned int)time(0));
    for (int i = 0; i < 10; i++) {
        for (int j = 0; j < 10; j++) {
            b = rand() % 101;
            ofs << b << " ";
        }
        ofs << endl;
    }
    ofs.close();
    // 파일이름 입력
    string str;
    cout << "파일 이름 : ";
    cin >> str;
    try {
        vector<vector<int>> a(10);
        ifstream fin(str);
        int num;
        if (!fin)
            throw FileNotFoundException(str);
        // 입력받은 파일이름에 맞는 파일을 읽어와 vector로 입력 후, 출력 구현
        for (int i = 0; i < 10; i++) {
            for (int j = 0; j < 10; j++) {
                fin >> num;
                a[i].push_back(num);
            }
        }
        int b,c;
        cout << "출력할 행 크기 : ";
        cin >> b;
        cout << "출력할 열 크기 : ";
        cin >> c;
        for (int i = 0; i < b; i++) {
            for (int j = 0; j < c; j++) {
                cout << a.at(i).at(j) << " ";
            }
            cout << endl;
        }
    }
    catch (exception& e) {
        cout << endl << e.what() << '\n';
    }
    return 0;
}

```

Microsoft Visual Studio 디버그

파일 이름 : temp.txt
출력할 행 크기 : 8
출력할 열 크기 : 5

57	31	62	72	39
17	29	50	76	35
95	62	39	91	25
8	76	66	13	74
79	0	16	60	87
66	28	43	31	5
27	49	4	51	15
76	11	0	82	75

Microsoft Visual Studio 디버그 ×

파일 이름 : temp

File "temp" not found

Microsoft Visual Studio 디버그 × +

파일 이름 : temp.txt
출력할 행 크기 : 5
출력할 열 크기 : 12

59	9	50	58	51	81	71	78	48	87
----	---	----	----	----	----	----	----	----	----

invalid vector subscript