1. Exercise 14

StudentRecord에 대하여 멤버 길이 오프셋(offset) 테이블을 만들어라

정수 자료형(int)의 크기는 4byte, 실수 자료형(float)의 크기는 4byte로 계산

```
typedef char String[9];
struct StudentRecord
{
        String firstName;
        String lastName;
        int id;
        float gpa;
        // char gender;
        int currentHours;
        int totalHours;
};
StudentRecord student;
StudentRecord students[100];
```

	Length	Offset
First name	9	0
Last name	9	9
Id	4	20 = 18 + 2
gpa	4	24 = 22 + 2
currentHours	4	28 = 26 + 2
totalHours	4	32 = 30 + 2

이론상 34 byte가 될 줄 알았다. 하지만 디버깅을 해보니 36byte였다. Visual studio로 진행했다.

```
    *(student.firstName) = 'a';
    *(student.lastName) = 'b';
    student.id = 1;
    student.gpa = 1.;
    student.currentHours = 1;
    student.totalHours = 3;
```

임의의 값을 대입한 후, 디버깅을 해본 결과 36byte였다. 9byte씩 18까지 추가 된 후, 2bye가 추가로 들어갔다. 4byte의 배수로 크기를 맞춰주기 위해, 2byte가 추가로 들어간 것 같다.

```
0x008CF900  61 cc cc cc cc cc cc cc  a???????
0x008CF908  cc 62 cc cc cc cc cc cc  ?b??????
0x008CF910  cc cc cc cc 01 00 00 00  ????....
0x008CF918  00 00 80 3f 01 00 00 00  ...?....
0x008CF920  03 00 00 00
```

2. Exercise 15

만약 student의 기본 주소가 100이면, student.gpa의 주소는 무엇인가?

124이다.

window에서 컴파일 할 시, 중간에 2byte가 추가로 들어가기 때문이다.

3. Exercise 16

students 에 대하여 컴파일러는 얼마나 많은 메모리 공간을 준비하는가?

3600byte를 차지한다.

4. Exercise 28

a. ADT SquareMatrix에 대한 규격 명세를 작성하라.

(정사각형 행렬(square matrix)은 N행과 N열을 가진 2차원 배열에 의해 표현될 수 있다.) 최대 크기로 50개의 행과 열을 가정할 수도 있다. 다음의 연산들을 포함하여라.

MakeEmpty(n) : 모든 n행과 열들을 0으로 설정한다.

StoreValue(i, j, value) : value를 i번째행, j번째 열의 위치에 저장한다.

Add : 두 행렬을 함께 더한다.

Subtract : 한 행렬을 다른 행렬로부터 뺀다.

Copy : 한 행렬을 다른 행렬로 복사한다

- **구조**

N * N의 정사각 행렬. Composite data type. Structed data type임. 즉 같은 type의 2차원 배열로 생각.

- **연산**

MakeEmpty(n)

기능: Matrix의 n 행 열 내부를 0으로 초기화 동적 할당을 쓸 것

조건: N의 최대 크기는 50

결과: N 안의 행 열은 0으로 초기화

StoreValue(i, j, value)

기능: i+1행 j+1열의 값을 value로 바꿈

조건: i와 j는 0부터 49까지 자연수

결과: i+1행 j+1열의 값이 바뀜

Add(Matrix)

기능: 크기가 같은 두 정사각 행렬을 더함

조건: 두 행렬의 크기가 같아야 함.

결과: 각 요소별로 합이 계산됨

Subtract (Matrix)

기능: 크기가 같은 두 정사각 행렬을 뺌.

조건: 두 행렬의 크기가 같아야 함.

결과: 각 요소별로 차가 계산됨

Copy(Matrix)

기능: deep copy를 진행함.

조건: operand가 정사각 행렬 matrix이어야 함.

결과: 새로운 행렬의 reference를 return함.

b. 위의 규격 명세를 c++ 클래스 선언으로 변화하라

exercise_28.h 에 declare 해 놨습니다.

c. 멤버 함수들을 구현하라

exercise_28.cpp
exercise_28_test.cpp

5. Exercise 29

다음의 규격 명세의 ComparedTo 함수를 추가하여 StrType 을 향상시켜라.

RelationType ComparedTo(StrType& otherString)
기능: self와 otherString을 알파벳 순으로 비교한다.
조건 : self와 otherString은 초기화되었다.
결과 : 함수 값 = LESS, 만약 self가 otherString 앞에 온다면
= GREATER, 만약 self가 otherString 뒤에 온다면
= EQUAL, 만약 self 가 otherString 과 같다면

a. 에 있는 strcmp 를 사용하여 멤버함수 ComparedTo 를 작성하라.

Q. ADT 에 예외 상황이 있는 것 같음. self 와 otherString 을 비교해서 self 가 other 의 앞에 오지 않고, 뒤에 오지 않고, 같지도 않은, 애초에 다른 문자이면 어떻할지 모르겠음.

// Implementation

// exercise_29.cpp 에 주석 부분에 있음

```
RelationType StrType::ComparedTo(StrType& otherString)
{
        if (strcmp(letters, otherString.letters) == 0)
                return EQUAL;
        else if (strcmp(letters, otherString.letters) > 0)
                return GREATER;
        else
                return LESS;
}
```

b. strcmp 라이브러리 함수를 사용하지 않고, 멤버 함수 ComparedTo 를 다시 작성하라

// exercise_29.cpp