

1. 아래와 같이 학생(Student) 클래스를 입력하고, 상속 받을 학부생 클래스(Undergraduate)와 조교 클래스(Graduate)를 만든 후에 정보를 출력해주세요.

- 1) 학생의 기본정보는 이름, 학번, 과, 학년, 이수점수가 있습니다.
- 2) 학부생의 정보는 추가적으로 소속 동아리명이 있습니다.
- 3) 조교의 경우에는 조교유형과 장학비율이 있으며, 조교 유형의 경우 연구조교와 교육조교로 나뉩니다.
- 4) Base Class인 Student는 print 함수 사용 시 "I'm student"를 출력합니다.

```
int main(void)
{
    Student Std;
    Graduate Kang("강석원", 2018311025, "컴퓨터공학과", 1, 18, false, 0.5);
    UnderGraduate Hong("홍길동", 20141111, "컴퓨터공학과", 2, 80, "알고리즘동아리");

    Std.print();
    Kang.print();
    Hong.print();

    return 0;
}
```

[참조]

```
class Student {
```

```
private:
```

```
    string name;           //이름
    int id;                 //학번
    string department;     //과
    int grade;             //학년
    int credit;            //이수학점
```

```
public:
```

```
    Student(string n = "default", unsigned int i = 0, string d = "default", int g = 0,
int c = 0) : name(n), id(i), department(d), grade(g), credit(c) {}
```

```
    /* Getter, Setter 함수를 구현해주세요 */
    /* 또한 virtual을 활용해서 print함수를 만들어주세요 */
```

```
};
```

1 - 출력화면

```
C:\> 선택 C:\WINDOWS\system32\cmd.exe

I'm student

=====
이름      : 강석원
학번      : 2018311025
소속학과  : 컴퓨터공학과
학년      : 1
이수학점  : 18
조교유형  : 연구조교
장학비율  : 0.5
=====

이름      : 홍길동
학번      : 20141111
소속학과  : 컴퓨터공학과
학년      : 2
이수학점  : 80
동아리명  : 알고리즘동아리
=====
계속하려면 아무 키나 누르십시오 . . .
```

2. 애너그램을 만들기위해 삭제해야할 최소 문자수를 출력하세요.

두 영단어의 순서를 바꾸어서 같아질 수 있다면, 두 단어는 애너그램 관계입니다. 예를들면 apple은 eplpa는 서로 애너그램 관계에 있습니다. 반대로 apple과 elpaa는 애너그램 관계가 될 수 없습니다.

그렇다면, 두 개의 영단어가 주어졌을 때, 두 단어가 애너그램이 되도록 삭제해야할 문자수를 카운트해서 출력해주세요.

2 - 출력 예시

```
cmd 실행 C:\WINDOWS\system32\cmd.exe
두 단어를 입력해주세요 : XXYYBB AABBC
8
```

- 여기서는 X,X,Y,Y,A,A,C,C 총 8개의 문자가 삭제되어야 두 단어가 애너그램(BB)이 될 수 있습니다.

```
cmd 실행 C:\WINDOWS\system32\cmd.exe
두 단어를 입력해주세요 : AABBCDD ABCDE
5
```

3. 숫자 야구를 구현해주세요.

- 1) 컴퓨터는 정답으로 쓰일 1 ~ 9 범위 내 세 개의 숫자를 임의로 선택합니다.
- 2) 유저는 매 턴 숫자 세 개를 입력합니다.
- 3) 컴퓨터는 유저가 입력한 값에서 숫자는 맞지만 위치가 틀릴 경우에는 Ball을, 위치도 맞고 숫자도 맞을 경우는 Strike를 카운트하여 출력합니다.
- 4) Strike와 Ball이 각각 0일 경우에는 Out을 출력합니다.
- 4) 9번의 시도에도 정답을 맞히지 못하는 경우 패배합니다.
- 5) 정답과 입력은 중복된 값을 사용할 수 없으며 1 ~ 9 범위 내에서 정해야합니다.

```
int main() {  
    int input[3];  
    int answer[3];  
  
    /* 1 ~ 9 범위 내 임의의 숫자 세개를 선정*/  
  
    startGame(input, answer);  
  
    return 0;  
}
```

3 - 출력예시

```
CA. 실행 C:\WINDOWS\system32\cmd.exe  
===== 1 =====  
1 ~ 9 범위의 숫자 세개를 입력해주세요 : 1 2 3  
0 Strike, 1 Ball  
  
===== 2 =====  
1 ~ 9 범위의 숫자 세개를 입력해주세요 : 4 5 6  
Out!!  
  
===== 3 =====  
1 ~ 9 범위의 숫자 세개를 입력해주세요 : 7 8 9  
0 Strike, 2 Ball  
  
===== 4 =====  
1 ~ 9 범위의 숫자 세개를 입력해주세요 : 7 2 9  
0 Strike, 2 Ball  
  
===== 5 =====  
1 ~ 9 범위의 숫자 세개를 입력해주세요 : 7 3 9  
1 Strike, 2 Ball  
  
===== 6 =====  
1 ~ 9 범위의 숫자 세개를 입력해주세요 : 9 3 7  
3 Strike, 0 Ball  
  
정답입니다!  
계속하려면 아무 키나 누르십시오 . . .
```

```
===== 9 =====  
1 ~ 9 범위의 숫자 세개를 입력해주세요 : 1 2 3  
0 Strike, 1 Ball  
  
패배했습니다.  
정답은 4 3 7 입니다.  
계속하려면 아무 키나 누르십시오 . . .
```

```
CA: 실행 C:\WINDOWS\system32\cmd.exe  
===== 1 =====  
1 ~ 9 범위의 숫자 세개를 입력해주세요 : 1 1 1  
중복된 숫자를 입력하셨습니다.  
1 ~ 9 범위의 숫자 세개를 입력해주세요 : 10 10 10  
1 ~ 9 범위 내의 숫자를 입력하여 주세요.
```

4. 번 [사진관리 프로그램 작성을 위한 단계 4]

lab#8에서 작성한 프로그램을 다음과 같이 확장한다.

** 아래 기능을 구현하기 위한 class들의 ADT는 [참조 1]과 같다. 이를 참조하면 쉽게 class를 설계하고 멤버함수들을 구현할 수 있을 것임.

1. Photo name을 자동으로 생성하도록 수정한다.
 - RecordType class의 ReadRecordFromKB()함수를 수정한다.
2. Photo name 순으로 photo list정렬한다.
 - Application class의 AddRecordToList()를 photo name 순으로 저장할 수 있도록 수정한다.
3. Application class에서 Binary Search를 이용하여 photo name검색을 수행하는 BinarySearchByPrimaryKey()함수를 구현한다.
4. 이벤트 이름을 입력받아서 해당 이벤트에서 찍은 모든 사진을 출력하는 기능을 구현한다.
5. 콘텐츠(contents)에 하나 이상의 단어를 포함하도록 다음 함수를 수정한다.
 - RecordType::ReadRecordFromKB()
 - RecordType:: ReadRecordFromFile()
6. 콘텐츠를 나타내는 단어(예: 홍길동)을 입력받아서 해당 단어를 포함하는 모든 사진을 찾아서 출력하는 함수를 구현한다.

** 위 기능들의 실행에는 [참조 2]와 같을 수 있다.

[참조 1]

< ApplicationType class>

```
#include "PhotoRecordType.h"
using namespace std;

class ApplicationType
{
public:
    ApplicationType(){ photoList.clear(); length=0; inFileName="test.txt";
    outFileName="test.txt";}; // initialize input and output file name
    int AddRecordToList(); // read a record from keyboard and add to the list
    int ReadAllFromFile(); // read all the records in file and store in vector
arrays
    void DisplayAllOnScreen(); // write all the record in the list on the screen
    int WriteAllToFile(); // write all photo records to disk.
    int GetCommand(); // get an integer representing a proper command
    void Run(); // select a command and execute
    int SearchByPrimaryKey(string inName); // Search a record by file name and return
index of the record.
    // Search a record by file name and return index of the record.
    int BinarySearchByPrimaryKey(string inName); // Search a record by file name and
return index of the record using binary search.
    // Search a record by file name and return index of the record.
    int RetrieveByPhotoName(); // find a record by name and display
    int DeleteByPhotoName(); // find a record by name and delete
    int RetrieveByEvent(); // find records by event
    int RetrieveByContents(); // find records by contents
private:
    vector<RecordType> photoList;
    int length;
    string inFileName;
    string outFileName;
};
```

< RecordType class>

```
using namespace std;

class RecordType {
public:
    int ReadRecordFromKB(); // read a record from keyboard
    int ReadPhotoNameFromKB(); // read photo name from keyboard
    int ReadEventNameFromKB(); // read event name from Keyboard
    int ReadRecordFromFile(ifstream& inFile); // read a record from file
    void WriteRecordToFile(ofstream& outFile); // write a record to file
    void GenFileName(string& name); //generate an unique file name using
the current time.
    void DisplayOnScreen(); // display a record on screen
    bool IsEqual(string name); // compare photoName with input string
    bool IsEqualEvent(string name); /// compare event name
    bool IsContentsIncluded(string name); /// compare event name
    int Compare(string nameKey);
    bool operator<(RecordType inItem); // operator overloading
    bool operator>(RecordType inItem); // operator overloading
private:
    string photoName;
    string eventName;
    int size;
    string contents;
};
```

[참조 2] 실행 스크린

```

--- ID ----- Command -----
| 1. Add a new record to list |
| 2. Read all records from disk |
| 3. Write all records to disk |
| 4. Display all record on screen |
| 5. Retrieve by photo name |
| 6. Delete by photo name |
| 7. Retrieve by event |
| 8. Retrieve by contents |
| 0. Quit |
|-----|

Choose a Command--> 4

***** [ Display on screen ] *****
Record 0 : 20180930160637 . 2016생일 . 23 . 이성규 김산 경복궁
Record 1 : 20180930160715 . 2016생일 . 32 . 김산 조영규 경복궁
Record 2 : 20180930160946 . 2018미국여행 . 23 . 이성규 정석 요세미티
Record 3 : aaa . 2017졸업식 . 33 . 이성규 김산 경희대국제캠퍼스
Record 4 : ddd . 2017졸업식 . 23 . 김강식 이성규 장일성 경희대국제캠퍼스
Record 5 : ffff . 2018미국여행 . 33 . 이성규 김산 금문교

--- ID ----- Command -----
| 1. Add a new record to list |
| 2. Read all records from disk |
| 3. Write all records to disk |
| 4. Display all record on screen |
| 5. Retrieve by photo name |
| 6. Delete by photo name |
| 7. Retrieve by event |
| 8. Retrieve by contents |
| 0. Quit |
|-----|

Choose a Command--> 7

***** [ Find by event name ] *****
Enter a Event Name --> 2018미국여행
Record 1 : 20180930160946 . 2018미국여행 . 23 . 이성규 정석 요세미티
Record 2 : ffff . 2018미국여행 . 33 . 이성규 김산 금문교

--- ID ----- Command -----
| 1. Add a new record to list |
| 2. Read all records from disk |
| 3. Write all records to disk |
| 4. Display all record on screen |
| 5. Retrieve by photo name |
| 6. Delete by photo name |
| 7. Retrieve by event |
| 8. Retrieve by contents |
| 0. Quit |
|-----|

Choose a Command--> 8

***** [ Find by contents ] *****
Enter a key in contents --> 경희대국제캠퍼스
Record 1 : aaa . 2017졸업식 . 33 . 이성규 김산 경희대국제캠퍼스
Record 2 : ddd . 2017졸업식 . 23 . 김강식 이성규 장일성 경희대국제캠퍼스

```