1. **[클래스 일반] 카메라 업계의 큰손인 KHUNON에서는 디지털 카메라 내에서 찍은 사진을 관리하는 SW를 전면 업그레이드 하기로 했다. 카메라의 사진 관리를 위한 다음 클래스를 구현하라. [25점]**
   1. KNPicManager 클래스
      1. 사진 및 동영상 관리 클래스. 여기에는 다수의 미디어 파일인 “picture” 와 “movie” 객체를 저장할 수 있어야 함
      2. 각 미디어 파일의 용량을 합산하여 현재 메모리 사용 량을 계산할 수 있어야 함
      3. 미디어의 추가 기능이 있어야 함
      4. 사진 아이디를 자동으로 부여하기 위해 사진 및 동영상의 개수를 저장하는 변수
   2. KNPicture 클래스
      1. 사진 아이디 (int), 사진 해상도 (int [2]), 사진 컬러 깊이 (int), 사진 날짜 (char[7])저장
      2. 실제 사진 데이터를 저장하는 char\* data; (사진의 가로해상도\*세로해상도\*컬러깊이 만큼의 메모리 공간을 할당해야 함. 메모리 상의 내용은 상관없음.)
      3. 이 사진의 용량을 계산해 주는 함수 (용량은 사진의 가로해상도\*세로해상도\*컬러깊이 가 됨)
   3. KNMovie 클래스
      1. 동영상 아이디 (int), 동영상 해상도 (int [2]), 동영상 컬러깊이 (int), 동영상 길이 (int), 동영상 날짜 (char[7])저장
      2. 실제 동영상 데이터를 저장하는 char\* data; (동영상 가로 해상도\*세로해상도\*컬러깊이\*동영상 길이 만큼의 메모리 공간을 할당해야 함. 이도 메모리 상의 내용은 상관없음.)
      3. 이 동영상의 용량을 계산해 주는 함수 (용량은 사진의 가로해상도\*세로해상도\*컬러깊이\*동영상 길이가 됨)
   4. 다음 Main 이 동작하게 위 클래스들의 내용을 적절히 만들어라. 필요하다면 맴버함수, 변수추가 가능
   5. KNPicture 와 KNMovie 는 공통적인 맴버변수와 맴버함수 몇 가지를 가진다. “상속” 이라는 기능으로 코드를 좀 더 효율적으로 만들 수 있다. (힌트: 공통된 부분을 묶어주는 공통 부모 클래스인 KNMedia 클래스). 이를 수행하라. 그리고, 아래 main 코드를 분석해 보면 이 상속관계는 단순 상속관계가 아니고, KNMedia는 “순수가상함수”를 가지는 “추상클래스”로 만들어야 할 것 같다. 그렇게 만들어라. (힌트: 공통으로 가지는 함수는 순수가상함수로 만들기)
   6. 또 다른 힌트 힌트: KNPicManager 에서 KNPicture와 KNMovie 리스트를 따로 관리하지 않고 다형성을 이용해 KNMedia의 리스트로 관리하면 아래 Main 가능!! 다형성을 적용하려면 리스트는 KNMedia \*의 배열이 되어야 함 (Media의 수는 100개라고 가정하던지, STL의 Vector 를 사용함.)
   7. 다음 Main 이 동작하게 만들어라.

KNManager myCamera;

KNMedia \* temp = new KNPicture(640, 480, 8, “070614”); //가로해상도, 세로해상도, 컬러깊이, 만든날짜

myCamera.addDriver(temp); //temp 가 가리키는 사진 혹은 동영상 추가

… //최소 3개 이상의 그림 추가

temp = new KNMovie(320,240,4,50,”070615”); //가로해상도, 세로해상도, 컬러깊이, 동영상 길이, 만든날짜

myCamera.addDriver(temp); //temp 가 가리키는 사진 혹은 동영상 추가

cout << “메모리 사용량 “ << myCamera.calculateMemoryUsage() << endl; //모든 Media 객체를 방문하면서 용량을 계산해 주는 함수를 호출하여 각 Media의 용량을 계산하고, 이를 합산하여 출력해 주는 함수.

myCamera.displayAllMedia(); //모든 Media들에 대한 정보를 다음과 같이 화면에 출력.

// ---사진 1번의 정보--- ---동영상 3번의 정보---

// 해상도: 640 x 480 해상도: 320 x 240

// 컬러깊이: 8 컬러깊이: 4

// 070614에 찍음 동영상길이: 50

// 데이터: x <- 첫 한글자만 찍음 070615에 찍음

// 데이터: g

1. **[탬플릿] 2번 문제에서 사진 및 동영상의 해상도를 표현할 때 int[2]를 사용하였다. 이를 좀 더 일반적인 형태로 표현하기 위해, 새롭게 2차원 백터를 표현할 수 있는 KNVector2d 클래스 탬플릿를 구현하고, 이를 사용하게 2번 문제를 수정하여라. KNVector2d 는 클래스 탬플릿으로, 2차원 int 백터 뿐만 아니고 다른 형의 벡터도 만들어 낼 수 있게 구현되어야 한다. [17점]**
2. **위 1번 예제의 사진 하나를 파일에 저장하고 싶다. 이를 수행하라. [18점]**
   1. 메인 함수에서는 KNPicture 객체 하나를 생성 (해상도는 4 x 4, 컬러깊이는 2, 따라서 총 용량은 32바이트).
   2. 32바이트의 char가 할당된 “data”에 아스키코드 65 부터 96까지 순서대로 저장.
   3. 이 객체를 아래와 같이 화면에 출력함. 이때는 data가 가리키는 문자열 모두를 출력해야 함.  
      ---사진 1번의 정보---  
      해상도: 4 x 4  
      컬러깊이: 2  
      070707에 찍음  
      데이터: ABCDEFGHIJKLMNOPQRSTUVWXYZ[/]^\_’
   4. 이 객체를 바이너리로 파일에 출력
   5. 이때, KNPicture는 포인터를 멤버변수로 가지고 있어서 이를 그냥 바이너리로 저장하면 파일에 data의 포인터 주소값만 저장이 된다. 따라서, 완벽한 파일 출력을 위해서는 객체의 저장 외 추가로 data가 가리키는 곳의 내용물을 파일에 별도 저장해야 함.
   6. 이를 수행하고, 저장된 파일의 내용물을 다시 읽어드리고 화면에 출력하여 제대로 출력, 입력이 되었는지 비교하라.
3. **[파일 입출력 및 스트링] 실습 14의 문제 8번은 텍스트를 모스부호로 인코딩하고 반대로 디코딩하는 문제였다. 여기에 다음 항목들을 추가 구현해 이를 확장하라. (부분점수 있으니 푼 부분만 확실히 동작하게 구현)**
   1. 메뉴의 인코딩을 누르면 인코딩 될 텍스트를 사용자로부터 입력을 받는 것이 아니라 파일로부터 텍스트를 읽어(파일이름을 사용자에게 받음) 이를 디코딩 하게 수정. 또한 인코딩을 완료하면 이를 화면에도 출력하고 “파일이름\_encoded.txt” 라는 새로운 파일에도 한번 더 출력하는 구문 추가. 택스트파일에서 엔터키가 들어올 수가 있는데 이는 “$$$#”로 인코딩 되게 추가.
   2. 디코딩 메뉴도 인코딩 메뉴와 비슷하게 수정
   3. Reverse 메뉴 추가. 이 메뉴를 누르고 파일명을 입력하면, 그 파일을 읽어 단어를 반대 순서로 재배치. 일반 텍스트와 모스 부호 모두 동작해야 함.
   4. 문제 7번에서는 대칭문자를 이용한 문자열 암호화를 수행하였음. 문제 7번과 8번을 연결하여, 문자열을 파일로부터 입력 받으면 이를 우선 암호화 하고, 암호화 된 문자열을 다시 모스 부호로 Encoding 하는 메뉴 Encrypted Encoding 추가.
      1. 모스부호표에 없는 기호들을 제외하기 위해 다음과 같이 변형된 대칭 문자열을 사용하라.
      2. String strEnglish = ＂ABCDEFGHIJKLMNOPQRSTUVWXYZ”;
      3. String strCode = ＂QWEMRNTBXYUVICOPZASDLFKGJH“;
   5. 반대의 경우를 수행하는 Encrypted Decoding 도 추가.
4. **[클래스 일반] 경희대학교에서 새로운 학사 및 수업 관리 시스템을 개발하려고 한다. 이에 대한 클래스 디자인을 수행하라. [20점]**
   1. 컴파일/실행되는 프로그램을 만들 필요는 없고, 클래스 껍데기 코드만 (헤더파일) 구현.
   2. 학생, 교강사, 수업, 강의실, 점수, 과제, 시험, 온라인 게시판 등등의 관련 클래스들에 대해, 이 클래스들이 어떻게 연결되어 있어야 하고 (상속, 집합, 함수호출 등) 어떤 함수 및 데이터 들이 필요한 지에 대한 디자인임. (위 클래스 외에 자유롭게 추가 가능, 예, 교강사는 강사, 교수, 조교를 자식으로, 수업은 실습수업, 이론수업을 자식으로, 사람(학생, 교강사)을 담당하는 인사관리 클래스를 추가하기 등등)
   3. 생각할 수 있는 모든 기능을 다 제공하게 아주 넓고 좀 추상적으로 디자인을 수행해도 되고, 특정 기능을 아주 디테일하게 디자인 해도 됨. 후자에서는 시스템이 제공하는 기능에 대한 정의를 명확하게 해야 함.
   4. 각각의 디자인의 타당성에 대한 설명을 코멘트로 추가하여야 함.