**2015, 봄 학기, 고급 객체지향 프로그래밍 시험 1**

1. **다음 세 함수를 구현하라. (25점)**
   1. 정수형 변수 3개를 입력 파라미터로 가지며, 정수형 트리플포인터를 반환하는 함수 memAlloc 함수를 구현한다. 함수의 프로토타입은 다음과 같다.  
      int\*\*\* memAlloc(int x, int y, int z);  
      memAlloc 함수는 입력받은 세 파라미터 값을 바탕으로 x\*y\*z크기의 3차원 배열을 동적으로 할당받고, 최종적으로 할당받은 정수형 배열의 각 원소값을 해당 원소의 x\*100+y\*10+z로 초기화 한다. (ex. (0, 0, 0)위치의 원소 값은 0, (2, 4, 1)위치의 원소값은 241)
   2. memAlloc 함수로 할당받은 3차원 배열의 포인터와 그 배열의 x, y, z축 크기를 입력 파라미터로 가지며, 모든 원소값의 평균을 반환하는 calcAverage 함수를 구현한다. 함수의 프로토타입은 다음과 같다.  
      float calcAverage(int\*\*\* ptr, int x, int y, int z);  
      이 함수에서 3차원 배열의 각 원소값에 접근할 때에는 포인터 오프셋 연산을 통해 접근해야만 한다.
   3. memAlloc 함수로 할당받은 3차원 배열의 포인터와 그 배열의 x, y 축을 입력 파라미터로 가지며, 반환값 없이 3차원 배열을 release 시키는 memRelease 함수를 구현한다. 함수의 프로토타입은 다음과 같다.  
      void memRelease(int\*\*\* ptr, int x, int y);
   4. 아래의 main함수가 정상적으로 동작되어 수행결과 출력 예상과 동일하게 나오도록 위 세 함수를 구현하라.

**int main() {**

**int\*\*\* Tptr = memAlloc(2, 3, 4);**

**cout << "Average value: " << calcAverage(Tptr, 2, 3, 4) << endl;**

**memRelease(Tptr, 2, 3);**

**return 0;**

**}**

* 1. <예상 출력 결과>  
     Average value: 61.5

1. **Bell triangle을 double pointer를 이용한 불규칙 배열로 구성하라. 높이가 7인 Bell Triangle은 다음과 같다. (25점)**

1

1 2

2 3 5

5 7 10 15

15 20 27 37 52

52 67 87 114 151 203

203 255 322 409 523 674 877

Bell triangle의 각 원소의 값은 왼쪽 원소의 값과 왼쪽 위 원소의 값의 합과 같다. 단, 각 행의 첫 번째 원소의 값은 윗 행의 마지막 열에 있는 원소의 값과 같다. 또한 가장 위의 원소 값은 1로 시작한다. 프로그램은 먼저 사용자로부터 높이를 입력받고, 입력 받은 높이에 맞게 불규칙한 배열 구조를 동적 메모리 할당으로 만든다.

**Height: 5**

**1**

**1 2**

**2 3 5**

**5 7 10 15**

**15 20 27 37 52**

배열의 구조를 만들 때 인덱싱을 위한 int형 변수 외에는 어떠한 다른 변수도 새롭게 만들지 않아야 하고, pp와 포인터연산과 간접 연산자(\*) 만을 사용해서 모든 부분을 구현해야 한다. 배열 구조를 만든 후, 배열 안의 내용물을 채워 넣는 구문을 구현하며, 이 때도 마찬가지로 pp와 포인터 연산과 간접연산자(\*) 만을 사용하여 구현한다. 배열 구조를 만든 후 위 배열을 적절한 구조로 출력한다. 출력시는 pp와 배열 첨자연산자([ ])를 사용해야 하며 옆은 출력 결과의 예이다.

1. **에어컨을 관리해 주는 클래스를 구현하라. (25점)**
   1. AirCon 클래스의 멤버 변수

**int main() {  
 AirCon aircon1(20,1,false);**

**AirCon aircon2(aircon1);**

**aircon1.setTargetTemp(24);**

**aircon1.setWindStrength(2);**

**aircon1.printCurrentStatus();**

**Aircon2.turnOnTurbo();**

**aircon2.printCurrentStatus();**

**return 0;**

**}**

* + 1. targetTemp: 목표 온도
    2. windStrength: 바람 세기
    3. turboMode: 터보모드인지 아닌지 저장하는 bool형 변수
  1. AirCon 클래스의 멤버 함수
     1. 기본생성자: 온도 20, 바람세기 2, 터보모드 false 로 초기화
     2. 복사생성자
     3. 3가지 정보 모두를 받는 생성자.
     4. setTargetTemp, setWindStrength: targetTemp와 windStrength를 업데이트 해 주는 함수
     5. turnOnTurbo, turnOffTurbo: 터보모드를 on, off해주는 함수
     6. printCurrentStatus: 현재 에어컨 상황을 출력해 주는 함수
  2. 위 클래스를 구현한 다음 main 함수에서 다음과 같은 일을 수행하라. 예상 결과는 오른쪽과 같음  
      \* 클래스 구현 시 멤버 변수와 함수들을 private, public으로 적절히 구분해 구현해야 함.

**An air-conditioner created.**

**---Air-conditioner status---**

**Target Temperature: 24**

**Wind Strength: 2**

**Turbo Mode: 0**

**---Air-conditioner status---**

**Target Temperature: 20**

**Wind Strength: 1**

**Turbo Mode: 1**

1. **3에서 만든 AirCon 클래스를 수정하고 새롭게 Air 클래스를 구현하여 AirCon 을 시뮬레이션 해 볼 수 있게 만들어라. (25점)**
   1. Air 클래스는 현재 방 안의 공기상태를 표현하기 위한 클래스이다. 이 공기 클래스와 에어컨 클래스가 상호작용하여 방안의 온도변화를 시뮬레이션 해 볼 수 있다.
      1. Air 클래스의 멤버 변수
         1. Temperature: 현재 온도
      2. Air 클래스의 멤버 함수
         1. 기본생성자: 현재 온도를 30으로 초기화
         2. update: 에어컨에서 나오는 바람의 세기를 보고 현재 방안의 온도를 갱신해 주는 함수. -wind strength \* 0.2 만큼 온도가 바뀐다. 예를 들어 현재 온도가 20도이고 에어컨의 바람세기가 4이면 update함수는 현재 온도를 19.2로 바꾼다.
         3. getTemp: 현재 온도를 리턴해줌.
   2. AirCon 클래스에 다음을 추가로 구현하라
      1. 멤버함수
         1. controlStrength: 현재 방안 온도를 입력으로 받고 이를 바탕으로 에어컨의 바람 세기를 갱신해 주는 함수. 만약 현재 온도가 에어컨의 target 온도보다 작다면 바람세기는 0. 그것이 아니라면 둘의 차이만큼 바람세기를 갱신. 바람세기의 최대치는 10임.
   3. 위를 구현 한 다음 main 함수에서 다음과 같은 일을 수행하라. 메인함수의 무한반복문에서는 먼저 현재 온도를 갱신하고 (update함수) 갱신된 온도를 바탕으로 에어컨 바람세기를 조정하는 과정을 반복함.

**int main() {  
 AirCon AC1(18,1,false);**

**Air air1;**

**int answer;**

**while(1) {**

**air1.update(AC1.getWindStrength());**

**cout << “Current Temperature: “ << air1.getTemp() << endl;**

**AC1.controlStrength(air1.getTemp());**

**AC1.printCurrentStatus();**

**cout << “---Menu---\n”;**

**cout << “1: continue to next tick\n2: set target temperature\n3:quit\n”;**

**cout << “Your command? ”;**

**cin >> answer;**

**//채우기: 1이면 계속 반복, 2이면 사용자 입력으로 target temperature 변경하고 반복**

**//3이면 종료**

**}**

**return 0;**

**}**

**An air-conditioner created.**

**An air created.**

**Current Temperature: 29.8**

**---Air-conditioner status---**

**Target Temperature: 18**

**Wind Strength: 10**

**Turbo Mode: 0**

**---Menu---**

**1: continue to next tick**

**2: set target temperature**

**3:quit**

**Your command? 1**

**Current Temperature: 27.8**

**---Air-conditioner status---**

**Target Temperature: 18**

**Wind Strength: 9.8**

**Turbo Mode: 0**

**---Menu---**

**1: continue to next tick**

**2: set target temperature**

**3:quit**

**Your command? 1**

**.. .. ..**