**2014, 봄 학기, 고급 객체지향 프로그래밍 시험 1**

1. 다음 두 함수를 구현하라. (20점)
   1. 두 3차원 벡터, **a** = (*a*1, *a*2, *a*3)와 **b** = (*b*1, *b*2, *b*3) 에 대해서 두 벡터의 외적(Cross Product) **a**×**b**는 두 벡터에 각각 수직이며 크기가 |**a**||**b**|sin*θ*인 또 다른 벡터가 된다. 이 외적을 산술적으로는 다음과 같이 구한다.

**a**×**b**=(*a*2*b*3−*a*3*b*2,*a*3*b*1−*a*1*b*3,*a*1*b*2−*a*2*b*1)

두 벡터가 주어졌을 때 이들의 외적을 구하는 함수를 구현하라. 이 함수의 prototype는 다음과 같이 구성하여야 한다.

**int \* crossProduct(int\* a, int\* b);**

여기서 입력 파라미터 a, b는 두 3차원 벡터로 크기가 3인 1차행렬이 된다. 이 함수는 내부적으로 새로운 벡터 하나(사이즈가 3인 1차행렬)를 생성하고 여기에 외적의 결과를 저장한 다음 이 행렬의 첫 번째 element를 가리키는 포인터를 리턴하여야 한다.   
힌트) 함수 안에서 dynamic memory allocation 사용

* 1. 행렬 에 대해 A의 전치행렬(transpose of A)은 AT로 나타내고 다음과 같이 정의한다.

http://matrix.skku.ac.kr/sglee/linear/ocu/img/2-1-15.gif

예)

3x3 2차원 행렬을 입력 파라미터로 받아서 이 행렬의 전치행렬을 파라미터로 반환해 주는 함수를 만들어라. 함수는 파라미터 2개를 가지는데, 첫 번째 파라미터는 입력 행렬이고 (3x3행렬), 두 번째 파라미터는 구해진 전치행렬을 저장하고 반환할 목적의 파라미터이다 (3x3 행렬). 이 함수의 리턴 값은 void이다.

**The cross product of**

**[6 7 2] and [1 2 9] is [59 -52 5]**

**The transpose of   
29 34 1  
20 45 43  
93 49 86  
is  
29 20 93  
34 45 49  
1 43 86**

위 두 함수를 만든 다음 옆과 같이 출력되게 적절히 main함수를 구성하라. A번의 두 입력 벡터는 1~10 사이의 임의의 정수로 구성하고 B번의 입력 행렬은 1~100 사이의 임의의 정수로 구성하라.

1. Pascal triangle을 double pointer를 이용한 불규칙 배열로 구성하라. 높이가 7인 Pascal triangle은 다음과 같다. (20점)

**1**

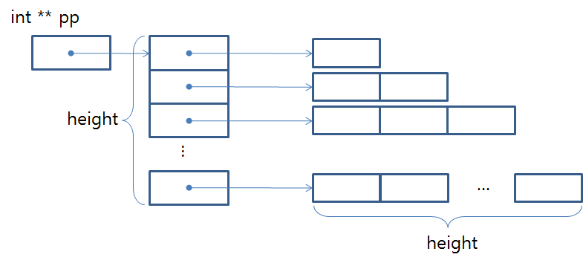
**1 1**

**1 2 1**

**1 3 3 1**

**1 4 6 4 1**

**1 5 10 10 5 1**

**1 6 15 20 15 6 1**

Pascal triangle의 각 원소는 바로 위 원소와 왼쪽 위 원소와의 합으로 나타낼 수 있다. 둘 중 하나가 없다면 없는 것은 0으로 가정한다. 가장 위 level의 첫 번째 원소는 1로 시작한다. 프로그램은 우선 사용자로부터 triangle의 높이를 입력 받는다. 그런 다음, dynamic memory allocation을 사용하여 옆 그림과 같은 불규칙한 배열 구조를 만든다.

배열 구조를 만들 때 인덱싱을 위한 int형 변수 외에는 어떠한 다른 변수도 새롭게 만들지 않아야 하고 pp와 포인터연산과 간접연산자(\*) 만을 사용해서 모든 부분을 구현해야 한다. 배열 구조를 만든 후 배열 안의 내용물을 채워 넣는 구문을 구현한다. 이 때도 마찬가지로 pp와 포인터연산과 간접연산자(\*) 만을 사용 모든 것을 구현한다. 배열구조를 만든 후 위 배열을 적절한 구조로 출력한다. 출력 시는 pp와 배열 첨자연산자([])를 사용하여야 한다. 옆은 출력 결과의 예이다.

**Height? 5**

**1**

**1 1**

**1 2 1**

**1 3 3 1**

**1 4 6 4 1**

1. 게임 서버의 사용자 관리를 위해 다음과 같이 사용자 클래스를 구현하려고 한다. (20점)
   1. User 클래스의 멤버 변수
      1. 사용자 이름을 저장할 수 있는 char array혹은 string 형 변수
      2. 사용자 보너스 점수를 저장하는 int형 변수
      3. 현재 접속유무를 저장하는 bool형 변수
   2. User 클래스의 멤버 함수
      1. 기본 생성자. 사용자 이름은 “anonymous”, 보너스 점수는 0점, 현재 접속 유무는 false 로 초기화
      2. 사용자 이름만 파라미터로 받는 생성자. 보너스 점수는 0점, 현재 접속 유무는 false 로 초기화
      3. 3가지 정보 모두를 받는 생성자.
      4. 다른 User 객체를 복사하는 복사 생성자
      5. 이름을 바꿔줄 수 있는 메소드 (changeName)
      6. 보너스 점수를 바꿔줄 수 있는 메소드 (changeBonusPoint)
      7. 접속유무를 바꿔줄 수 있는 메소드 (changeOnline)
      8. 사용자 정보를 프린트 해 주는 메소드 (printInfo)
   3. 위 클래스를 구현한 다음 main 함수에서 다음과 같은 일을 수행하라. 예상 결과는 오른쪽과 같음

**int main() {  
 User user1;  
 User user2(“SlayerS”,20,true);  
 User user3(user2);  
 user1.printInfo();**

**user2.printInfo();**

**user3.printInfo();**

**user1.changeName(“YellOw”);**

**user1.changeOnline(true);**

**user1.changeBonusPoint(50);**

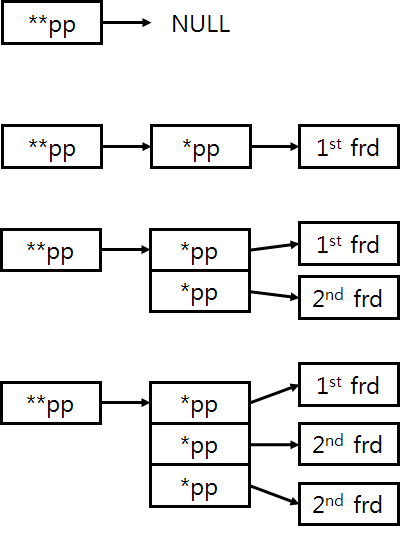
**user1.printInfo();**

**return 0;**

**}**

**Anonymous created.  
SlayerS created.  
SlayerS created.  
Anonymous has 0 bonus points and is now offline  
SlayerS has 20 bonus points and is now online  
SlayerS has 20 bonus points and is now online  
YellOw has 50 bonus points and is now online**

\* 클래스 구현 시 멤버 변수와 함수들을 private, public으로 적절히 구분해 구현해야 함. 그리고 사용자 이름 복사 및 할당 시 자기가 원하는 string 관리 방법 사용.

1. 3에서 만든 User 클래스를 확장하여 사용자가 서로 친구관계를 맺을 수 있게 하라. (20점)
   1. 다음의 멤버 변수 추가
      1. 친구 객체 리스트: 이 user와 친구 관계인 user들의 오브젝트 리스트. 기본적으로 친구인 User들을 가리키는 포인터들을 모아놓은 배열. 리스트의 크기가 가변적이며, 친구가 추가되면 ‘친구 수 \* 1’ 크기의 포인터 배열로 구성됨.
   2. 다음의 멤버 함수 수정 혹은 추가
      1. 모든 생성자에서 친구 리스트를 NULL 로 초기화
      2. 친구추가 메소드 (addFriend): 파라미터로 User 객체를 하나 받음. 자신과 파라미터로 받은 객체와 친구를 맺음. 자신의 친구 오브젝트 리스트를 갱신하고 (파라미터로 받은 오브젝트를 가리키는 포인터를 추가), 파라미터로 받은 오브젝트의 친구 리스트도 갱신해야 함. 리턴값은 없음. 친구가 추가되면 현재 친구 수 + 1 크기만큼의 새로운 리스트를 할당받아 기존의 리스트를 복사하고, 마지막 엘리먼트에 파라미터로 받아온 객체의 주소를 대입함 (오른쪽 그림 참조). 추가하려는 오브젝트가 이미 있는지 검사는 하지 않아도 됨.
      3. 사용자 친구 정보를 출력해 주는 함수.
   3. 위를 구현 한 다음 main 함수에서 다음과 같은 일을 수행하라. 예상결과는 오른쪽과 같음

**int main() {  
 User user1(“YellOw”,50,true);  
 User user2(“SlayerS”,20,true);  
 User user3(“GoGo”,30,false);**

**User user4;**

**user1.addFriend(user2);**

**user1.addFriend(user4);**

**user2.addFriend(user3);**

**user1.printFriends();**

**user2.printFriends();**

**user3.printFriends();**

**user4.printFriends();**

**return 0;**

**}**

**YellOw created.  
SlayerS created.  
GoGo created.**

**Anonymous created.**

**YellOw’s friends are SlayerS, Anonymous.**

**SlayerS’s friends are YellOw, GoGo.**

**GoGo’s friends are SlayerS.**

**Anonymous’s friends are YellOw.**

1. 4에서 만든 User 클래스를 활용하여 게임 서버 (GameServer)를 구현하라. (20점)
   1. GameServer 클래스의 멤버 변수
      1. 게임 이름
      2. 가입 회원 수
      3. User 오브젝트 리스트
   2. GameServer 클래스의 멤버 함수
      1. 기본 아규먼트를 파라메터로 가지는 기본 생성자:
         1. 기본 아규먼트
            1. 게임 이름 = “Anonymous”
         2. 초기화
            1. 가입 회원 수 = 0
            2. User 객체 리스트 = NULL
      2. 게임 이름을 입력받는 함수
      3. 회원 가입 함수
         1. 사용자 이름, 보너스 점수, 접속유무를 입력받아 User 객체 리스트에 추가 (4번의 리스트와 동일한 메커니즘으로 구현)
      4. 총 회원 정보 열람
         1. User 리스트에 있는 회원의 정보를 모두 출력
      5. 접속 회원 정보 열람
         1. User 리스트에 있는 회원 중 접속중인 회원의 정보 출력

**int main() {  
 GameServer Anypang(“Anypang”);  
 Anypang.addUser();  
 Anypang.addUser();**

**Anypang.printUser();  
 Anypang.printOnlineUser();**

**GameServer lol;**

**lol.getName();  
 lol.addUser();**

**lol.addUser();  
 lol.addUser();**

**lol.printUser();  
 lol.printOnlineUser();**

**return 0;**

**}**

**Anypang created.**

**Enter username, bonus points, and access(0: offline/1: online):**

**SlayerS 20 1**

**SlayerS created.**

**Enter username, bonus points, and access(0: offline/1: online):**

**GoGo 30 0  
GoGo created.**

**2 Users was joined Anypang:**

**SlayerS has 20 bonus points and is now online.**

**GoGo has 30 bonus points and is now offline.**

**Online users in Anypang:**

**SlayerS has 20 bonus points and is now online.**

**Anonymous created.**

**Enter game name:**

**lol**

**Enter username, bonus points, and access(0: offline/1: online):**

**JSH 100 1**

**JSH created.**

**Enter username, bonus points, and access(0: offline/1: online):**

**JBH 80 1**

**JBH created.**

**Enter username, bonus points, and access(0: offline/1: online):**

**KBS 50 0**

**KBS created.**

**2 Users was joined lol:**

**JSH has 100 bonus points and is now online.**

**JBH has 80 bonus points and is now offline.**

**KBS has 50 bonus points and is now offline.**

**Online users in lol:**

**JSH has 100 bonus points and is now online.**

**JBH has 80 bonus points and is now offline.**