**2017, 봄 학기, 객체지향 프로그래밍 시험 1**

1. **다음 함수들을 구현하라. (25점)**

두 10차원 벡터, **a** = (*a*1, …, *a*10)와 **b** = (*b*1, …, *b*10) 사이의 합, 차, 내적 연산을 수행하는 함수들을 구현하려고 한다. 두 10차원 벡터의 3가지 연산은 다음과 같이 계산한다.

**a-b**=(*a*1−*b*1,…, *a*10−*b*10)

**a+b**=(*a*1+*b*1,…, *a*10+*b*10)

**a·b**=*a*1*b*1+ *a*2*b*2,…,*a*10*b*10

참고로, 합과 차는 결과가 10차원 벡터로, 내적은 스칼라 (숫자 하나)로 나오는 것에 주의하라. 다음과 같은 함수의 Prototype 3가지를 사용하여 위 연산을 수행하는 함수 3가지를 각각 구현하라.

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

이 함수는 내부적으로 새로운 벡터 하나(사이즈가 10인 1차행렬)를 생성하고 여기에 합의 결과를 저장한 다음 이 벡터의 첫 번째 원소를 가리키는 포인터를 리턴하여야 한다. Dangling pointer 에 주의하라 (함수의 로컬 배열은 함수가 끝나면 없어짐)

**void vectorSubtract(int\* a, int\* b, int\* c);**

이 함수는 a, b 벡터의 차를 c가 가리키는 행렬에 저장하는 함수. c에 대한 메모리를 함수 외부에서 할당하던지, 내부에서 할당하던지 자유임.

**void vectorInnerProduct(int\* a, int\* b, int& c);**

이 함수는 a, b 벡터의 내적을 c에 저장하는 함수.

Main 함수에서는 자유롭게 a, b 벡터를 만들어 내용을 채우고 화면에 출력한 후, 위 3함수를 이용해 a, b사이의 3가지 연산을 수행하여 결과를 출력하라.

1. **불규칙 2차원 배열을 구현하라 (25점)**
   1. 실습 1의 6번 문제는 불규칙한 길이의 2차원 배열을 만드는 문제였다. 이를 활용하여 다음을 수행하라.
   2. 사용자에게 int형 숫자 하나 (lines)를 입력 받아라.
   3. lines 개수만큼 1~6까지의 랜덤 넘버를 생성하라.
   4. 세로 줄 수가 lines 이고, 각 줄의 가로의 길이가 이 랜덤넘버인 길이가 불규칙한 2차원 배열을 만들어라.
   5. 이 2차원 배열의 내용을 1~100사이의 랜덤수로 채워라.
   6. 각 줄에서 최대값을 찾아주는 lineMax함수를 구현하라. 이 함수는 위 2차원 배열의 한 줄(첫 번째 원소를 가리키는 포인터)과 그 줄이 길이를 파라미터로 받고, 최대값을 리턴 해 주는 함수이다.
   7. main에서는 lines 크기의 int형 배열 (maxValues)를 또 하나 만들고, 위 함수를 이용해서 구한 각 줄의 최대값으로 채워라.
   8. 최종적으로, 위에서 만든 불규칙 2차원 배열과 maxValues 를 화면에 적절히 출력하라.
   9. lines 가 4이고 각 줄의 길이가 2,3,4,1 이라면

2차원 배열

|  |  |  |  |
| --- | --- | --- | --- |
| 43 | 67 |  |  |
| 34 | 78 | 62 |  |
| 9 | 12 | 23 | 76 |
| 8 |  |  |  |

maxValues

|  |  |  |  |
| --- | --- | --- | --- |
| 67 | 78 | 76 | 8 |

1. **KHUALA라는 서점에서 책 관리를 위한 프로그램 의뢰가 들어왔다. 이를 위해 다음과 같이 책 클래스를 구현하라. (25점)**
   1. Book 클래스의 멤버 변수
      1. 책 이름을 저장할 수 있는 char \* 변수
      2. 책 가격을 저장하는 int형 변수
      3. 책 평점을 저장하는 int형 변수
      4. 현재까지 평점 개수를 저장하는 int 형 변수
      5. 현재 재고 개수를 저장하는 int형 변수
   2. Book 클래스의 멤버 함수
      1. 기본 생성자. 책 이름은 “anybook”, 평점 점수 및 개수 0, 가격은 100원, 현재 재고는 1로 초기화
      2. 책 이름만 파라미터로 받는 생성자. 평점 점수 및 개수 0, 가격은 100원, 현재 재고는 1로 초기화
      3. 5가지 정보 모두를 받는 생성자.
      4. 다른 Book 객체를 복사하는 복사 생성자
      5. 이름을 바꿔줄 수 있는 메소드 (changeName)
      6. 평점을 하나 추가해 주는 (addPoint). 이 함수는 평점 개수를 하나 추가하고 지금까지 평점 평균으로 평점을 갱신해 줘야 함
      7. 책 입고 메소드 (addBook) 재고 수 하나 증가
      8. 책 출고 메소드 (releaseBook) 재고 수 하나 감소, 0 미만이면 감소하지 말고 경고메시지 출력
      9. 책 정보를 프린트 해 주는 메소드 (printInfo)
   3. 위 클래스를 구현한 다음 main 함수에서 다음과 같은 일을 수행하라. 예상 결과는 오른쪽과 같음

**int main() {  
 Book book1;  
 Book book2(“Bible”,50000, 75, 2, 5);  
 Book book3(book2);  
 book1.printInfo();**

**book2.printInfo();**

**book3.printInfo();**

**book1.changeName(“Bible 2”);**

**book1.addPoint(50);**

**book1.addPoint(100);**

**book1.addBook();**

**book1.releaseBook();**

**book1.releaseBook();**

**book1.releaseBook();**

**book1.printInfo();**

**return 0;**

**}**

**Anybook 책 생성**

**Bible 책 생성  
Bible 책 생성  
Anybook:평점(0), 가격(100), 재고(1)**

**Bible:평점(75), 가격(50000), 재고(5)**

**Bible:평점(75), 가격(50000), 재고(5)**

**Anybook -> Bible 2로 이름 바뀜**

**Bible 2:평점(50) 추가, 평점평균(50)**

**Bible 2:평점(100) 추가, 평점평균(75)**

**Bible 2:재고 하나 추가**

**Bible 2:재고 하나 감소**

**Bible 2:재고 하나 감소**

**Bible 2:재고가 없어 출고 불가!!!**

**Bible 2:평점(75), 가격(100), 재고(0)**

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

1. **인터넷 전략 게임인 Star KHUraft 의 유닛중의 하나를 클래스로 구현하라. (수업시**
2. **간에 한 Class example의 4-2 를 참조하여 문제를 푸세요) (25점)**
   1. Star KHUraft 의 최강 유닛인 Battle\_KHUruiser 클래스를 구현하라.
   2. 다음 main 함수의 코드가 동작하도록 Battle\_KHUruiser 클래스를 적절히 구현하라.

**int main()**

**{**

**Battle\_KHUruiser BK1("BK1", 100, 30, 10, 40, 50, 10, 2); //이름, 에너지, 공격력, 방어막, x좌표, y좌표, 무기 사정거리, 이동속도**

**Battle\_KHUruiser BK2("BK2",200, 50, 0, 40, 63, 30, 1);**

**while (1) {**

**int order;**

**BK1.printStatus(); //결과창과 같이 모든 정보 출력**

**cout << "유닛" << BK1.GetName() << "에 대한 명령을 입력하시오\n" << "1:왼쪽이동, 2:오른쪽이동, 3:위쪽이동, 4:아래쪽이동, 5: 상대유닛 공격, 6: 종료 : ";**

**cin >> order;**

**switch (order) {**

**case 1:**

**BK1.moveLeft(); break; //x 좌표를 이동속도만큼 감소시킴**

**case 2:**

**BK1.moveRight(); break; //x 좌표를 이동속도만큼 증가시킴**

**case 3:**

**BK1.moveUp(); break; //y 좌표를 이동속도만큼 증가시킴**

**case 4:**

**BK1.moveDown(); break; //y 좌표를 이동속도만큼 감소시킴**

**case 5:**

**BK1.Attack(BK2); break;//만약 두 유닛사이의 거리가 사정거리보다 작으면 상대방 에너지를 공격력-상대방의방어력 만큼 감소시킴.**

**//만약 두 유닛사이의 거리가 사정거리보다 크면 공격실패 메시지만 출력**

**//두 유닛 사이의 거리는 sqrt((float)((x - enemy.x)\*(x - enemy.x)) + ((y - enemy.y)\*(y - enemy.y))) 로 구함**

**case 6:**

**return 0;**

**}**

**BK2.printStatus(); //여기서부터는 BK2 명령. 턴 제라서 한 턴(한 loop)에 BK1한번, BK2 한번씩 명령 수행**

**cout << "유닛" << BK2.GetName() << "에 대한 명령을 입력하시오\n" << "1:왼쪽이동, 2:오른쪽이동, 3:위쪽이동, 4:아래쪽이동, 5: 상대유닛 공격, 6: 종료 : ";**

**cin >> order;**

**switch (order) {**

**case 1:**

**BK2.moveLeft(); break;**

**case 2:**

**BK2.moveRight(); break;**

**case 3:**

**BK2.moveUp(); break;**

**case 4:**

**BK2.moveDown(); break;**

**case 5:**

**BK2.Attack(BK1); break;**

**case 6:**

**return 0;**

**}**

**}**

**return 0;**

**}**

아래는 결과 예시

**-----유닛BK1---------**

**위치:40,50**

**에너지, 공격력, 방어막, 사정거리, 이동속도: 100,30,10,10,2**

**유닛BK1에 대한 명령을 입력하시오**

**1:왼쪽이동, 2:오른쪽이동, 3:위쪽이동, 4:아래쪽이동, 5: 상대유닛 공격, 6: 종료 :5**

**사정거리가 짧아 공격 불가. 공격 유닛의 사정거리는 10인데 두 유닛 사이의 거리는 13임.**

**-----유닛BK2---------**

**위치:40,63**

**에너지, 공격력, 방어막, 사정거리, 이동속도: 200,50,0,30,1**

**유닛BK2에 대한 명령을 입력하시오**

**1:왼쪽이동, 2:오른쪽이동, 3:위쪽이동, 4:아래쪽이동, 5: 상대유닛 공격, 6: 종료 :5**

**BK1을40 만큼 공격하였다.**

**-----유닛BK1---------**

**위치:40,50**

**에너지, 공격력, 방어막, 사정거리, 이동속도: 60,30,10,10,2**

**유닛BK1에 대한 명령을 입력하시오**

**1:왼쪽이동, 2:오른쪽이동, 3:위쪽이동, 4:아래쪽이동, 5: 상대유닛 공격, 6: 종료 :3**

**BK1 이 40,52로 이동**

**-----유닛BK2---------**

**위치:40,63**

**에너지, 공격력, 방어막, 사정거리, 이동속도: 200,50,0,30,1**

**유닛BK2에 대한 명령을 입력하시오**

**1:왼쪽이동, 2:오른쪽이동, 3:위쪽이동, 4:아래쪽이동, 5: 상대유닛 공격, 6: 종료 :4**

**BK2 이 40,62로 이동**

**-----유닛BK1---------**

**위치:40,52**

**에너지, 공격력, 방어막, 사정거리, 이동속도: 60,30,10,10,2**

**유닛BK1에 대한 명령을 입력하시오**

**1:왼쪽이동, 2:오른쪽이동, 3:위쪽이동, 4:아래쪽이동, 5: 상대유닛 공격, 6: 종료 :5**

**BK2을30 만큼 공격하였다.**

**-----유닛BK2---------**

**위치:40,62**

**에너지, 공격력, 방어막, 사정거리, 이동속도: 170,50,0,30,1**

**유닛BK2에 대한 명령을 입력하시오**

**1:왼쪽이동, 2:오른쪽이동, 3:위쪽이동, 4:아래쪽이동, 5: 상대유닛 공격, 6: 종료 :5**

**BK1을40 만큼 공격하였다.**

**-----유닛BK1---------**

**위치:40,52**

**에너지, 공격력, 방어막, 사정거리, 이동속도: 20,30,10,10,2**

**유닛BK1에 대한 명령을 입력하시오**

**1:왼쪽이동, 2:오른쪽이동, 3:위쪽이동, 4:아래쪽이동, 5: 상대유닛 공격, 6: 종료 :5**

**BK2을30 만큼 공격하였다.**

**-----유닛BK2---------**

**위치:40,62**

**에너지, 공격력, 방어막, 사정거리, 이동속도: 140,50,0,30,1**

**유닛BK2에 대한 명령을 입력하시오**

**1:왼쪽이동, 2:오른쪽이동, 3:위쪽이동, 4:아래쪽이동, 5: 상대유닛 공격, 6: 종료 :5**

**BK1을40 만큼 공격하였다.**

**BK1유닛 사망!!**

**-----유닛BK1---------**

**위치:40,52**

**에너지, 공격력, 방어막, 사정거리, 이동속도: -20,30,10,10,2**

**유닛BK1에 대한 명령을 입력하시오**

**1:왼쪽이동, 2:오른쪽이동, 3:위쪽이동, 4:아래쪽이동, 5: 상대유닛 공격, 6: 종료 :6**