C++ language report

Deadline 2019

[201601639 컴퓨터공학부 홍승현] e-mail : cx\_nf@naver.com Contact : 010 – 4019 - 2514

목차

[Horse.h 클래스 Horse 1](#_Toc21565822)

[private 공간 1](#_Toc21565823)

[PUBLIC 공간 1](#_Toc21565824)

[정리 1](#_Toc21565825)

[HorseManager.h 클래스 HorseManager 2](#_Toc21565826)

[private 공간 2](#_Toc21565827)

[PUBLIC 공간 2](#_Toc21565828)

[정리 2](#_Toc21565829)

[Horse.h 와 Horse.cpp의 알고리즘 설명 3](#_Toc21565830)

[함수 및 생성자 설명 5](#_Toc21565831)

[HorseManager.h와 HorseManager.cpp의 알고리즘 설명 6](#_Toc21565832)

[클래스 설명 6](#_Toc21565833)

[함수 및 생성자 설명 7](#_Toc21565834)

[생성자 7](#_Toc21565835)

[소멸자 7](#_Toc21565836)

[경마 입력 함수 horse\_input() 8](#_Toc21565837)

[선택 경마 함수 select\_input() 11](#_Toc21565838)

[경마 게임 진행 함수 Progress() 12](#_Toc21565839)

[경기 진행 보여주는 함수 Show() 17](#_Toc21565840)

[결과 출력 함수 End\_Announcement() 19](#_Toc21565841)

[Main함수 21](#_Toc21565842)

[출력 사진 21](#_Toc21565843)

[두 번째 출력결과 23](#_Toc21565844)

[이 과제를 함으로써 느꼈던 점 25](#_Toc21565845)

[어려웠던 점, 몰랐던점, 알게 된 점 25](#_Toc21565846)

# Horse.h 클래스 Horse

## private 공간

Int move (말이 타임별로 움직이는 값을 넣기위한 변수)

Int position(말의 위치를 표시하기 위한 변수)

## PUBLIC 공간

1. Horse()

생성자, move와 position의 값을 초기화 해줌

1. void Addposition()

position 멤버에 move값을 더해 얼만큼 갔는지에 대한 위치를 계산해주기 위한 함수

1. 그 외 setMove, getMove, getPosition 등 다양한 set,get함수가 있음.

## 정리

Horse라는 클래스를 만들어 말 그 자체의 객체를 생성 할 목적으로 만듦.

# HorseManager.h 클래스 HorseManager

## private 공간

int first\_win[5] (1등으로 도착한 말을 저장하기위한 배열, 공동 1등이 나올 수 있기 때문에 배열로 만듦)

int count (경마의 수로 초기화 하기 위한 변수, 후에 코딩할 때 간단하게 사용하기 위함)

int select (실행자가 선택한 경마를 입력받았을 때 저장하기 위한 변수)

Horse\* horse (경마의 수 만큼의 객체를 만들기위해 포인터 객체로 선언, 후에 동적할당)

## PUBLIC 공간

1. HorseManager()

생성자, 경마의 수를 입력받아 그 만큼 horse 객체에 동적할당 및 count에 마리수를 초기화, first\_win 변수의 각 요소들을 -1로 초기화, 선택한 경마를 입력받아 select에 초기화 해줌

1. ~HorseManager()

동적할당 받은 객체의 메모리를 해제하기 위한 소멸자 선언

1. int horse\_input()

경마 수를 입력받고 반환할 함수 선언

1. void Process()

경기를 진행시키는 주 함수, 메인 함수에서 사용할 함수

1. void Show(int time\_i)

경기 진행상태를 콘솔애 출력해주는 함수. 매개변수 time\_i는 각 시간대 마다 얼만큼 갔는지를 구분해주기 위해 받는다.

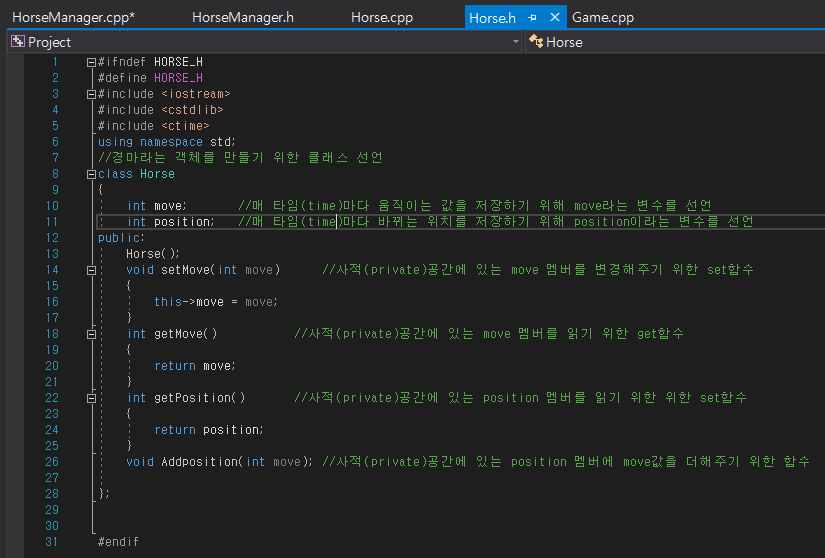
1. void End\_Announcement()

모든 경마가 다 도착을 했을 때 멘트 출력하는 함수

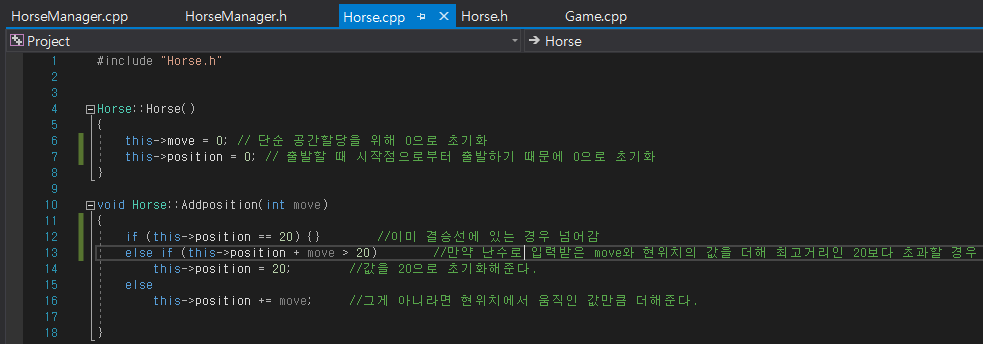
## 정리

Horse객체들을 직접 관리하는 클래스.

# Horse.h 와 Horse.cpp의 알고리즘 설명



<Horse.h 코드>

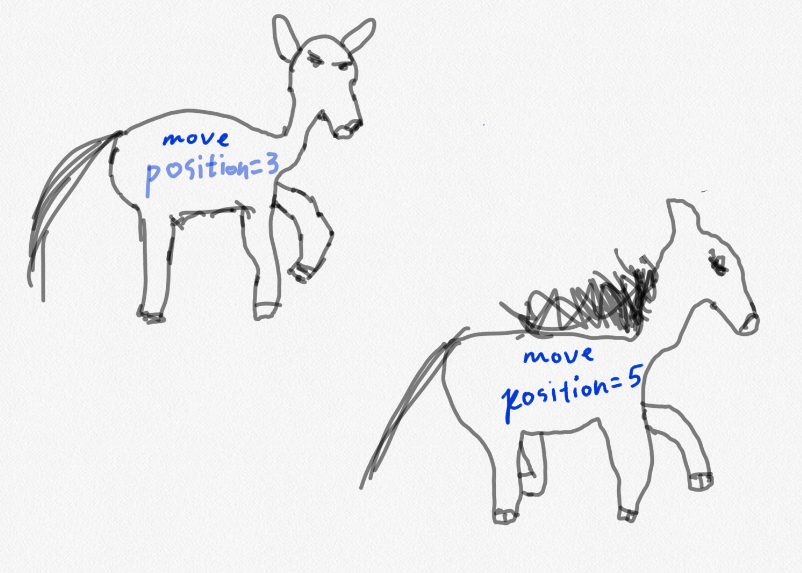


<Horse,cpp 코드>

말 그대로 경마 그 자체의 객체를 만들었다.

경마가 갖추어야할 조건은 매 타임마다 얼만큼 가는지, 그리고 그 위치는 어디인지만 확인하면 되기 때문에, 간결하게 만들었다.

예를 들면 아래 그림과 같이 표현할 수 있다.



<Horse 객체를 표현하기 위한 그림>

(놀랍지만 작성자가 그렸다.)

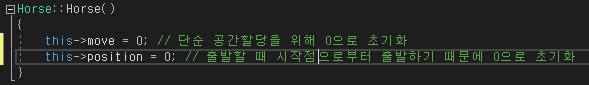
보다시피 경마 하나당 하나의 객체로 볼 수 있고

Position의 값에 따라서 자신의 위치가 얼만큼 가 있는지를, 그리고 move를 통해 얼만큼 가는지를 나타낼 수 있다.

## 함수 및 생성자 설명

위 코드와 같이 일반적인 get, set 함수들은 넘어가겠음.

생성자.

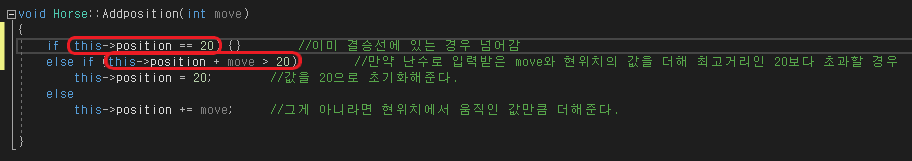


모두 0으로 초기화를 하였으며,.

position은 출발지점으로부터 도착점까지 가는 것이기에 초기값을 0으로 설정,

move는 단순한 값을 저장할 공간을 위해 0으로 초기화.

move, position와 같은 기본값들을 선언해주었고, get, set함수도 만들었으며 이제 매 타임마다 바뀌는 position의 값을 변경시켜줘야 하는 함수가 필요하기에 아래와 같은 함수를 만들었다.

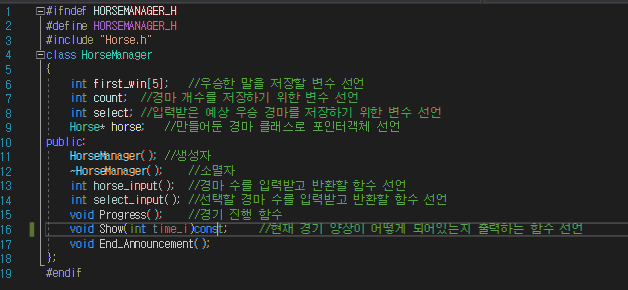


저렇게 예외 표시를 처리해 둔 이유는 결승점 까지의 거리는 20 이며, 만약에 한 마리의 경마가 18이라는 위치에 있었을 때 3 이상의 move를 한다면 결승지점을 벗어나기 때문에 결승지점인 20으로 변경해주는 것이다.

물론 이미 결승지점인 20에 있는 상태면 if문을 빠져나오게끔 하였다.

두가지 조건을 충족하지 못하면 아직 결승점에 도착하지 않은 것이므로 지금 위치에서 움직인 거리를 더해준다. 예를 들면 전 타임에는 위치가 14에 있었는데, 5만큼 가서 19에 위치해있는 것과 같이 계산될 수 있다.

# HorseManager.h와 HorseManager.cpp의 알고리즘 설명



<HorseManager.h 코드>

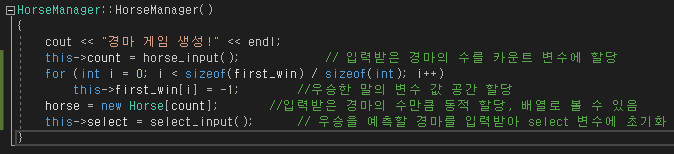
## 클래스 설명

근본적으로 경마라는 객체를 활용한 클래스며, 경마 게임에 대한 전반적인 부분을 담당하고 관리한다. 그래서 클래스의 이름을 HorseManager로 지었다.

근본적인 부분들을 앞에서 설명을 하였으므로 각 생성자와 함수들의 기능들을 알아볼 것이다.

## 함수 및 생성자 설명

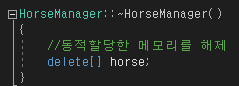
### 생성자



<HorseManager 생성자 코드>

* 메인에서 HorseManager를 불러와 생성자가 실행이 될 때, horse\_input() 함수를 통해 경마의 수를 입력받아 count 변수에 초기화시킨다.
* first\_win배열의 요소의 개수만큼 반복하여 각 요소마다 -1로 초기화해준다.
* 클래스 Horse의 포인터 객체 \*horse를 입력받은 수만큼 동적할당 한다.
* 우승할 것 같은 경마를 입력받아 select 변수에 저장

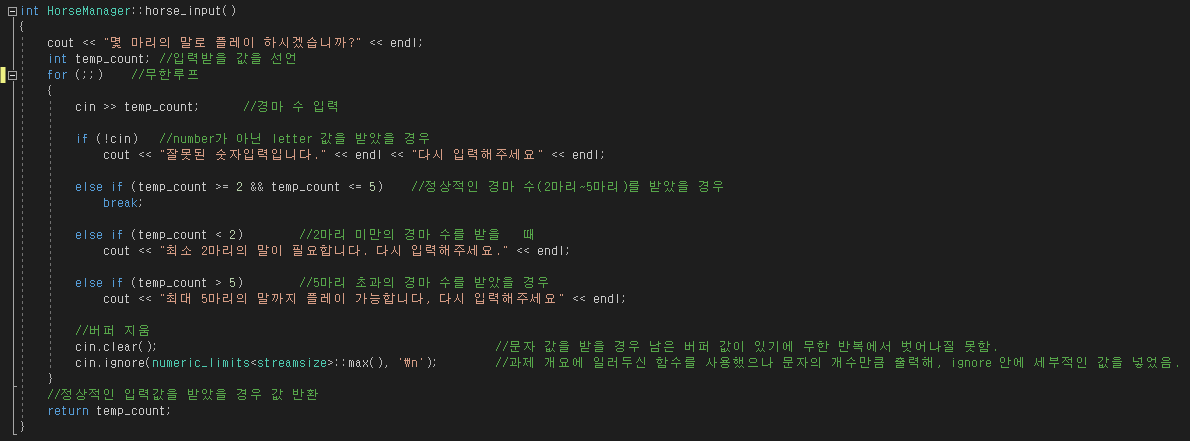
### 소멸자



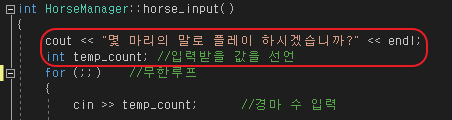
<HorseManager 소멸자 코드>

* 동적할당을 했기 때문에 소멸자를 선언하여 메모리를 해제시킨다.

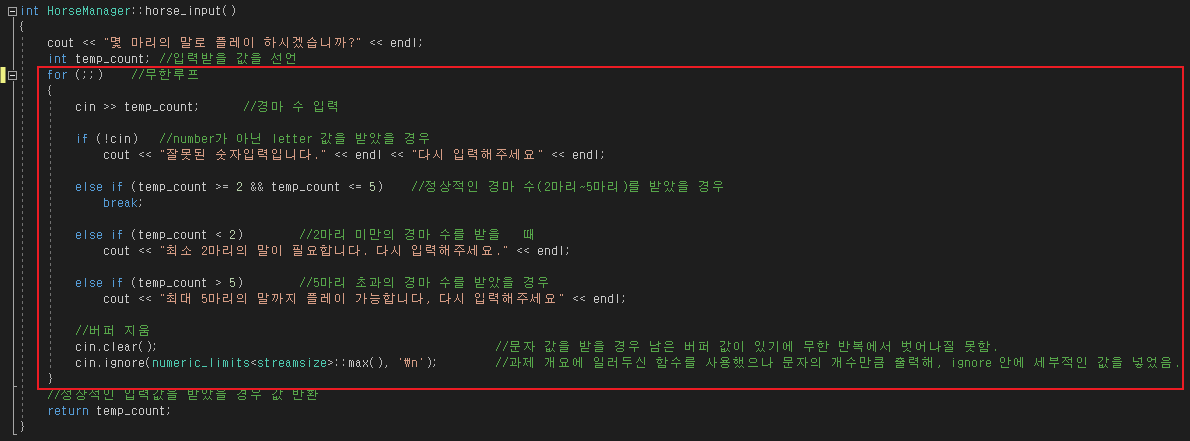
### 경마 입력 함수 horse\_input()



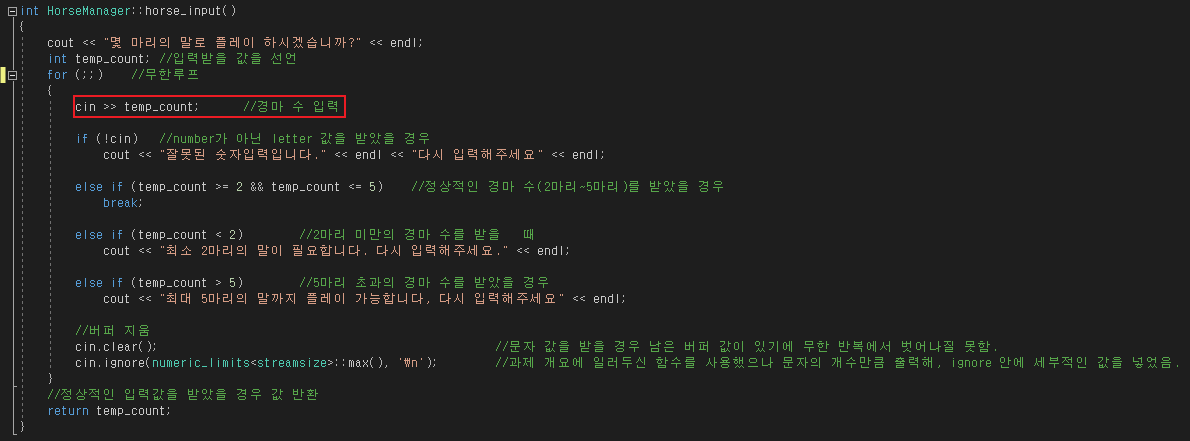
<horse\_input() 코드>



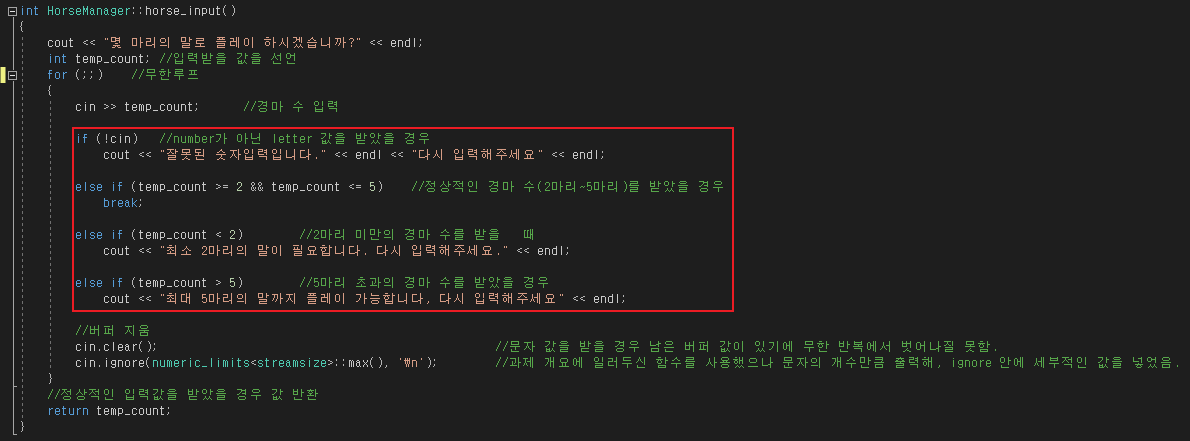
1. 처음에 경마 입력을 받기위한 임시 변수를 만들어 둔다.



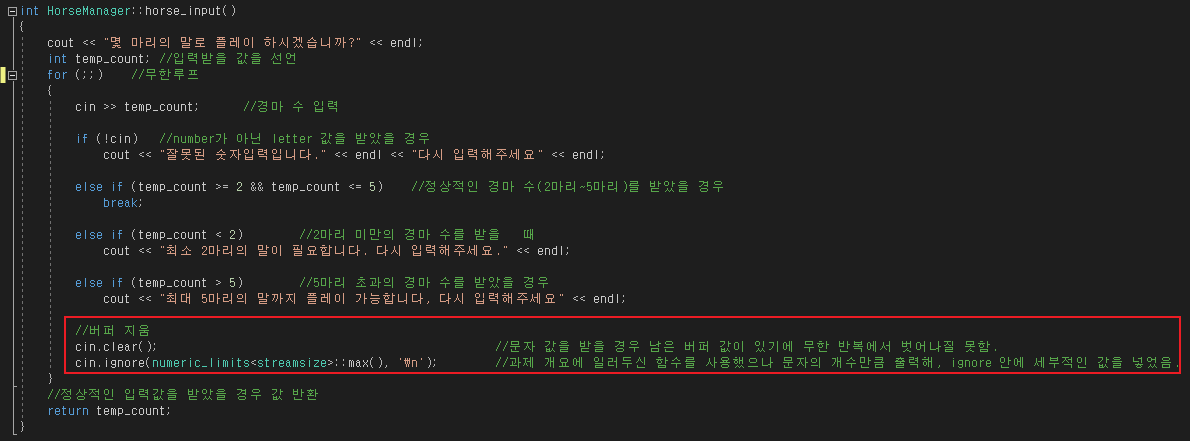
1. 그 후 무한루프문을 돈다.



1. 반복문 앞에 경마 수를 입력 받는다.



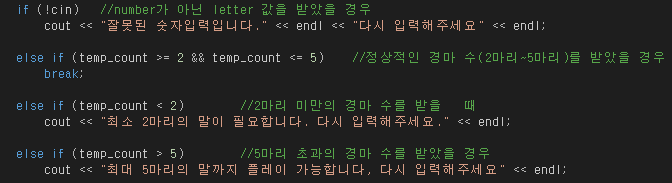
1. 그 후 첫 조건문으로 문자열을 입력 받았을 때 오류문자를 출력하고, 2~5 사이의 숫자를 입력하면 루프문을 빠져나가며, 그 밖의 숫자를 입력할 경우 오류문자를 출력한다.



1. 그 후 남아있을 문자와 버퍼를 지운다.

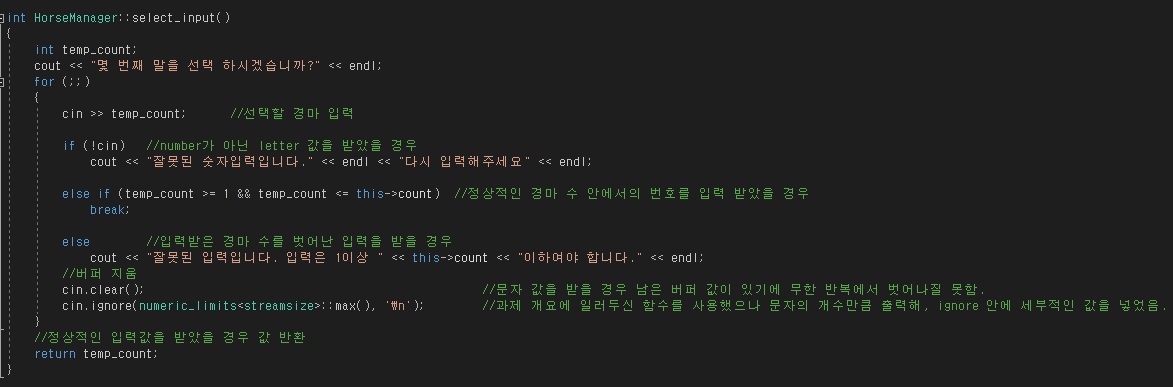
전체적인 함수를 부분적으로 나누어 설명해보았다.

그 중 조건문을 조금 더 자세하게 살펴보자면,



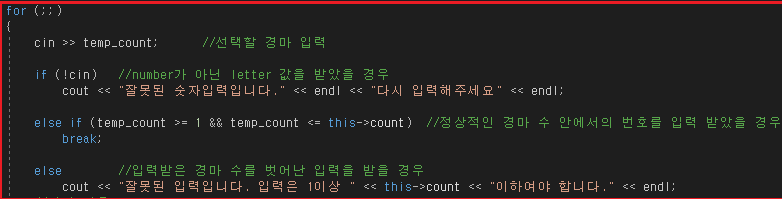
조건문 안에 cin을 넣으면, 입력 받은 게 문자일 경우 거짓을 출력한다고 한다.

### 선택 경마 함수 select\_input()



<select\_input() 코드>

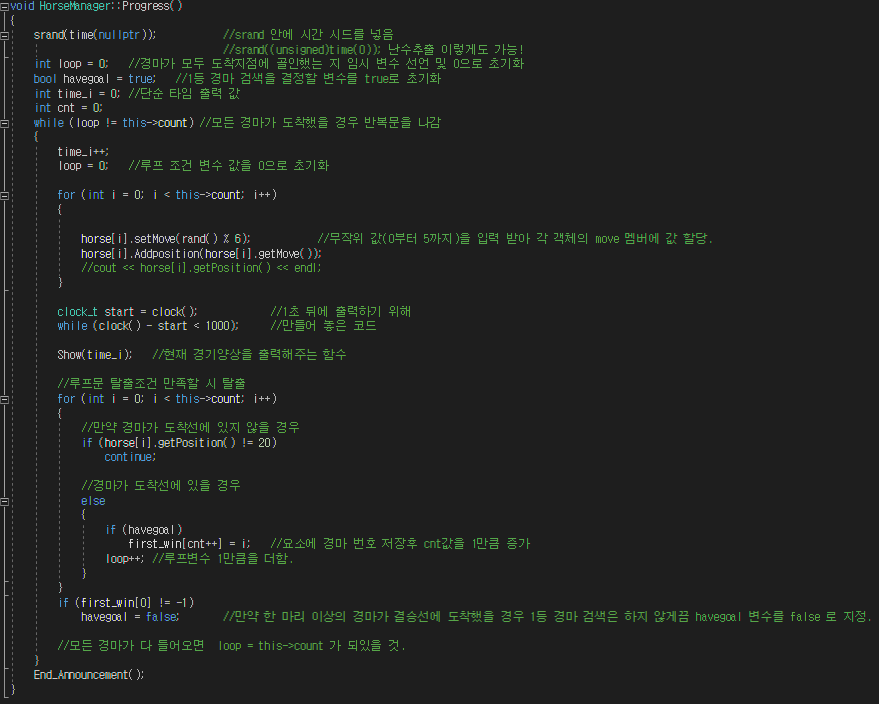
* 경마 수를 입력받는 함수인 horse\_input 과 비슷하나 조건문에서 약간의 차이가 있다.



첫 if 조건문은 같지만 2번째 else if 문부터 살짝 다르다.

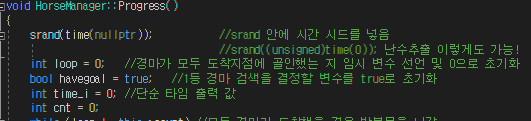
생성자에서 보면, 경마 수를 입력 받은 다음 우승할 것 같은 경마를 선택하는데, 경마 수보다 더 많은 수를 입력하거나, 음수의 숫자를 입력하면 오류가 나기 때문에, 1보다 크고 경마 수보다 작게끔 설정을 해주었으며 오류가 날 숫자를 입력하면 마지막 else문을 통해 경마의 수 이하만큼 입력하라고 출력해준다.

### 경마 게임 진행 함수 Progress()



<Progress() 코드>

* 이 코드가 모든 경마게임의 부분에서 주를 이루며 대부분의 계산들은 여기에서 이루어 진다.



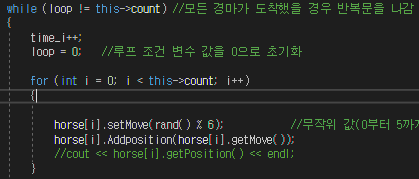
<Progress 함수의 지역변수와 난수>

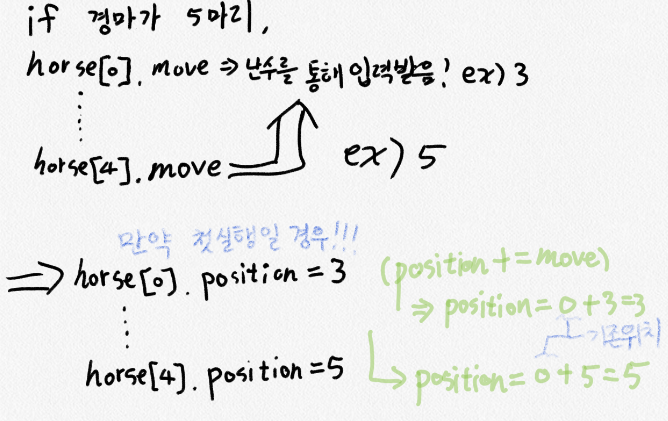
1. Srand를 통하여 시간시드를 넣어 난수를 발생시킨다.
   1. 경마의 움직임을 무작위로 받아서 이동시키기 위함!
2. loop라는 지역변수를 선언하고 0으로 초기화한다.
   1. 후에 모든 경마가 도착하기 전까지 도는 루프문을 벗어나기위한 변수라 loop라 지정해줌!
3. havegoal라는 변수 선언 후 true로 초기화한다.
   1. 만약 특정 타임 때 처음으로 어느 한 경마나 그 이상의 경마가 동시에 도착점으로 왔을 경우, 그 다음에 오는 경마들을 1등 경마를 넣는 변수에 넣지 않게끔 하기 위함(뒤에서 자세히 설명하겠음)
4. time\_i 을 0으로 초기화시킨다. 단순 타임 출력값이며 경마게임이 한 번 실행할 때마다 타임을 출력해준다.
5. 단순 카운트 변수이며, 후에 1등 경마 저장변수를 통해 사용하게 될 변수다.

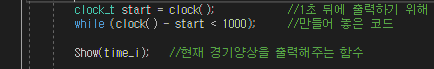


<Progress 주 while문>

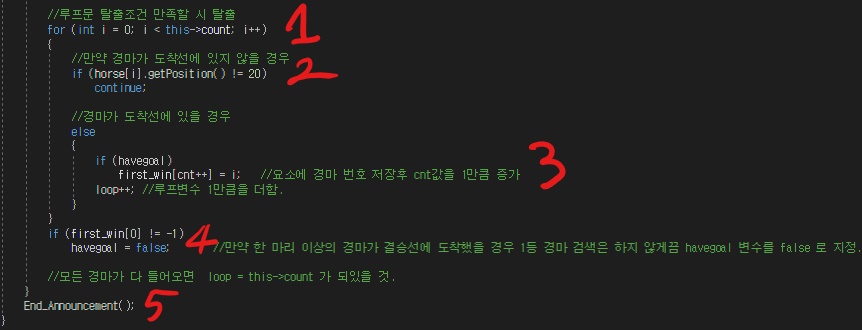
* 처음부터 차근차근 하나씩 짚어서 보도록 할 것이며, 그에 따라 무슨 역할을 하는지 설명에 기재하겠다.



* While문 안에 조건문을 보면 loop != this->count로 되어있는데 뒤에 경마가 도착할 때마다 loop를 1씩 증가시켜서 모두 들어왔을 시에는 count값이랑 같게 되므로 반복문을 빠져나간다.  
  즉, 모든 경마가 도착했을 경우 경기진행하는 while문을 빠져나간다는 의미가 된다.
* time\_i++를 해주는 이유는 단지 몇 번 실행되었는지에 대한 타임을 출력해주기 위함일 뿐이며,  
  loop를 0으로 초기화 하는 이유는, 예를들어 4마리의 경마중 3마리의 경마가 도착했다고 하였을 때, while문에 맞지 않으므로 다시 반복하게 되겠지만, loop에는 3의 값이 들어가 있고, 또 도착하는 검사를 하였을 경우, loop 안에는 6이 되기 때문에, 항상 0으로 초기화 해주어야 한다.
* for문을 각 경마의 개수만큼 돌고, 각 경마의 객체마다 난수를 받아 move에 넣어준다.  
  그 후 경마의 위치와 난수로 입력받아 움직인 move 값을 더해 위치 변수에 넣어준다.
  + 난수는 rand()에서 6으로 나눈 나머지를 통해 move값을 0부터 5까지로 설정해준다.
  +    
    <for문 안 설명하기 위한 정리도>>

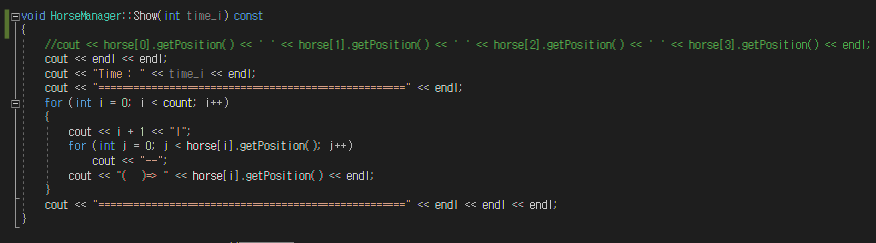


* 1초 뒤에 출력하기 위해 만든 코드
* Show(time\_i) 함수를 실행한다. 후에 설명하도록 하겠음.



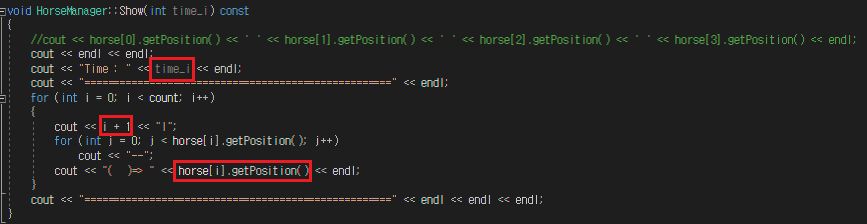
* 경마 개수만큼 반복하면서 경마가 결승선에 도착했는지를 본다.
* 만약 결승선에 도착하지 않았다면 그대로 for문을 다시 돈다.
* 도착했을 경우에는 1등이 도착했을 때 1등경마 저장 변수인 first\_win 배열에 몇 번째의 경마가 1등을 했는지 첫번째 요소부터 넣은다음에 cnt값을 1 증가시킨다.
  + cnt는 단순히 first\_win 배열의 요소를 위해 만들어진 임시 변수다.
  + havegoal이 true일 경우만 실행된다.
* 만약 first\_win의 첫번째 요소에 -1이 아닌 경우는, 이미 1등경마가 들어와서 저장이 된 후기 때문에, havegoal을 false로 바꾸어 다음 검색 때 이루어지지 않게 한다.
* 모든 경마가 다 도착하고 나서End\_Announcement() 함수를 실행한다.
  + End\_Announcement()함수는 마지막 멘트 처리를 위해 만든 함수로 후에 설명하겠음.

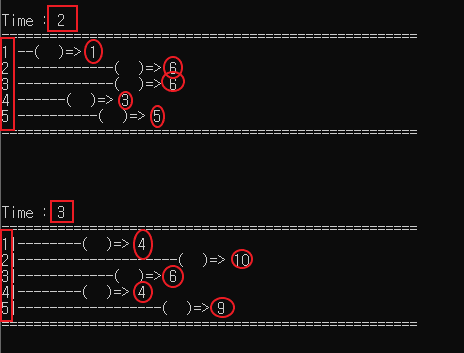
### 경기 진행 보여주는 함수 Show()



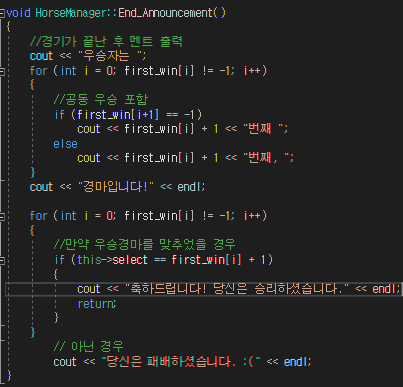
<Show() 코드>

* 전반적인 출력 코드를 표현하였다.
  + 함수 뒤에 const를 붙인 이유는, 단지 경기진행양상을 매개변수를 활용하여 출력하기 위한 함수일 뿐이며 혹여 원치않을 매개변수의 값이 바뀔 수 있기 때문에 const를 붙여 읽기 전용으로 만들어 놓았다.



* 위 빨간 박스에 표시되어 있는 부분은 변수나 함수를 불러와서 출력하는 경우이다. 이 부분은  
  아래 사진에서의 표시된 곳을 출력해주는 부분이다.
* Time\_i는 그 때 일어나는 상황에 대해 몇 번째인지 출력을 해주는 변수이며, 매개변수로 받는다.
* i+1로 출력을 하는 이유는 i는 0부터 시작하기 때문이다. 만약, i로 출력한다면, 0번째부터 4번째까지가 될 것이다.
* Horse[i].getPosition()은 get함수이므로 경마 객체의 Position 변수를 불러오는 함수이다.

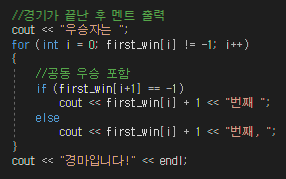
### 결과 출력 함수 End\_Announcement()



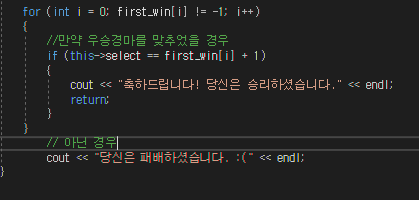
<End\_Announcement() 코드>

* 모든 경마가 결승선에 도착하고 나서 출력하는 함수로, 몇 번째의 경마가 우승했는지, 게이머가 선택한 경마가 1등경마인지를 판별후 패했는지 승리했는지를 출력해준다.

세부적으로 다음 사진부터 설명하도록 하겠음!

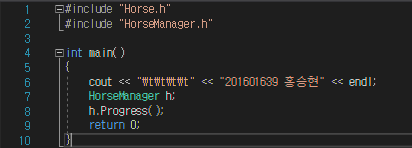


* 1등으로 도착한 경마를 출력하는 코드이며, 동시에 2마리 이상의 경마가 도착할 수 있기에, 그래서 first\_win배열의 요소가 0부터 시작하여 -1이 나오기 전까지를 반복한다. 즉, for문은 1등 경마의 수만큼 돈다는 의미이다.
* 중간의 if 을 넣을 이유는 없지만, 넣은 이유는 단순 출력할 때 “1번째 2번째 3번째” 보다는 , “1번째, 2번째, 3번째”와 같이 쉼표의 디테일을 넣고싶은 마음에 활용하게 되었다.  
  그러므로 first\_win[i+1] == -1 이라 함은 다음 요소가 저장이 안되어 있을 때(-1로 초기화를 했었음)이기 때문에 그 때만 쉼표를 붙이지 않고 그러지 않을 때는 쉼표를 붙여주었다.



* 우승 경마를 맞추었을 경우 승리, 아닐 경우 패배를 출력해주는 코드다.
  + 여러 마리의 경마가 공동 1등일 수 있기에 1등의 경마 수만큼 반복문을 돌아  
    선택했던 경마와 겹치는 게 있는지 비교를 해보고 있을 경우 승리, 아닐경우  
    패배를 출력한다.

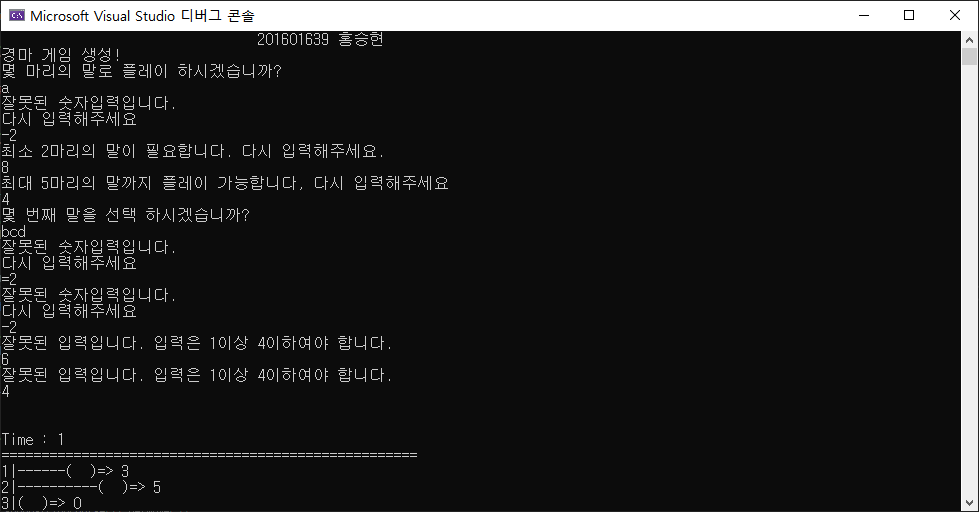
## Main함수

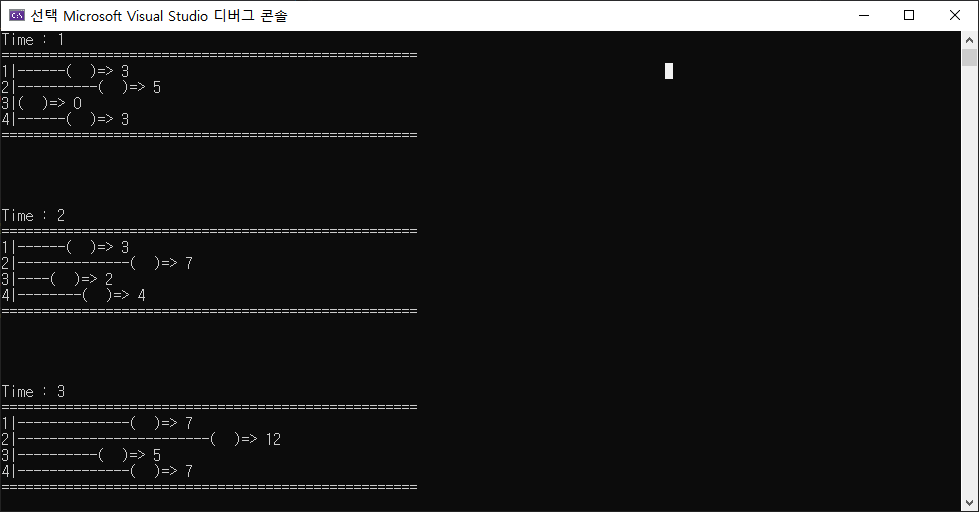


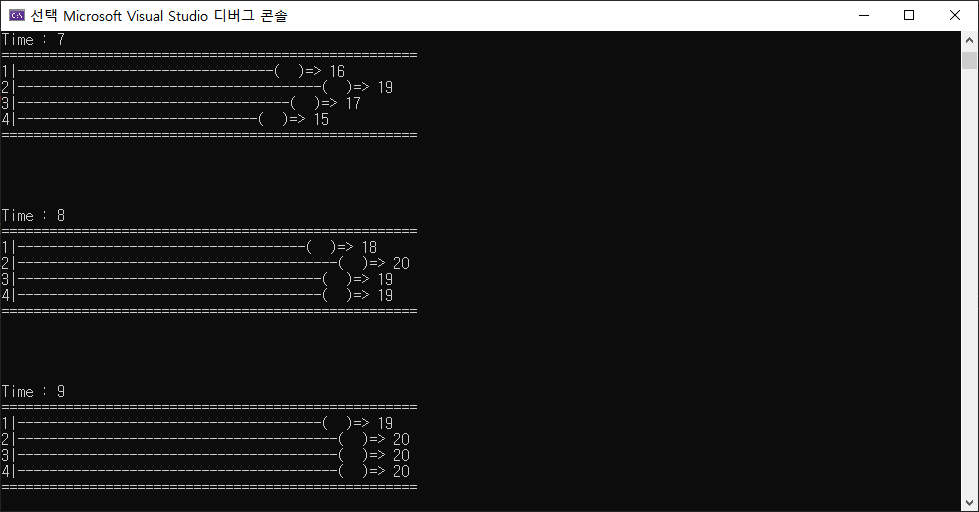
<main 함수 코드>

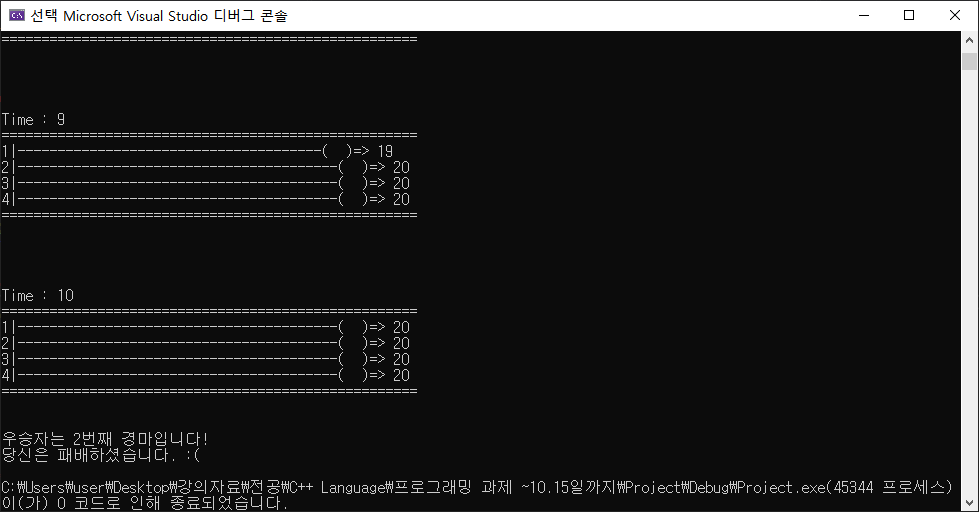
* HorseManager 클래스의 h라는 객체를 선언한 뒤 h.Progress()를 실행한다.

## 출력 사진

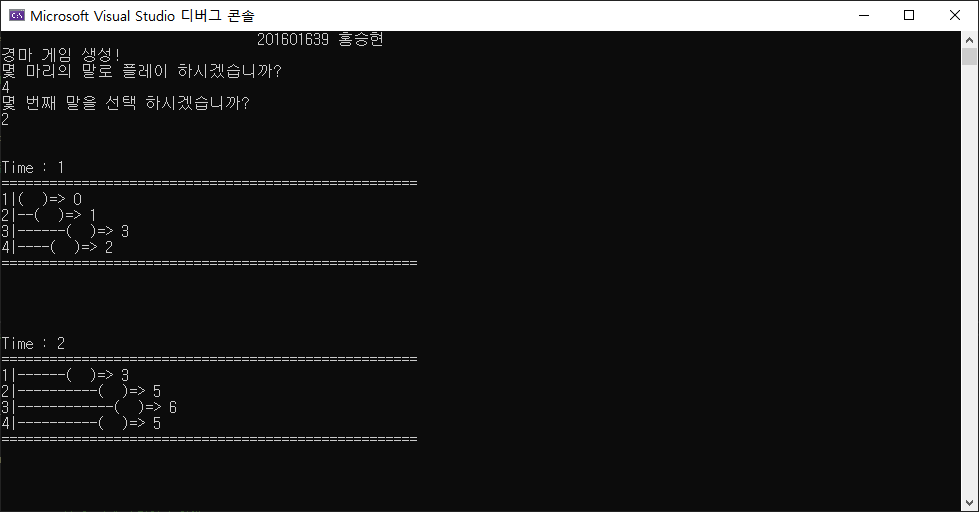


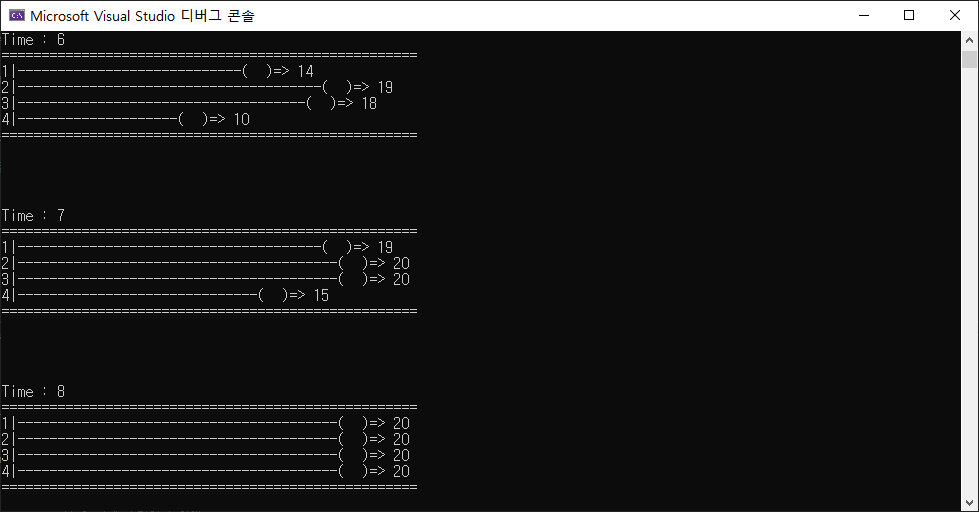


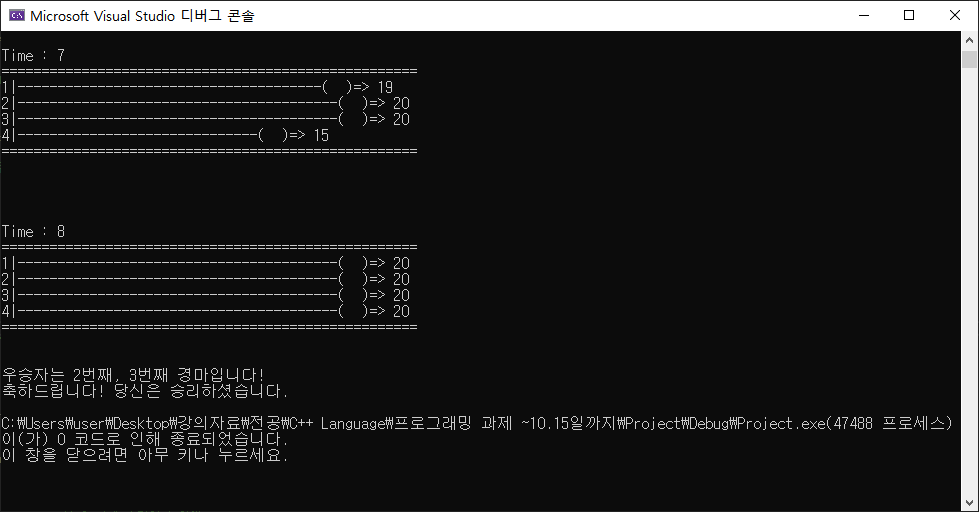




### 두 번째 출력결과







## 이 과제를 함으로써 느꼈던 점

### 어려웠던 점, 몰랐던점, 알게 된 점

1. 처음 코딩을 하려고 시작하자마자 난관을 겪었다.

경마의 수를 입력받으려고 코드를 짰는데 문자를 받을 경우 무한루프 돈 것 마냥 계속 코딩해두었던 에러출력메시지가 나왔기 때문이다. 그 후 cin.clear()과cin,ignore()을 사용하라고 명시되어 있는걸 그제서야 본 후에 사용했으나, 문자 4개를 쓰면 4개만큼의 에러메시지가 출력이 되어서 조교님에게 질문하였다. 그 후에 알게 된 코드가 였고, 이는 무시할 최대 문자수를 설정하는 것이며 뒤의 ‘\n’은 개행문자가 구분자 역할을 해서 즉, 캐릭터가 끝날때까지 무시한다는 의미가 된다.

1. 코딩을 어느정도 한 단계에서 각 타임마다 경마의 위치를 출력하고자 했는데, 1초간격으로 출력하게끔 만들고 싶었다. 허나 대부분의 검색을 통해선 window.h를 통해서 sleep함수를 사용하라는 것이였고 time 라이브러리를 통해서 할 수 있지 않을까란 생각에 조금 더 검색을 해보았다. 그 결과 와 같은 코드들을 알게 되었다.  
   1000은 1초를 나타내고, clock()은 점점 증가하는 함수이다. 그래서 start를 clock()으로 지정하는 순간부터 1초동안은 while문을 돌게 된다.

경마게임을 프로그래밍 해 보면서 알지못했던 것을 깨닫고, 경험해주게 해주신 교수님에게 이 글을 빌어 감사의 말씀을 드린다. 앞으로도 더더욱 학생들을 위해 힘써주셨으면 하는 바람이다.

이상으로 보고서를 마치겠습니다.

작성자 : 201601639 홍승현