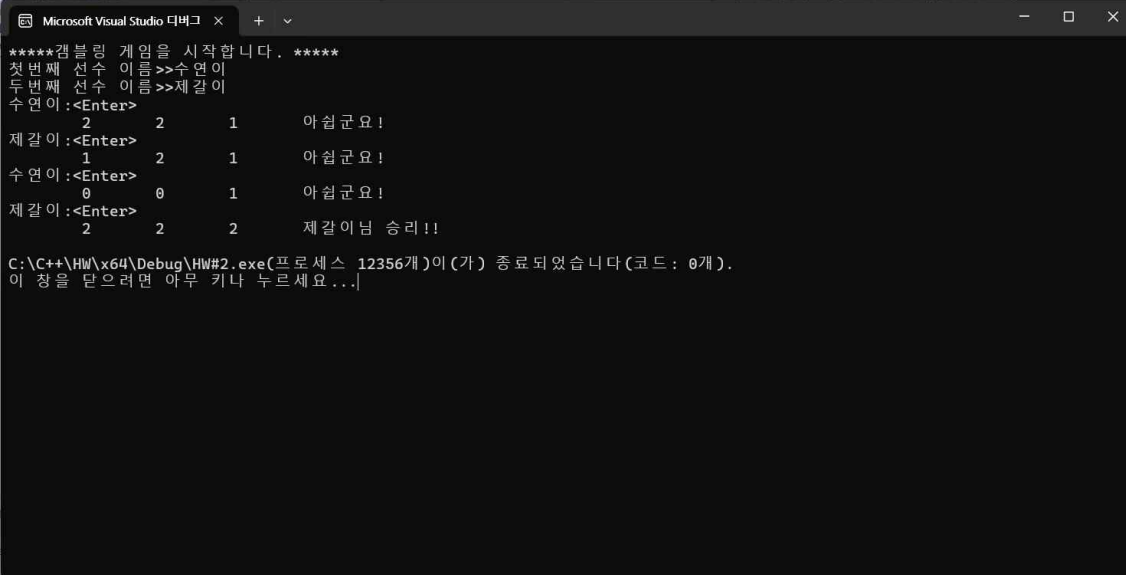


◆ 소스 수행 결과 화면



```
Microsoft Visual Studio 디버그 x + v
****갬블링 게임을 시작합니다. ****
첫 번째 선수 이름 >>수연이
두 번째 선수 이름 >>제갈이
수연이 :<Enter>
2      2      1      아쉽군요!
제갈이 :<Enter>
1      2      1      아쉽군요!
수연이 :<Enter>
0      0      1      아쉽군요!
제갈이 :<Enter>
2      2      2      제갈이님 승리!!

C:\C++\HW\x64\Debug\HW#2.exe(프로세스 12356개)이(가) 종료되었습니다(코드: 0개).
이 창을 닫으려면 아무 키나 누르세요...
```

◆ 소스 구현 설명

- 문제 정의

이 문제는 두 선수가 이름을 입력받고, 각 선수는 Enter 키를 눌러 랜덤으로 생성된 숫자 3개를 확인하여 생성된 숫자가 모두 동일할 경우 해당 선수가 이기게 되는 게임을 만드는 문제이다. 게임은 한 선수가 승리할 때까지 계속 진행된다.

- 문제 해결 방법 / 아이디어 평가

1. Player 클래스

Player 클래스는 선수의 이름과 숫자 정보를 포함하며, 갬블링 게임에서 숫자를 생성하고 승리 여부를 확인하는 기능을 가져야 한다. 선수가 가진 숫자 정보와 선수 이름을 포함하기 위해 card 배열과 name 변수를 설정하였다. Player()라는 기본 생성자와 이름을 입력받은 생성자를 만들고 이름을 반환하기 위한 getName() 메소드와 playGambling() 이라는 갬블링 메소드를 작성하였다.

playGambling 메소드는 각 선수가 세 개의 card 배열에 값을 랜덤하게 할당받고, 이를 출력하는 부분과 숫자의 모든 값이 동일한지 체크하는 부분이 포함되게 만들어야 한다. rand() 함수를 사용하여 0, 1, 2의 난수를 생성하여 랜덤 숫자를 할당하고 card 배열을 채우게 한 후, 숫자를 출력하게 한다. 카드 숫자가 모두 같은지 확인하기 위해 for문을 사용하였고 card 배열의 값을 비교하기 위해 if문을 이용해 숫자가 다르면 false를 반환, 모두 같으면 true를 반환하게 하였다.

2. GamblingGame 클래스

GamblingGame 클래스는 갬블링 게임을 진행하는 데 필요한 모든 것을 관리하게 만들어

야 한다. 즉, 두 명의 선수를 관리하고 게임의 진행 흐름을 제어한다. 두 명의 선수는 배열로 구성하기 위해 player 배열을 만들었고 게임의 완료 여부를 관리하는 isGameCompleted 변수, GamblingGame() 생성자 그리고 게임 진행 메소드인 play() 메소드를 작성하였다.

GamblingGame() 생성자는 선수의 이름을 입력받아 초기화한다. player 배열에 이름을 입력받는 생성자인 Player()로 첫 번째 선수와 두 번째 선수의 이름을 입력받아 초기화하였고 입력 버퍼 클리어를 위해 getchar() 함수를 사용하였다.

그런 다음, 게임을 진행하므로 반복문 내에서 각 선수를 차례대로 턴을 진행해야 한다. 이를 위해 while문을 사용하여 게임이 완료되지 않는 동안 반복하게 만들었고 입력을 대기하기 위해 getchar() 함수를 사용하였다. 잼블링 게임 진행을 위해 if-else문을 사용하여 숫자가 같을 때 게임을 종료하는 경우와 숫자가 다를 때 다른 선수로 턴을 전환하는 경우로 나누어 설정하였다.

3. main함수

프로그램의 시작점이기 때문에 게임 객체를 생성하고 게임을 실행하게 해야 한다. 이를 위해 game이라는 객체를 생성하고 게임 진행 메소드인 play() 메소드를 사용하여 게임을 시작하게 만든다. 매번 다른 결과를 얻을 수 있게 하기 위해 rand() 함수를 사용하기 전에 srand(time(NULL)) 함수를 호출하여 랜덤 시드를 초기화시킨다.

- 문제를 해결한 키 아이디어

문제를 해결한 키 아이디어는 'rand()'함수라고 생각한다. 숫자를 어떻게 랜덤하게 생성할 수 있을지가 의문이었었는데 rand()함수를 사용하여 난수를 생성하여 배열을 채워 숫자를 랜덤하게 출력할 수 있었다. 또한 rand()함수를 사용함으로써 코드가 더 간단하고 직관적이고 효율적으로 되었다.