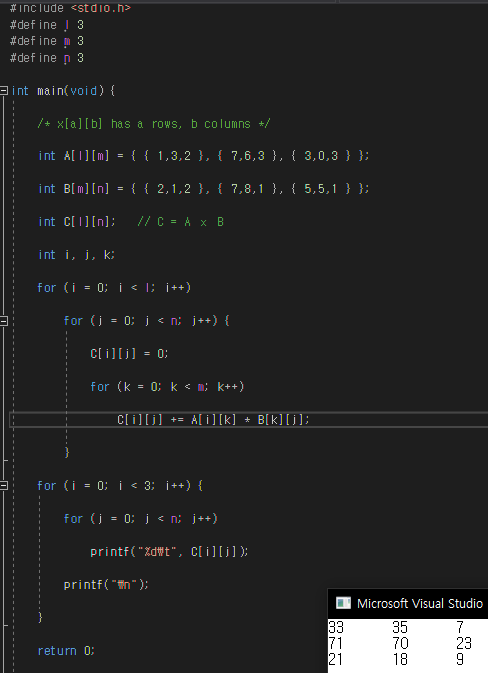


#8

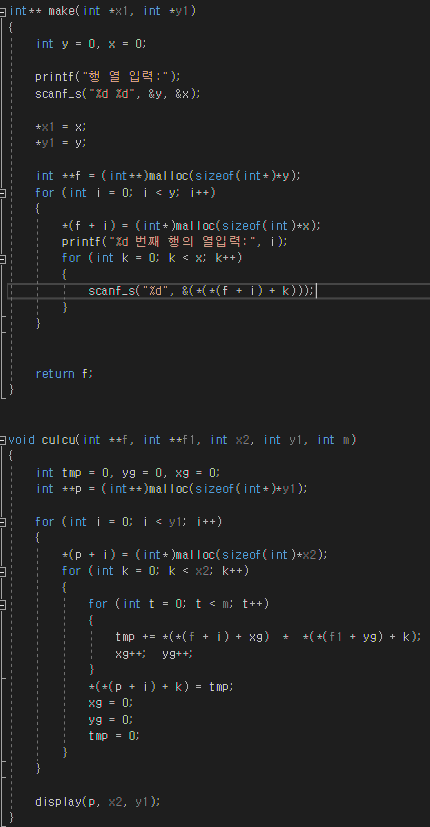
20174627 김혜진

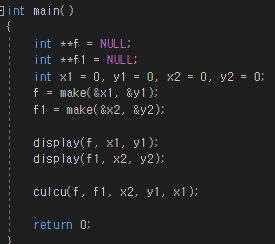
**7. 단지 첨자만을 사용하여 이차원 배열의 원소들에 대한 많은 참조를 포함하는 C 프로그램을 작성하라. 이와 동일한 연산을 수행하나, 포인터와 기억공간 사상 함수를 위한 포인터 산술을 사용하여 배열 참조를 수행하는 두 번째 프로그램을 작성하라. 이 두프로그램의 시간적 효율성을 비교하라. 이 두 프로그램 중에서 어느 것이 더 신뢰적일 것 같은가? 그 이유는 무엇인가?**

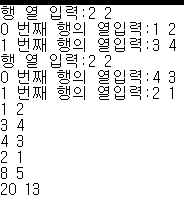
**< 첨자만을 사용 >**



**< 포인터 사용 >**





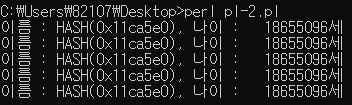
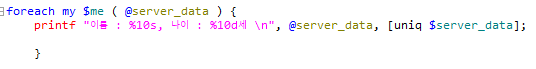


첨자를 사용한 것이 더 신뢰적이며, 포인터를 사용한 것은 비신뢰적 이다. 배열은 매번 배열선두에서부터 출발하지만 포인터는 대상체로 직접 이동해서 바로 읽으므로 액세스 속도가 빠르다. 포인터 변수는 주소를 직접 참조하기 때문에 의도하지 않게 원본의 값이 수정 될 수 있으며, 메모리가 절대 번지에 접근 시 시스템 오류를 초래하기 때문이다. 그만큼 비신뢰적인 만큼 포인터 산술 연산은 동작속도에 있어서 빠르다.

**8. 해시를 사용하고 해시에 대한 많은 연산을 포함하는 Perl 프로그램을 작성하라. 예를 들면, 해시는 사람의 이름과 나이를 저장할 수 있다. 난수 생성기를 사용하여 3개 문자로 구성된 이름과 나이를 생성하고, 이들을 해시에 추가할 수 있을 것이다. 중복된 이름이 생성될 때, 이는 해시에 대한 접근을 야기하나 새로운 원소를 추가하지 않을 것이다. 해시를 사용하지 않고서 이와 동일한 프로그램을 다시 작성하라. 이 두 프로그램의 실행 효율성을 비교하라. 이 두 프로그램의 프로그래밍 용이성과 판독성을 비교하라.**

|  |  |
| --- | --- |
| **해시 사용** | **결과** |
|  |  |
| **해시 비사용** | **결과** |
|  |  |

해시를 사용한 경우 uniq라는 함수를 생성해 $server\_data라는 변수에 함수를 적용시켜 중복 값이 제거 되도록 하려 했지만, 중복 값은 제거 되지 않고 각각의 key와 value값이 랜덤하게 생성되어 출력되는 것을 볼 수 있다. [uniq $server\_data]로 값을 주게 되면 ARRAY의 주소 값이 이름으로 출력 되지만, @server\_data로 값을 주게 되면 아래 이미지와 같이 HASH의 주소 값이 이름으로 출력 된다. 그리고 랜덤으로 생성 되었던 나이는 모두 같은 값으로 변경 되어 출력되는 것을 볼 수 있다.



반면에, 해시를 사용하지 않은 경우 각각의 이름과 나이는 랜덤으로 출력 되며, @set을 사용해 중복 값이 나오지 않도록 해주었다. 2페이지에서 보인 결과와 같이 해시를 사용한 경우와는 다르게 확실히 난해하지 않은 결과가 도출된 것을 확인할 수 있다.

결과적으로 해시를 사용할 경우 해시를 사용하지 않았을 때 보다 판독성과 용이성이 떨어지게 된다. 먼저 판독성이 떨어지는 이유는, $나 @의 형태를 참조하게 됐을 경우 key와 value의 형식에 따라 출력 되는 값이 제각각 이었다. 또한, Key와 value에 맞춰 참조 값을 지정해주게 되면 해당 참조 값에 따른 선언도 별도로 필요 했으며, 애초에 해시라는 기준을 놓고 봤을 때 key와 value를 지정해 값을 도출해내야 한다는 전제라면, 각각의 값을 지정해주지 않고 랜덤으로 값이 생성될 범위만 설정해 주면 되는 후자가(해시를 사용하지 않는) 용이성을 더 높여주지 않았을까 싶다.