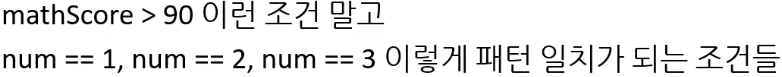
**1. switch 문**

* **If 문에서의 가독성이 좋지 않은 경우**
* 조건문이 많으면 가독성이 좋지 않음  
  - 하지만 반드시 그러지는 않음
* 조건문에서 사용한 표현식의 형식에 따라 다름  
  - ex) 다음의 경우에는 가독성이 좀 더 나음  
    
  테이블이(가) 표시된 사진

  자동 생성된 설명
* **case 문의 형태**텍스트이(가) 표시된 사진

  자동 생성된 설명
* **(c#에서는) 모든 case 구문 다음에는 break 구문을 넣어야 함**
* 없다면 컴파일 오류가 발생함  
  - fall-through: break문이 없어서 코드를 따라 그대로 쭉 실행되는 경우
* **default 구문**
* 매치 표현식의 반환값과 일치하는 case 구문이 없을 경우 실행  
  - default 구문의 끝에도 반드시 break 구문을 넣을 것
* 때로는 이렇게 잘못된/예상하지 못한 반환값을 잡아야 할 때가 있음  
  - 이 경우 어서트(assert)를 사용함
* **case 문에서 사용할 수 있는 자료형**
* int(기본), long, char, bool, string(**c# 전용**)  
  - 부동 소수점형들은 사용 불가
* char형이 가능한 이유  
  - char형은 숫자형이므로 따라서 int가 가능하면 char도 가능함
* **switch문이 if문보다 가독성이 좋은 이유**
* if문의 경우마다 모든 조건식을 서술해야 함  
  - 이때 한 대상이 아니라 다른 대상도 비교하는 경우 실수할 확률이 증가함  
  - 특정 대상 하나만을 검사하고 비교할 때는 switch문이 훨씬 가독성과 실수할 확률이 낮아짐
* 결과적으로 실수를 방지할 수 있는 코드가 좋음