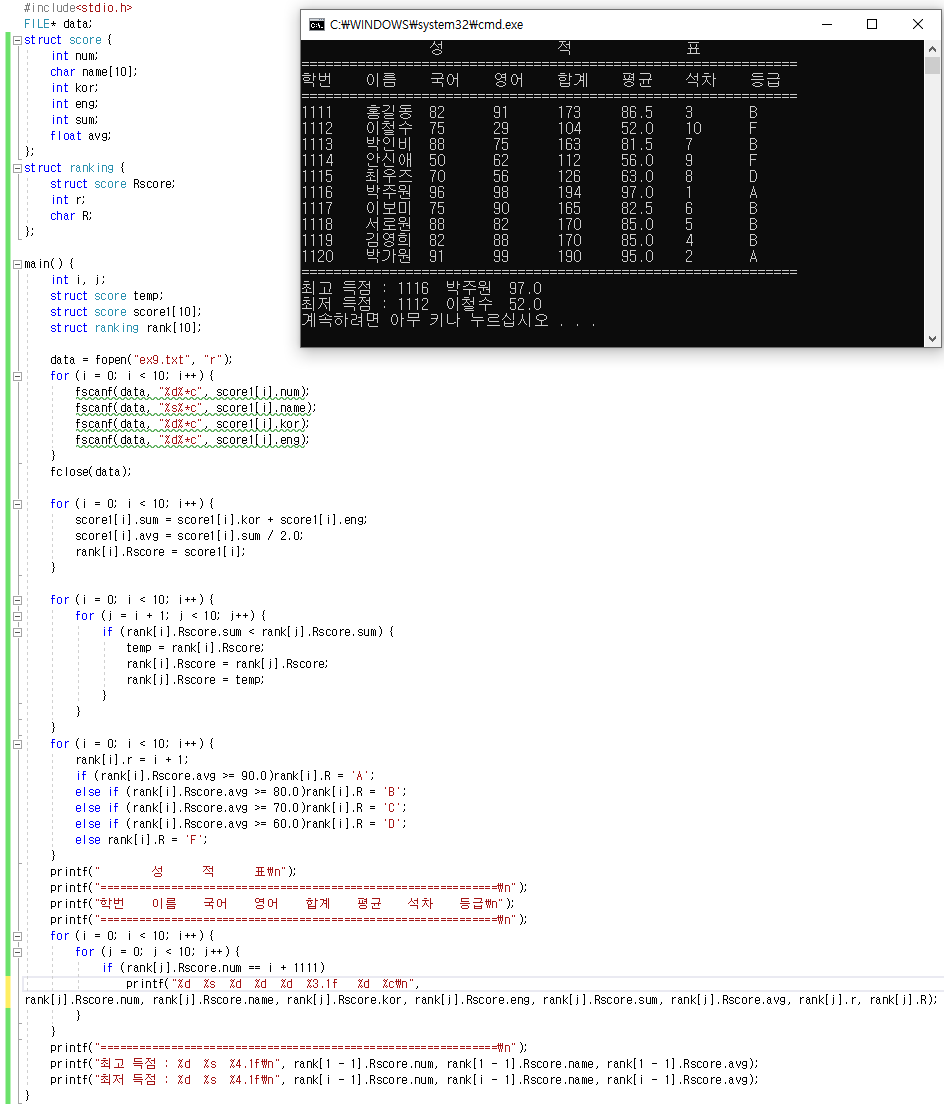
15-9.



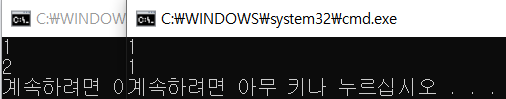
<12장 연습문제>

1. 기억 클래스 :

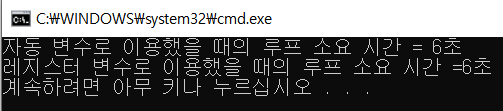
|  |  |  |
| --- | --- | --- |
| 자동 변수 | (auto) 데이터형 변수명; | 블록 내(중괄호)에서 적용되며, 블록을 탈출하는 순간 할당되었던 주소에 해당하는 값이 제거된다. (다시 돌아오면 새로운 주소에 값을 재 지정한다.)  전역 변수로 사용할 수 없다. |
| 정적 변수 | static 데이터형 변수명; | 블록 내에서 적용되나, 블록을 탈출해도 변수 값이 저장된 주소가 유지되어 해당 블록으로 돌아와도 변경된 변수 값이 그대로 적용된다.  전역 변수로 사용할 수 있다. |
| 레지스터 변수 | register 데이터형 변수명; | 자동 변수와 비슷하게 적용되나 주소가 cpu 메모리에 할당되어 대용량 데이터를 처리할 때 수행속도에서 유리하다.  전역 변수로 사용할 수 없다. |
| 외부 변수 | extern 데이터형 변수명; | 프로그램 내에서 다른 소스코드에 있는 변수 값이나 블록 외부의 프로그램 순서상 뒤에 정의되는 변수를 사용하기 위한 선언이다.  정적 변수와 비슷하게 주소에 해당하는 값이 유지된다.  외부에 지정된 자동 변수나 레지스터 변수도 불러와 전역 변수로 사용할 수 있다. |

전역 변수: 선언 후 프로그램이 종료될 때까지 유효한 변수를 말한다. 함수 외부에 지정하여 코드 전체에 대하여 지정할 경우 코드 내의 블록에서 해당 변수와 똑 같은 이름의 변수를 지정하면 블록 내의 변수를 우선한다. 프로그램 코드 순서를 따른다.

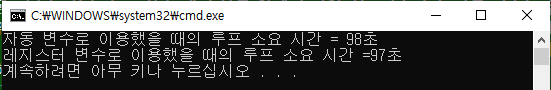
1. 변수의 유효 범위는 지정된 블록과 블록 내부의 하위 블록에서 유효하며, 전역 변수와 블록 내의 변수가 같은 이름일 경우 블록 내의 변수를 우선한다. 전역 변수의 경우 1번의 추가 설명에 따른다.
2. 정적 변수의 경우 주소 값이 한번 할당 되면 프로그램 종료까지 유지되지만, 자동 변수의 경우에는 변수 지정 코드를 다시 지나가게 되면 해당 주소에 다시금 값을 지정한다.



1. max = 50000



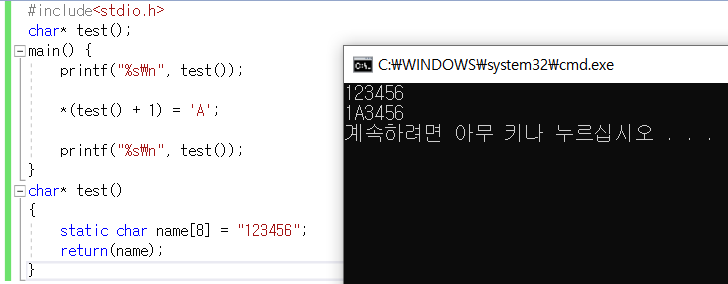
max = 200000



큰 데이터의 경우 작업에 소요되는 시간이 기하 급수적으로 늘어나며, 자동 변수와 레지스터 변수의 작업 완료 시간도 눈에 띄는 차이가 나타난다.

빅 데이터의 경우 프로세서의 메모리에 할당되는 레지스터 변수를 사용하는 것이 유리하다.

1. 변수 name의 2번째 문자인 2가 A로 변경되어 출력되며, 이는 name 변수 주소에 저장된다.



1. 변수의 유효 범위에 따른다.

