**태어나서 처음으로 혼자 프로그램을 만들어 보았다.exe**

1519정현서

프로젝트명: 즐거운 산화 환원

내용: 화학 반응식을 입력받았을 때 각 원소의 산화수 변화를 파악하여, 궁금한 원소의 산화/환원 여부를 알려주는 프로그램을 만든다.

입/출력 예시: 1. 입력 반응물: 2H2O2(0), 생성물: O2(0)+2H2O(0), 알고싶은 원소:O

출력 O 은(는) O2 로 될 때 산화됨 (산화수: -1.0 -> 0.0 )

O 은(는) H2O 로 될 때 환원됨 (산화수: -1.0 -> -2.0 )

2. 입력 반응물:2S2O3(-2)+I2(0), 생성물:S4O6(-2)+2I(-1), O

출력 O 은(는) 산화되지도 환원되지도 않음 (산화수: -2.0 -> -2.0 )

알고리즘: 반응물과 생성물을 따로 입력받은 후, 반응물과 생성물에 대해 일련의 가공을 거쳐 산화수를 각각 구한 후에 반응 전, 후의 산화수를 비교할 것이다.

1. A라는 원소를 알고 싶다고 할 때, 반응물 중 A가 포함되어 있는 물질을 골라내서 그 물질들에 어떤 원소가 몇 개나 포함되어 있는지 알아낸다. 그 후 리스트를 만든다.

[전하량, 원소1, 원소1의 개수, 원소2, 원소2의 개수…]

1. 산화수를 결정하는 규칙을 분기문을 이용해서 지정하고, 반응물 리스트와 일대일대응되는 산화수 리스트를 만들어서 각각의 원소들의 산화수를 결정한다.
2. 비교한다.

해결해야 할 문제

1. 입력받을 때 위첨자, 아래첨자 구분이 없기 때문에 어떤 물질의 전하량을 제대로 파악할 수 없다.
2. 원자단 이온이 있는 물질, 즉 분자식에 괄호가 있는 물질에 대한 원소 분리는 어떻게 하지?
3. 산화수가 결정될 때 플루오린(F)이 가장 우선이고 그 후에 알칼리 금속,… 순인데 물질에서 원소를 분리한 순서는 그 순서와는 무관하게 그냥 분자식에서 먼저나온 순서이다. F부터 산화수를 지정해 줘야 하는데 F부터 할 수가 없는 상황이다. 반복문을 여러 번 돌려야 하나??
4. 알칼리 금속이나 알칼리 토금속은 해당되는 원소가 너무 많아서 반복문에 나열하기가 힘들다. 어떻게 처리하지?
5. 반응물, 생성물이 각각 하나라는 보장이 없다. 반응물 여러 개, 생성물 여러 개인 상황에서는 어떻게 해결해야 하지?
6. 시성식은 어떻게 처리할 것인가…. CH3COOH와 같은 경우에는 앞에 있는 C의 산화수는 -3이고 뒤에 있는 C의 산화수는 +3이다.

해결방안

1. 입력받을 때 물질의 가장 뒤에 (전하량)을 입력하게 하면 전하량만 구분할 수 있다.
2. 코드100줄 정도로 해결할 수 있다. 이중 반복문을 돌리고, 경우를 이것저것 따져주면 된다.
3. 어떤 물질에 있는 원소의 종류만큼 산화수 리스트에 0을 추가해 놓고, F가 있으면 F의 자리에 0을 -1로 바꾸고 알칼리 금속이 있으면 1로 바꾸는 방식으로 하면 된다. 분기문으로 우선순위 지정도 가능하다.
4. 알칼리 금속, 알칼리 토금속 원소를 포함한 리스트를 각각 만들어서 그 리스트들과 반응물 리스트의 공통 원소를 리턴하는 함수를 정의하였다.
5. 이차원 리스트를 만들면 해결 가능하다. 이러면 반응물이 1개이고 생성물이 여러 개인 상황이나 반응물이 여러 개고 생성물이 하나인 상황은 커버가 가능한데 반응물도 여러 개고 생성물도 여러 개인 상황은 커버가 안된다. 구현은 가능하지만 어떤 물질이 어떤 물질로 변했는지를 판단할 수가 없다. 그나마 그럴싸한 방법은 각각의 물질에 들어있는 원소의 종류를 비교해서 비슷한 물질을 찾는 것이다. 서로 비슷한 물질은 반응에서 관계가 있다고 추론할 수 있을 것이다. 근데 이게 예외가 많아서 처리가 좀 많이 힘들 것 같다.
6. 과학적으로 접근하지 말고, 그냥 해결을 위해서만 접근하면 원소 종류별로 개수만 다 합쳐서 계산할 수 있다. 근데 그러면 의미가 없어서… 미해결 상태로 둬야겠다. 알고리즘적으로 해결도 불가능할 것 같다.

힘들었던 점

1. 변수의 타입: str, int 구분하는 것이 너무 힘들었다. 솔직히 말해서 이 코드 짜면서 제일 많았던 오류가 typeerror 같다.
2. 예외 처리: 원소를 분리해낼 때(원자단이 있는 물질에서)도 특별한 상황 때문에 코드가 많이 길어졌고, 산화수 규칙 분기문에서도 예외 때문에 길어졌다.
3. 변수가 너무 많다: 일반적인 상황까지 모두 커버할 수 있는 코드를 구상하다 보니 전체 스케일이 너무 커져 버렸고, 변수도 너무 많이 지정했다. 변수 가독성이 나쁘지는 않아서 코드를 짜는 과정에서 혼란은 없었지만 그것 때문에 코드가 너무 길어졌고 메모리도 커진 것 같다.

결론

400줄짜리 코드 완성.