**예비프로젝트 1-1 DFA 시뮬레이터**

20160389 원종하

1. 소스 코드 및 실행 파일

zip 파일에 .py 파일 별도 첨부.

2. 매뉴얼

2-1. 입력 방식

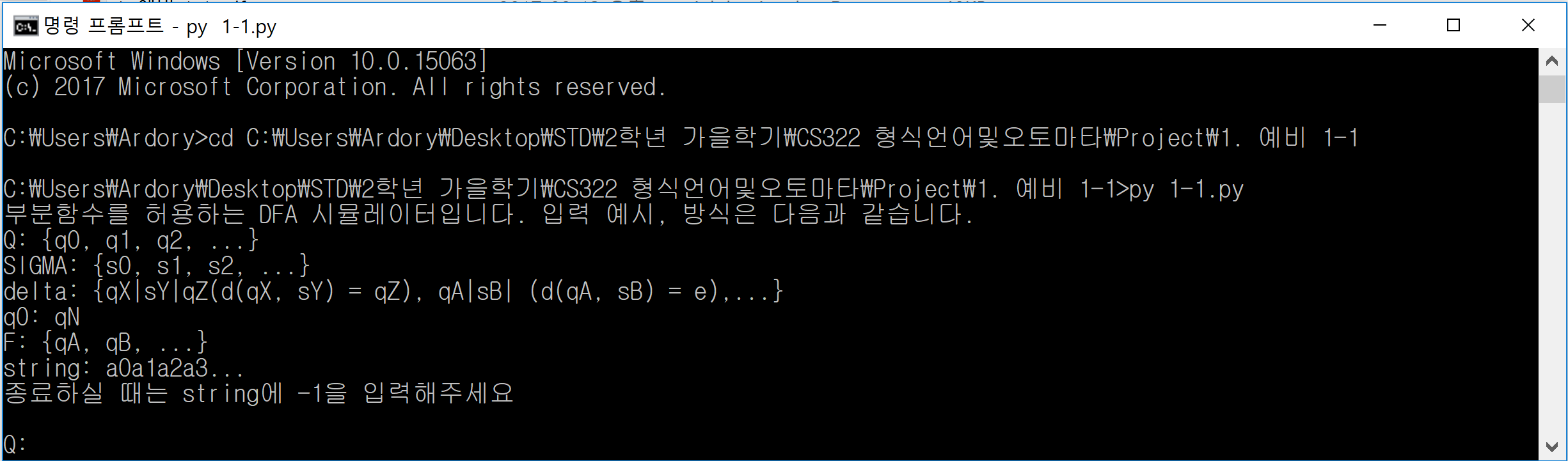


Figure 1

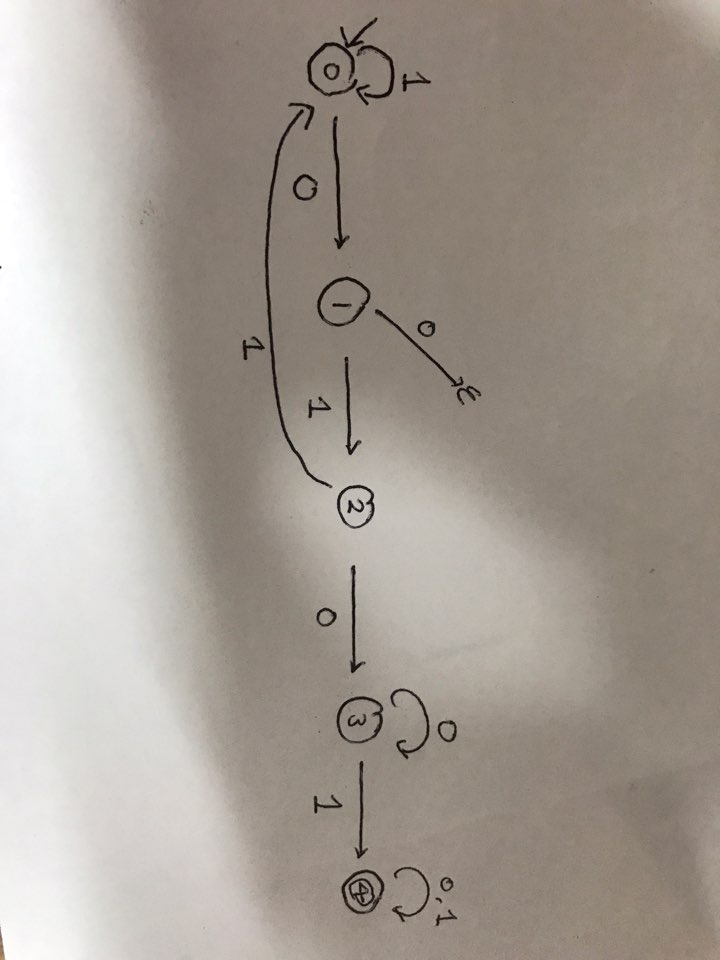


Figure 2

Figure 1에 나온 바와 같이, Figure 2와 같은 DFA을 입력하려면 차례로

**{0, 1, 2, 3, 4}**

**{0, 1}**

**{0|0|1, 0|1|0, 1|0| , 1|1|2, 2|0|3, 2|1|0, 3|0|3, 3|1|4, 4|0|4, 4|1|4}**

**0**

**{4}**

을 입력한 후, 원하는 string을 그 후에 입력해주면 된다.

이 입력은 공백을 인식하므로 주의할 필요가 있다. 반점(,) 이후에는 항상 공백을 입력해야 한다.

3-2. 출력 방식

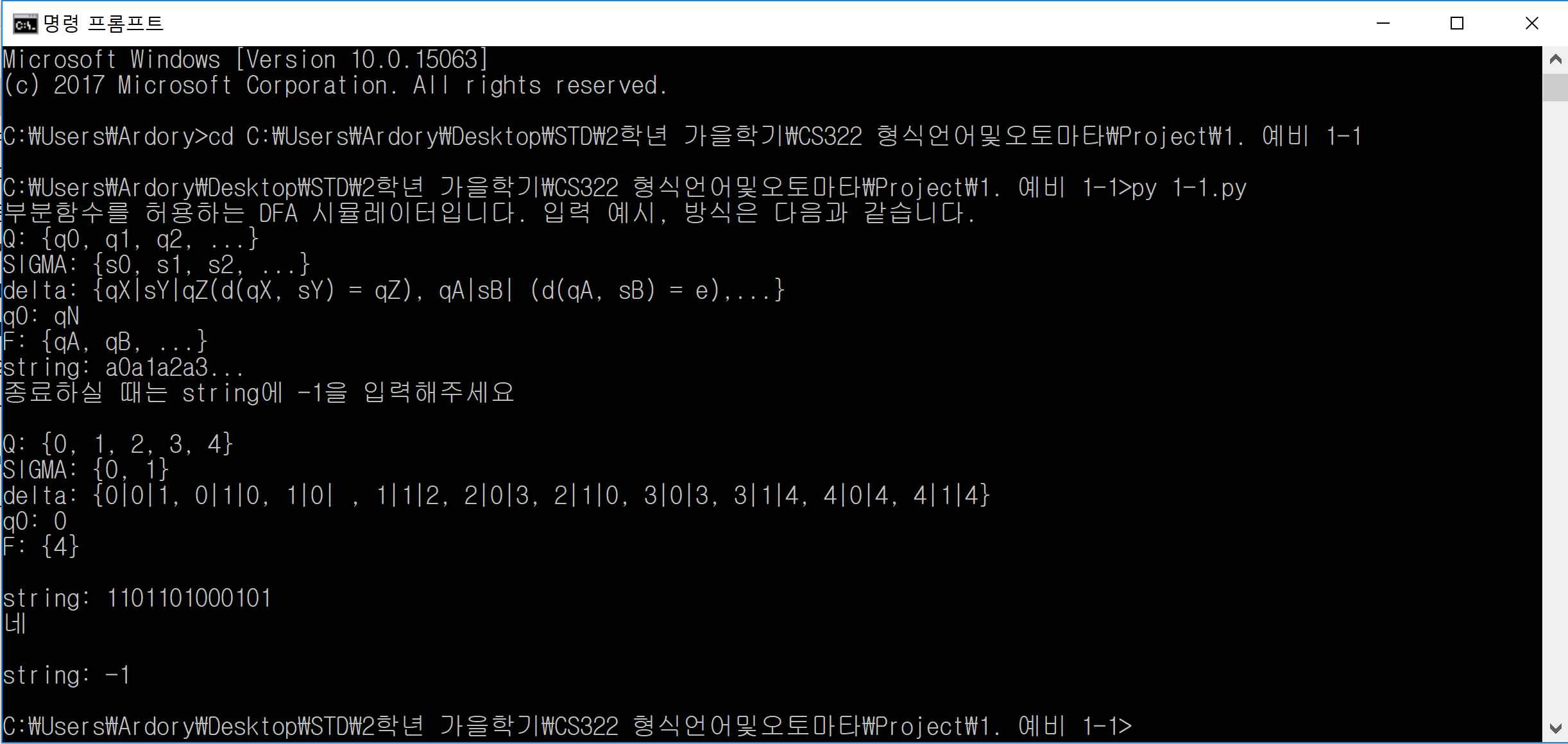


Figure 3

Figure 3과 같이, 정상적인 상황에서는 ‘네’ 혹은 ‘아니오’만을 출력한다. string에 -1을 입력하면 출력하지 않는다.

비정상적인 상황에서의 에러 케이스는 총 4가지로,

1. function에 비정상적인 state() 입력
2. function에 비정상적인 input symbol() 입력
3. 에 비정상적인 state () 입력
4. 에 비정상적인 state () 입력이 있다.

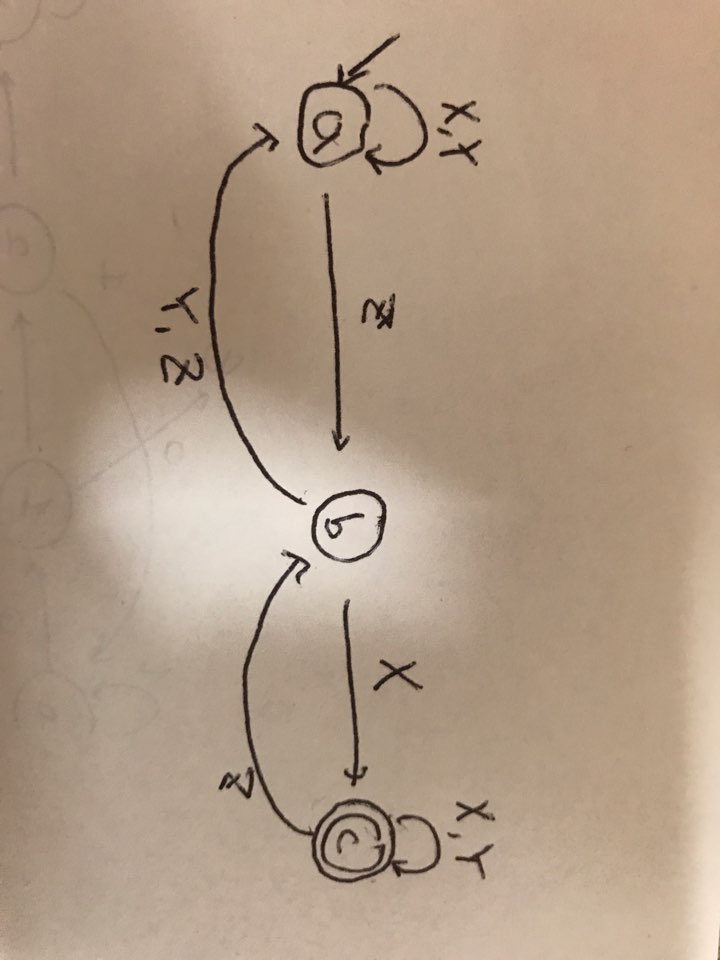
4. 프로그램 설명

\_\_init\_\_, in2df, Flstcheck, act, YES, NO, ErrorMessage 메서드로 이루어진 class DFA를 이용한다.

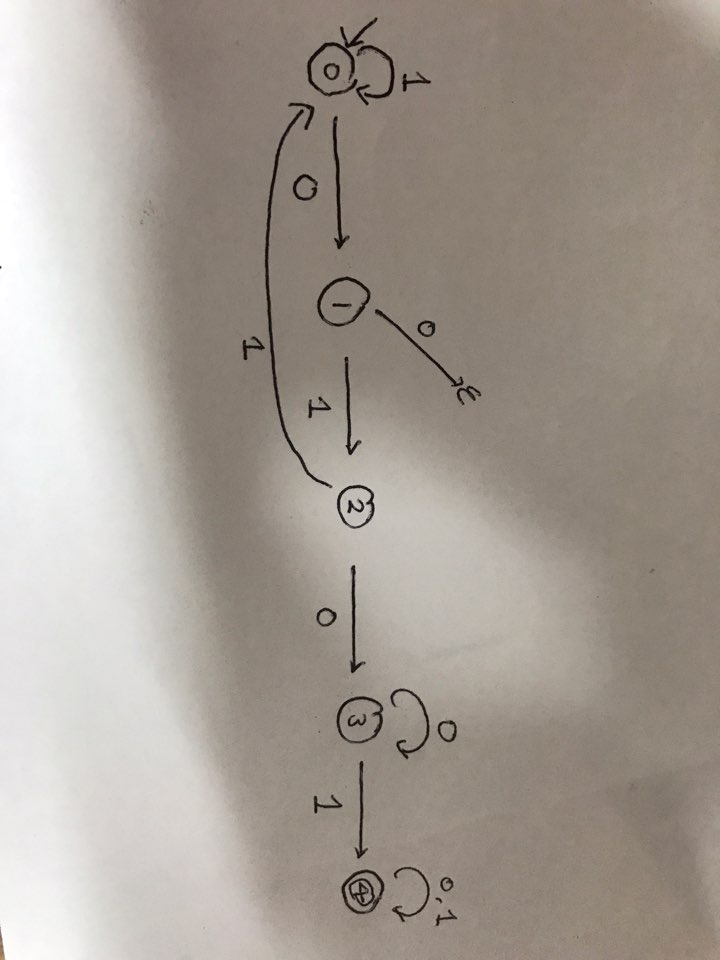
|  |  |
| --- | --- |
| 메서드 | 설명 |
| \_\_init\_\_ | active: 실행시킬 수 있는 DFA인가? (boolean)  Qlst(Q-list): 상태들의 유한 집합 (list)  Slst(Sigma-list): 입력 문자들의 유한 집합 (list)  ddic(delta-dictionary): 상태변화함수. (qN, sM) tuple을 key로, qM을 Value로 하는 dictionary (dict)  qzr(q-zero): 초기 상태 (str)  Flst(Final-list): 최종 상태들의 유한 집합 (list)    active는 정상적인 상황에서 True이며, ErrorMessage에 진입하였을 때 False가 된다.  Qlst와 Slst는 중괄호 ({})을 슬라이싱한 후 ', '을 기준으로 스플릿하면 입력을 원소로 가지는 list가 된다.  ddic은 마찬가지로 슬라이싱/스플릿 과정을 거친 후 in2df 메서드를 이용하여 dict type, 혹은 ErrorMessage(1 or 2)를 return한다.  qzr은 입력된 값이 Qlst에 있다면 입력된 값을, Qlst에 없다면 ErrorMessage(3)를 return한다.  Flst는 Flstcheck 메서드를 통해 검사를 거친 후 list type, 혹은 ErrorMessage(4)를 return한다. |
| In2df | in2df: input을 dictionary type으로 변환하는 메서드  newdel은 {qX|sY|sZ(d(qX, sY) = sZ), qA|sB| (d(qA, sB) = e),...} 형식으로  입력된 input을 슬라이싱/스플릿하여 'X|Y|Z' (Z는 공백일 수 있음) 와 같은 형태의 string을 원소로 가지는 list이다  이 원소들을 다시 |을 매개로 스플릿하여 3개의 원소를 가지는 list로 만든 후, 몇 개의 검사를 거친다.  1. tmp[0](X)가 Qlst에 없거나, tmp[2](Z)이 공백이 아니면서 Qlst에 없다면 ErrorMessage(1)  2. tmp[1](Y)이 Slst에 없다면 ErrorMessage(2)  이 검사를 거친 후 Error 조건을 만족하지 않는다면, (X, Y) tuple을 value로, Z를 key로 가지는 원소들을 dic에 넣고 return한다. |
| Flstcheck | Flstcheck: 입력(Fin)을 검사하여 정상적이라면 list를, 아니라면 ErrorMessage를 return  입력을 슬라이싱/스플릿 한 후, set type을 형변환하여  F - Qlst가 공집합이라면 (F in Q) list를  공집합이 아니라면 (F not in Q) ErrorMessage(4)를 return한다 |
| act | act: DFA를 string을 통해 실행  DFA를 시뮬레이팅한다.  YES case는 Final state가 Flst의 원소일때  NO case는 Final state가 Flst의 원소가 아니거나, d(q\_n, s\_n) = e(epsilon)일 때 |
| YES, NO | YES: "네"를 출력하는 메서드  NO: "아니요"를 출력하는 메서드 |
| ErrorMessage | ErrorMessage: num에 따라 알맞은 오류 메시지를 출력하는 메서드  active를 False로 바꿔준다. num은 다음과 같은 case를 의미한다.  1: q\_n not in Q in delta function  2: s\_n not in SIGMA in delta function  3: q\_0 not in Q  4: F not in Q |

5. 출력 예시

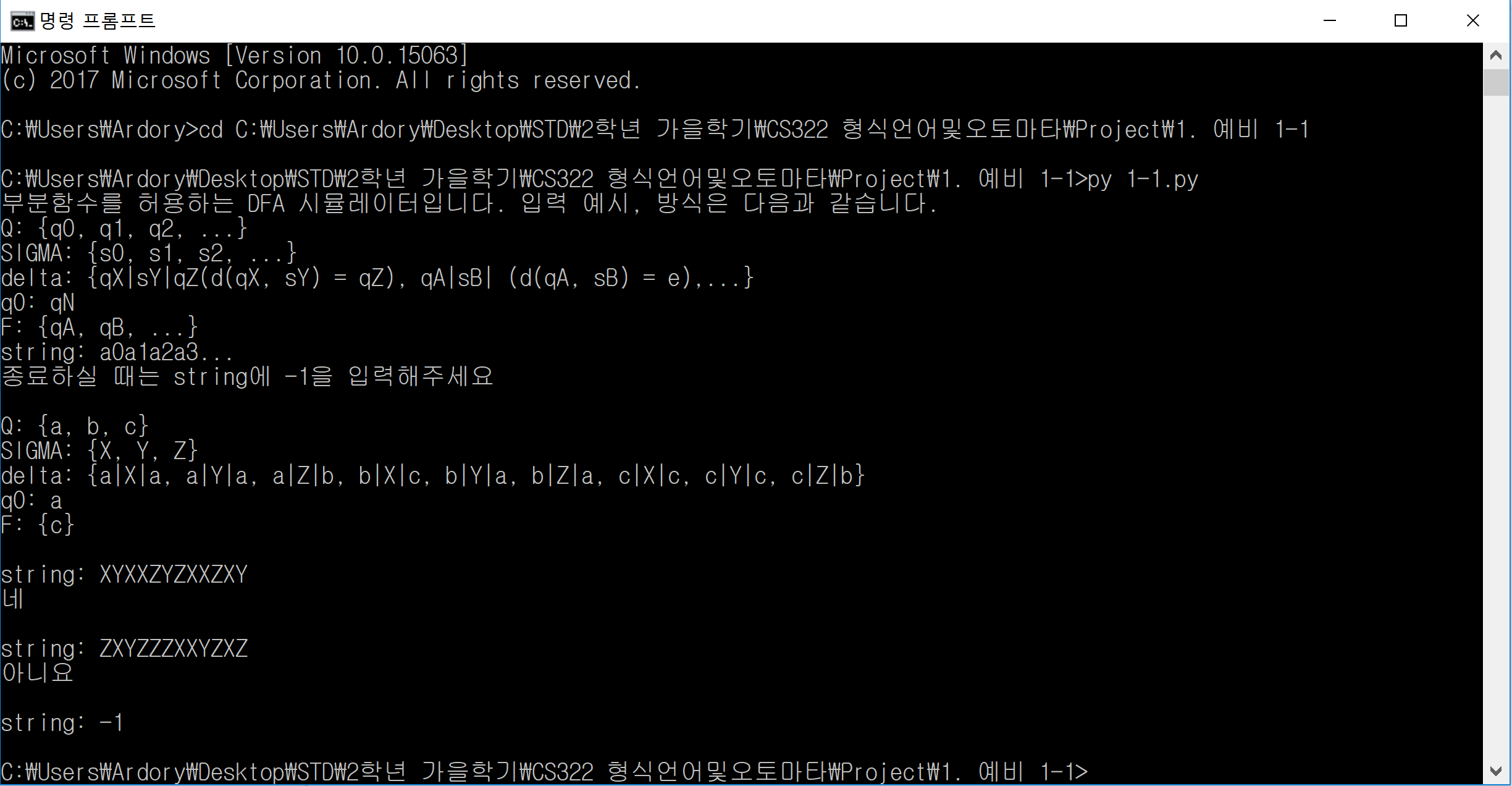
Case 1)



Case 2)

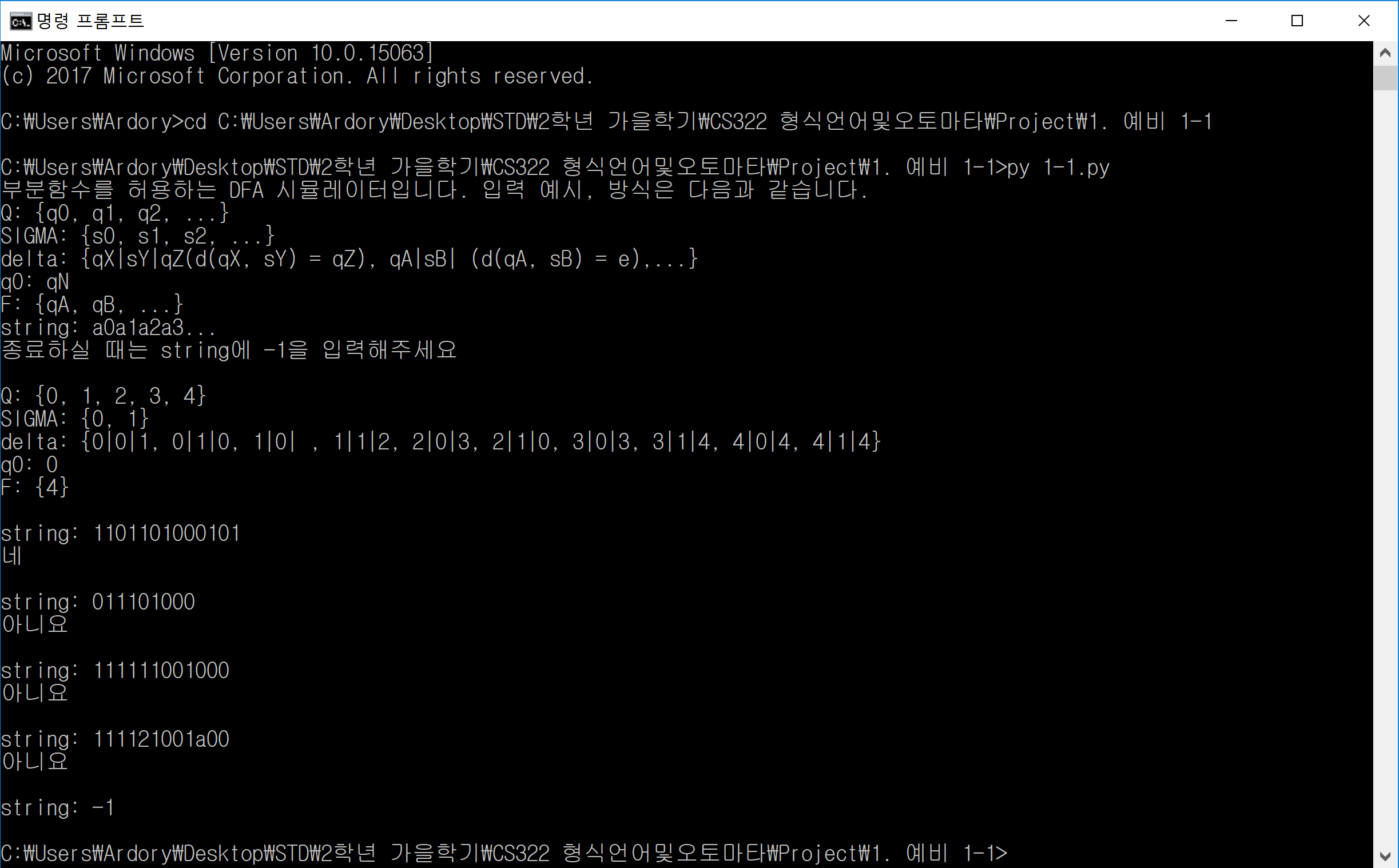


Case 1 출력 예시)

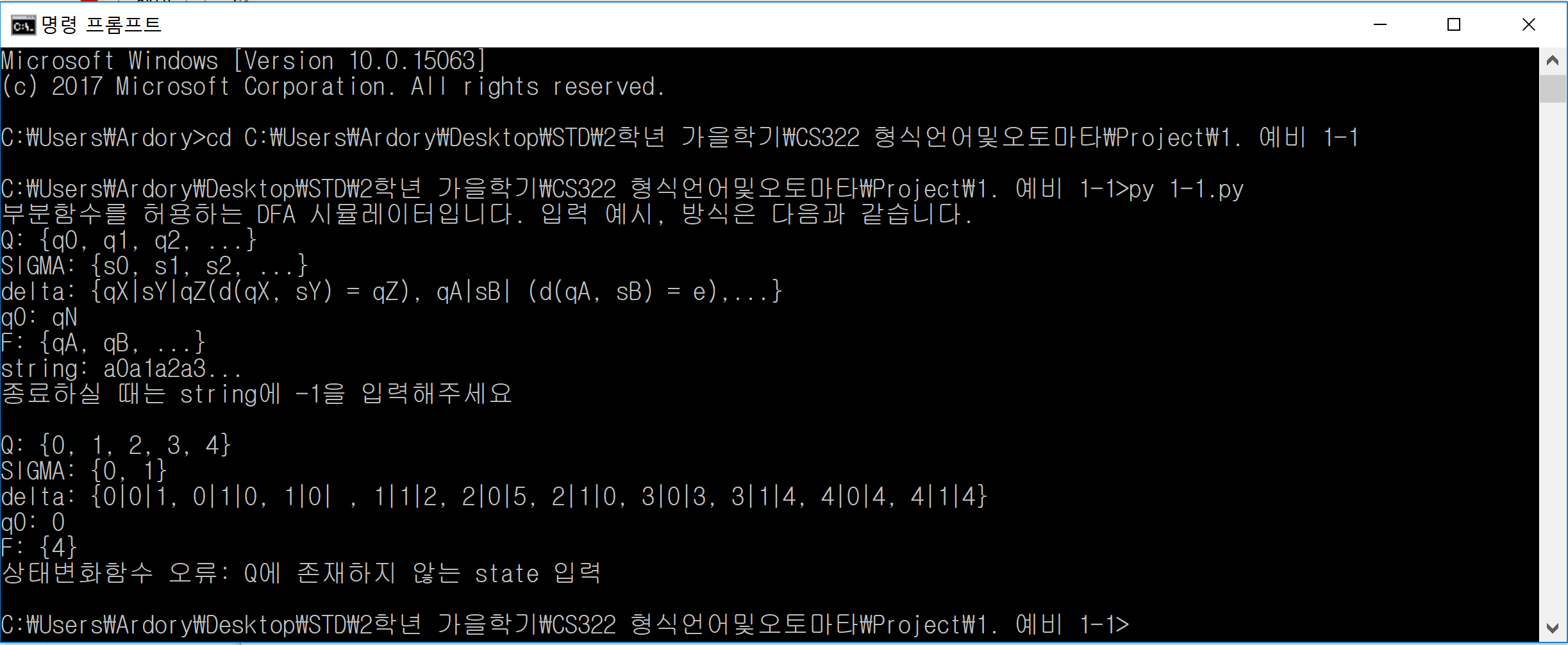


Case 2 출력 예시)

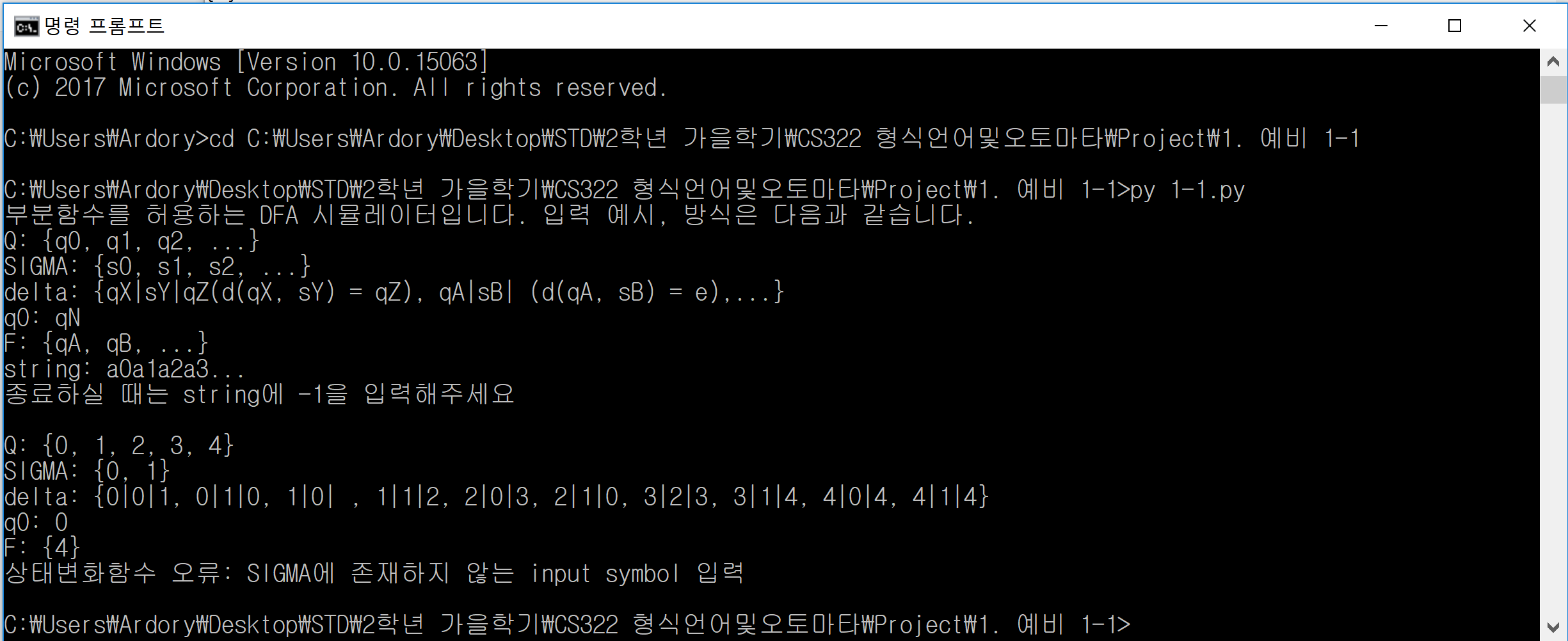
1. Normal case



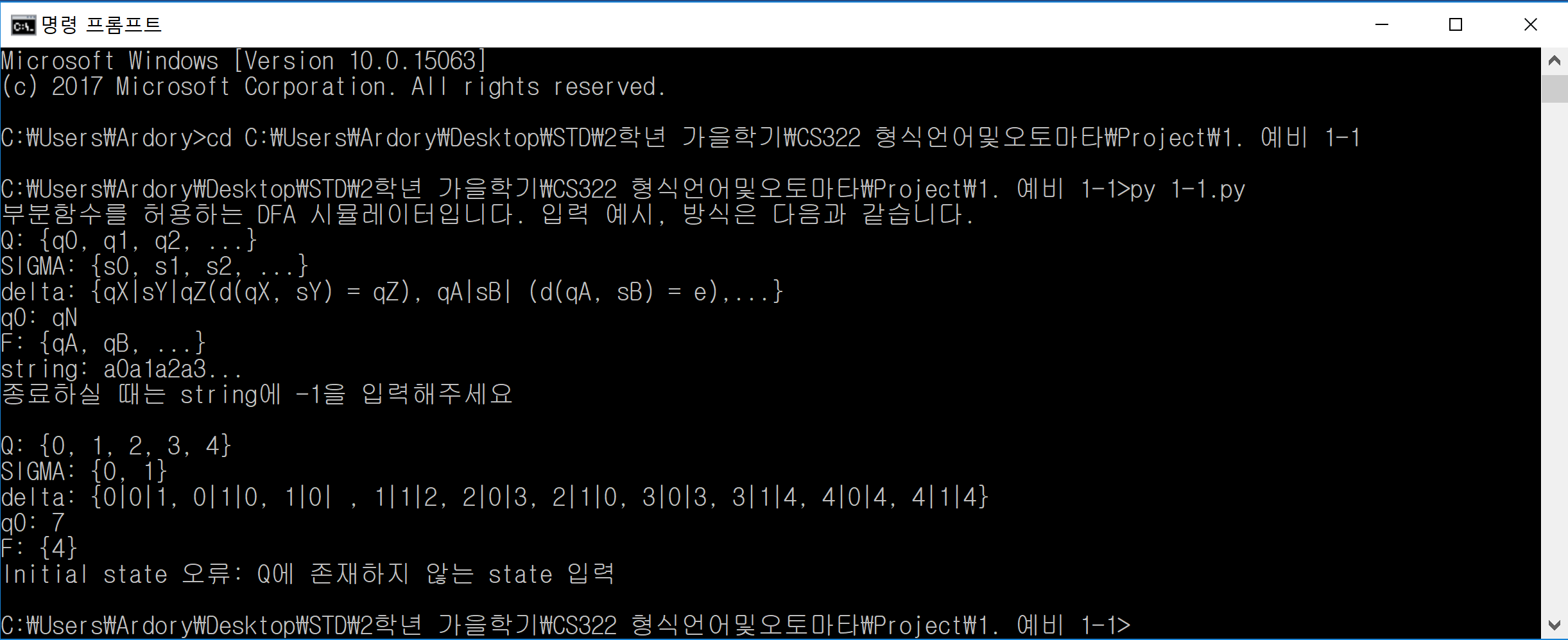
1. Error case
   1. Case 1:



* 1. Case 2:



* 1. Case 3:



* 1. Case 4:

