

토큰의 시작위치(st)와 종료위치(et) 변수를 만들고 한 단어가 끝나면 st 와 et 를 업데이트해 다음 단어를 찾아봅니다.

단어들을 파싱하기위해 먼저 문장의 전체 길이를 찾기로 했습니다.

문장의 전체 길이를 찾기위해 len 변수를 만들고 while 문을 사용해 NULL 이 나올때까지 len 을 한개씩 늘렸습니다.

그리고 토큰의 개수를 저장하는 t 변수와 토큰의 시작위치와 종료위치를 저장할 st(start tokens), et(end tokens)를 만들었습니다.

파싱하는 방법은 4 가지로 나뉘는데, command[i]가 공백인지아닌지, command[i+1]이 공백인지아닌지로 나뉘었습니다.

	isspace(command[i])	isspace(command[i + 1])		
1	T	T	i 가 공백이라면 시작위치를 현재 i 의 다음으로 업데이트합니다.	
2	T	F		
3	F	T		
			i 가 공백이 아니라면 et 를 현재 i 로 업데이트합니다.	i 가 공백이 아니고 i+1 이 공백이면 토큰의 한 부분이 끝났기때문에 토큰에 et-st + 2 만큼 공간을 할당하고 command 의 st 부터 et 까지를 tokens 에 저장합니다. 변수 t 를 하나 늘립니다.
4	F	F		x : et 업데이트만 합니다.

모든 문장이 파싱되고나면 nr_tokens 에 t 를 저장합니다.