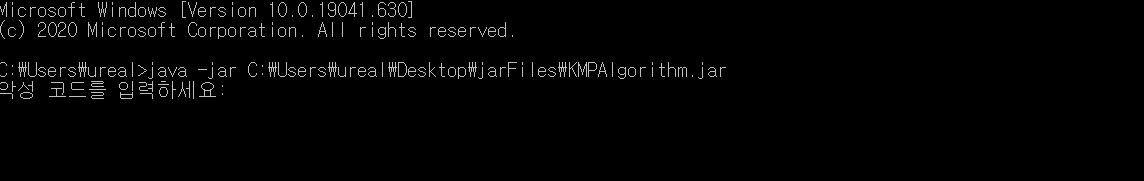
[ 멀티미디어 알고리즘 Report 1 ]

멀티미디어공학과

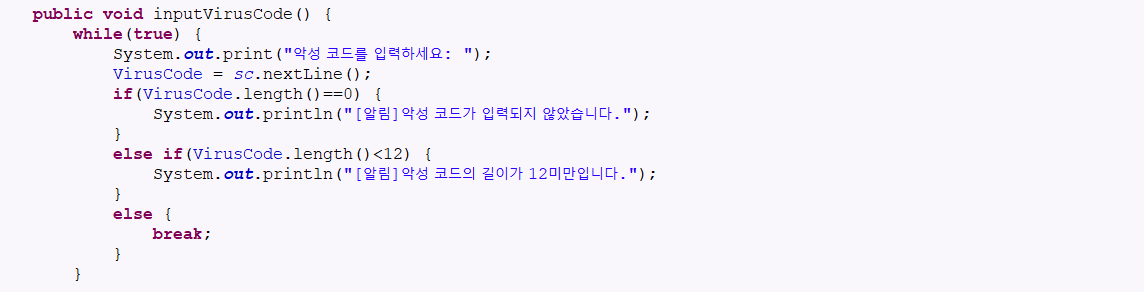
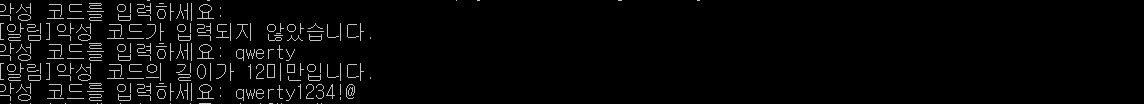
2019112541 양유진

1. 프로그램 설명

KMP 알고리즘과 Boyer Moore 알고리즘을 각각 이용하여 프로그램 코드를 분석하여 프로그램 내 악성코드 패턴을 탐지하는 프로그램이다. 프로그램은 텍스트 파일로 15개 이상의 프로그램 코드를 전달받으며, 악성코드와 악성코드에 대한 패턴의 길이를 사용자에게 입력 받아 악성코드 패턴을 분리한다. 프로그램 분석에서는 프로그램 코드에서 나타난 악성 코드가 패턴 마다 몇 번 나타나는 지, 바이러스 패턴을 가진 프로그램의 이름 및 개수, 바이러스 패턴을 가지지 않은 프로그램 이름 및 개수를 확인할 수 있다.

1. 실행
2. 입력

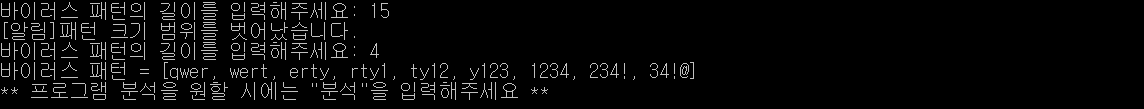
* 악성 코드



\* 예외처리: 입력 값이 없을 경우와 입력 값의 크기가 12 미만일 경우

악성코드는 qwerty1234!@로 입력해주었다.

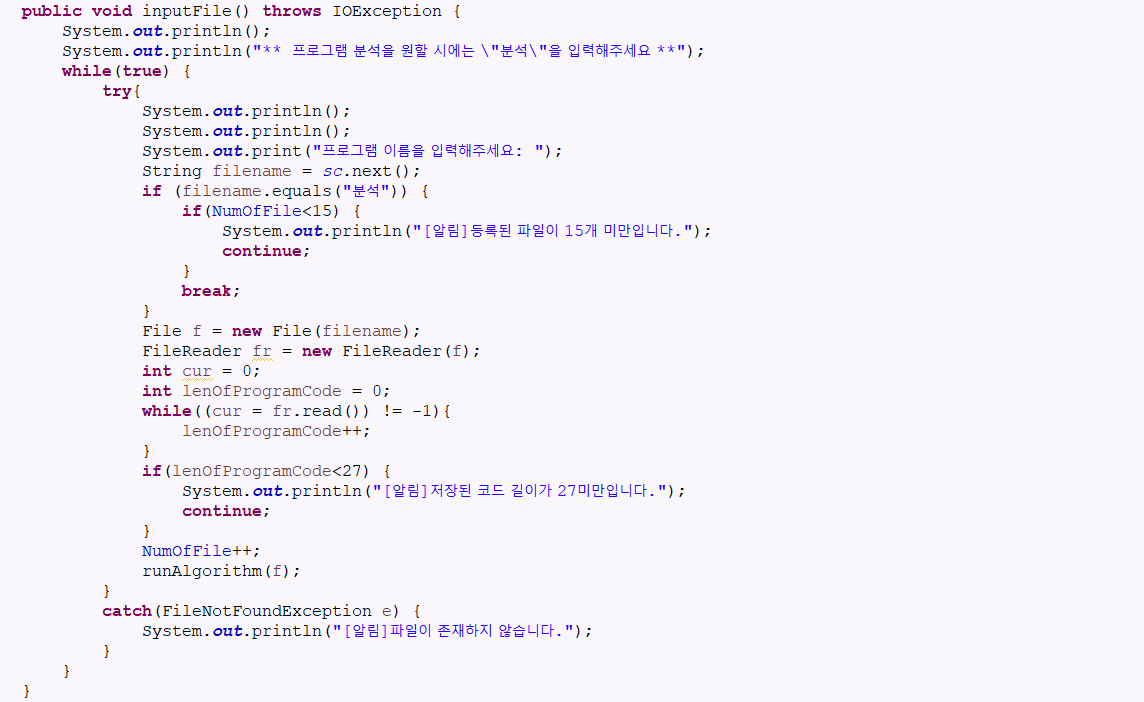
* 바이러스 패턴의 길이

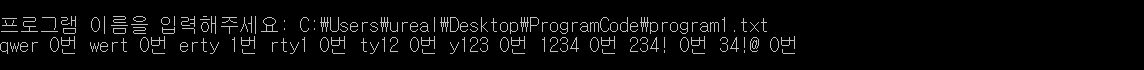


\* 예외처리: 패턴 크기 범위를 벗어난 입력

올바르게 모두 입력하였을 경우, 바이러스 패턴의 목록이 나타난다. 이 때, 바이러스 패턴의 길이는 4로 입력해주었다.

* 프로그램 코드





프로그램 이름을 입력하면, 텍스트 파일 안에 있는 프로그램 코드를 분석하여 각 패턴 당 몇 번 매칭되는지 나타난다.

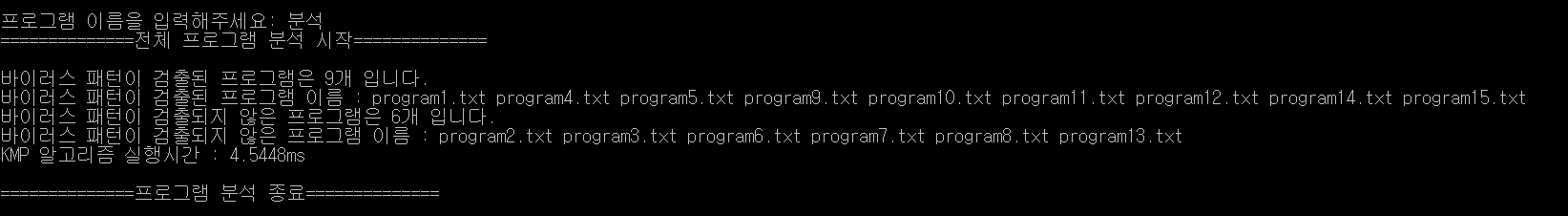


\* 예외처리: 파일이 존재하지 않을 경우

- 등록된 파일이 15개 미만일 경우

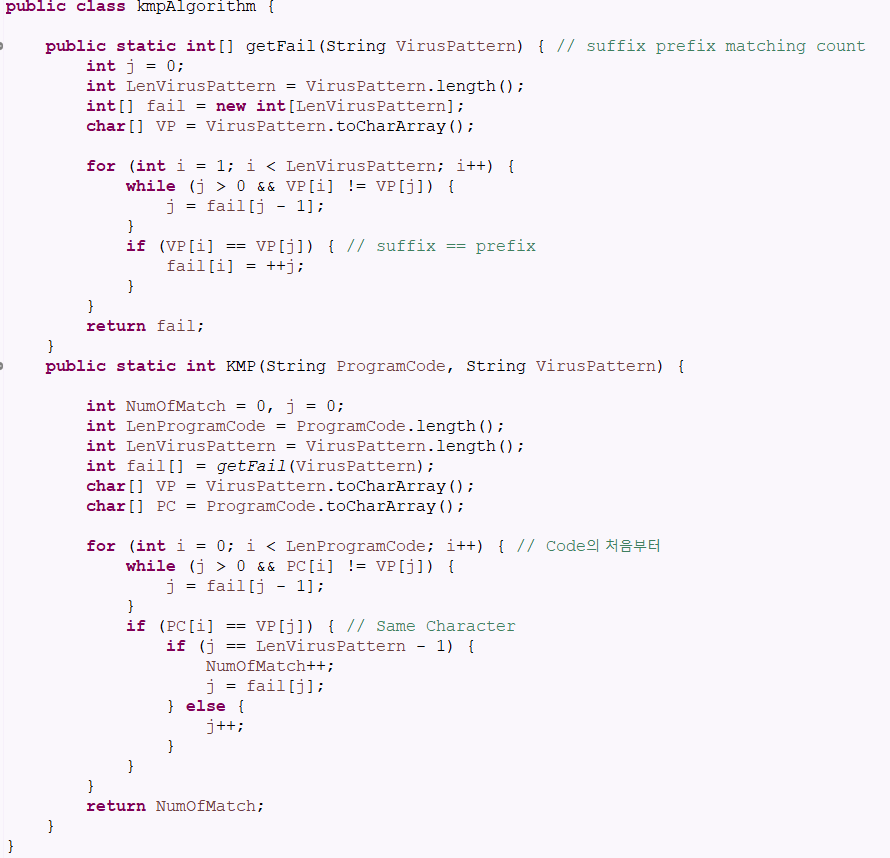
- 저장된 코드 길이가 27미만일 경우

1. 출력



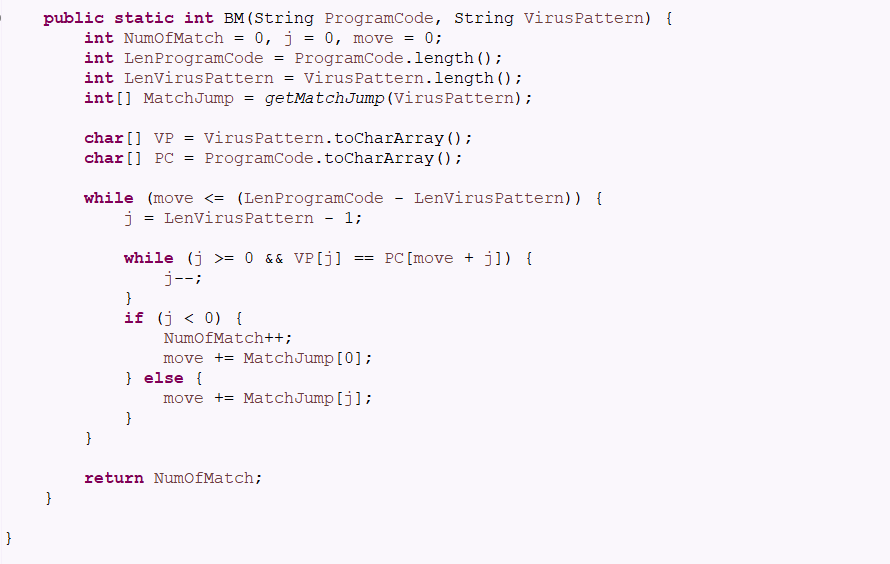
프로그램명 대신 “분석”을 입력하면 다음과 같이 입력한 전체 프로그램의 악성코드 패턴 매칭 여부와 실행 시간을 분석해준다.

1. 알고리즘 설명
2. KMP Algorithm



getFail() 메소드에서 패턴의 suffix와 prefix를 이용하여 fail값을 구해 배열에 넣어준다. Suffix 와 prefix를 비교하여 패턴의 해당 길이 (j)를 카운트해준다. KMP메소드에서는 문자를 비교하여 해당 인덱스가 패턴의 끝과 일치하면 매칭이 성공된 것으로 간주한다. 매칭이 성공될 때마다 NumOfMatch를 하나씩 증가시켜 패턴이 몇 번 매칭되는지를 파악한다.

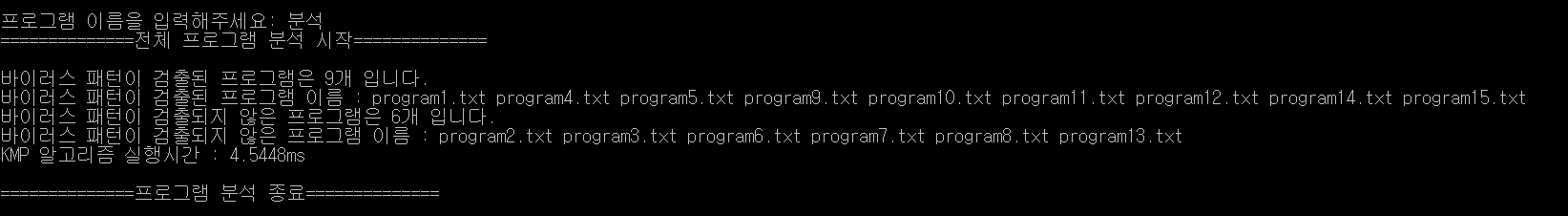
1. Boyer Moore



BM old idea의 MatchJump를 이용하여 알고리즘을 구현하였다. getMatchJump 메소드에서는 패턴이 얼마나 움직여야 하는지를 나타내는 MatchJump의 값을 구한다. BM 메소드에서는 구해놓은 MatchJump 값으로 패턴을 이동시킨다.

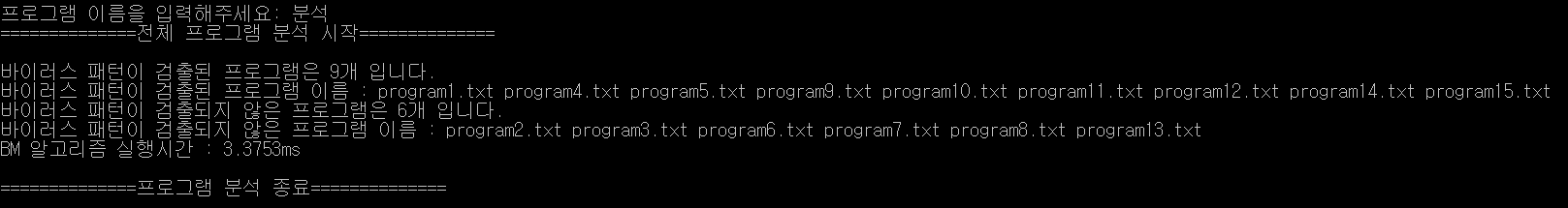
1. 실행 결과 및 분석

[1] KMP Algorithm

- 매칭 알고리즘 실행 시간 : 4.5448ms

[2] Boyer Moore Algorithm

- 매칭 알고리즘 실행 시간 : 3.3753ms



[3] 결론

평균적으로 boyer-moore가 KMP보다 빠르지만. DNA 염기 서열처럼 적은 수의 문자를 이용한 코드를 비교해보면 kmp가 더 유리하다고 한다. 실제로 비슷한 문자로만 프로그램 코드를 작성하여 비교해보니 kmp가 boyer-moore보다 훨씬 빠른 것을 확인할 수 있었다. 다른 프로그램 코드에는 악성코드 패턴에 잘 나오지 않는 문자들을 섞어주었더니, boyer moore 알고리즘이 kmp보다 빠른 것을 확인할 수 있었다.