

# 과제 1

텍스트 에디터 작성

# 줄 단위 출력

- 1) 명령 줄 인수로 받은 텍스트 파일을 검사하여 사용자가 원하는 줄을 출력하는 프로그램을 작성하시오.
- 2) 이 프로그램은 시작하면 프롬프트를 내주며 사용자가 원하는 줄 번호를 입력 받아 해당 줄을 줄 번호와 함께 출력한다.
- 3) 줄 번호는 다음과 같은 4가지 형태로 입력할 수 있다.

n    한 줄 번호

n, ..., m    줄 번호 리스트

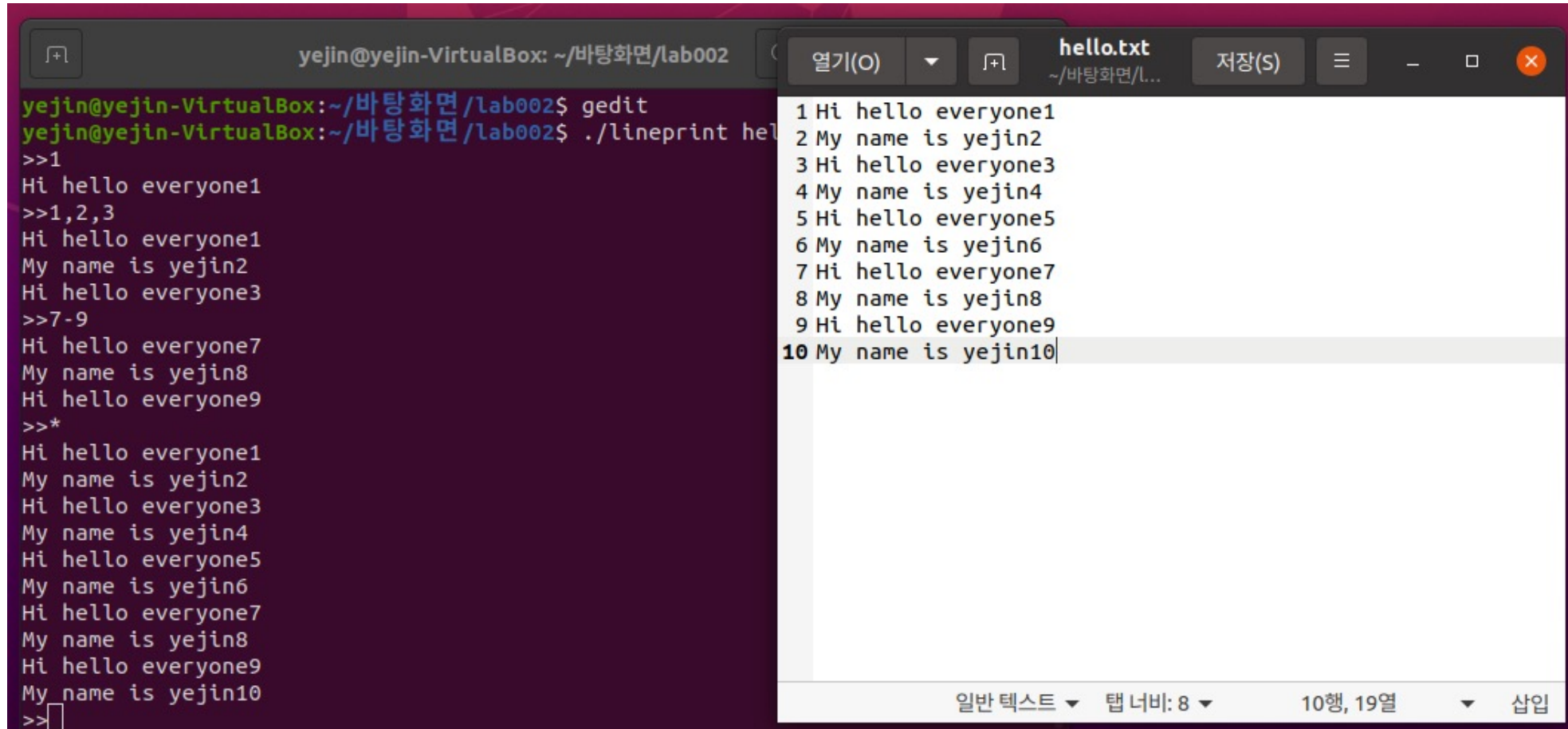
n-m    줄의 범위

\*    모든 줄

## ※힌트※

먼저 파일 내의 각 줄의 시작 위치와 길이를 저장하는 테이블을 만들고 이를 이용해서 해당 줄을 찾아서 읽은 후에 출력한다.

# 줄 단위 출력



The image shows a terminal window and a gedit editor window. The terminal window, titled 'yejin@yejin-VirtualBox: ~/바탕화면/lab002', shows the execution of the 'gedit' command and the './lineprint' script. The script outputs lines 1 through 10 of the file 'hello.txt' in a loop. The gedit editor window, titled 'hello.txt', shows the content of the file, which consists of 10 lines of text. The status bar at the bottom of the gedit window indicates '일반 텍스트' (Plain Text), '탭 너비: 8' (Tab width: 8), '10행, 19열' (10 lines, 19 columns), and '삽입' (Insert).

```
yejin@yejin-VirtualBox: ~/바탕화면/lab002$ gedit
yejin@yejin-VirtualBox: ~/바탕화면/lab002$ ./lineprint hello.txt
>>1
Hi hello everyone1
>>1,2,3
Hi hello everyone1
My name is yejin2
Hi hello everyone3
>>7-9
Hi hello everyone7
My name is yejin8
Hi hello everyone9
>>*
Hi hello everyone1
My name is yejin2
Hi hello everyone3
My name is yejin4
Hi hello everyone5
My name is yejin6
Hi hello everyone7
My name is yejin8
Hi hello everyone9
My name is yejin10
>>
```

hello.txt  
~ /바탕화면/l...

열기(O) ▼ [icon] 저장(S) [icon] [icon] [icon]

```
1 Hi hello everyone1
2 My name is yejin2
3 Hi hello everyone3
4 My name is yejin4
5 Hi hello everyone5
6 My name is yejin6
7 Hi hello everyone7
8 My name is yejin8
9 Hi hello everyone9
10 My name is yejin10
```

일반 텍스트 ▼ 탭 너비: 8 ▼ 10행, 19열 ▼ 삽입

# 라인 테이블

File

30	75	102	156	
\n	\n	\n	\n	...

## ■ 라인 테이블 작성

- table의 index → 줄의 번호
- table의 첫번째 column → 줄의 시작 위치
- table의 두번째 column → 줄의 길이

0	0	30
1	30	45
2	75	27
3	102	54
4	156	
	...	...
	...	...
	...	...

index가 줄의 번호

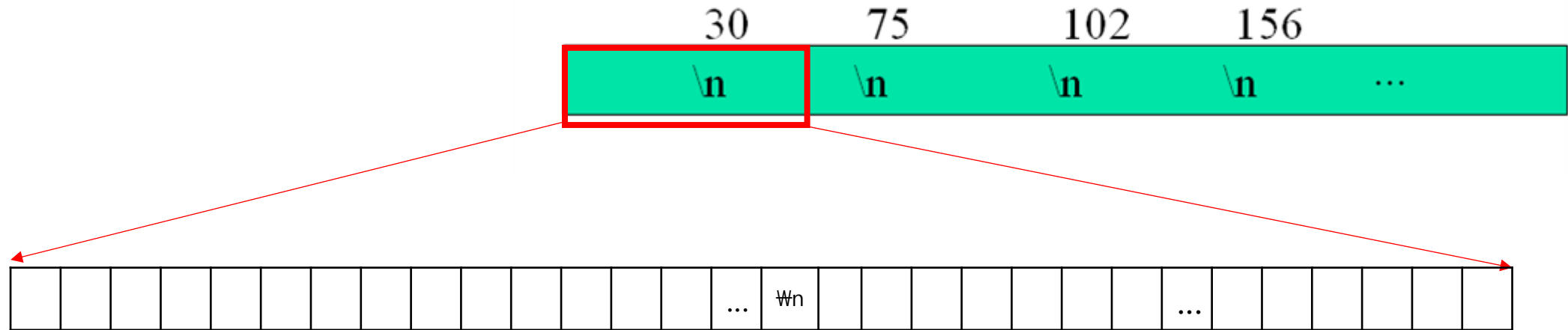
줄의 시작 위치

줄의 길이

# 라인 테이블 만들기

- fread 함수

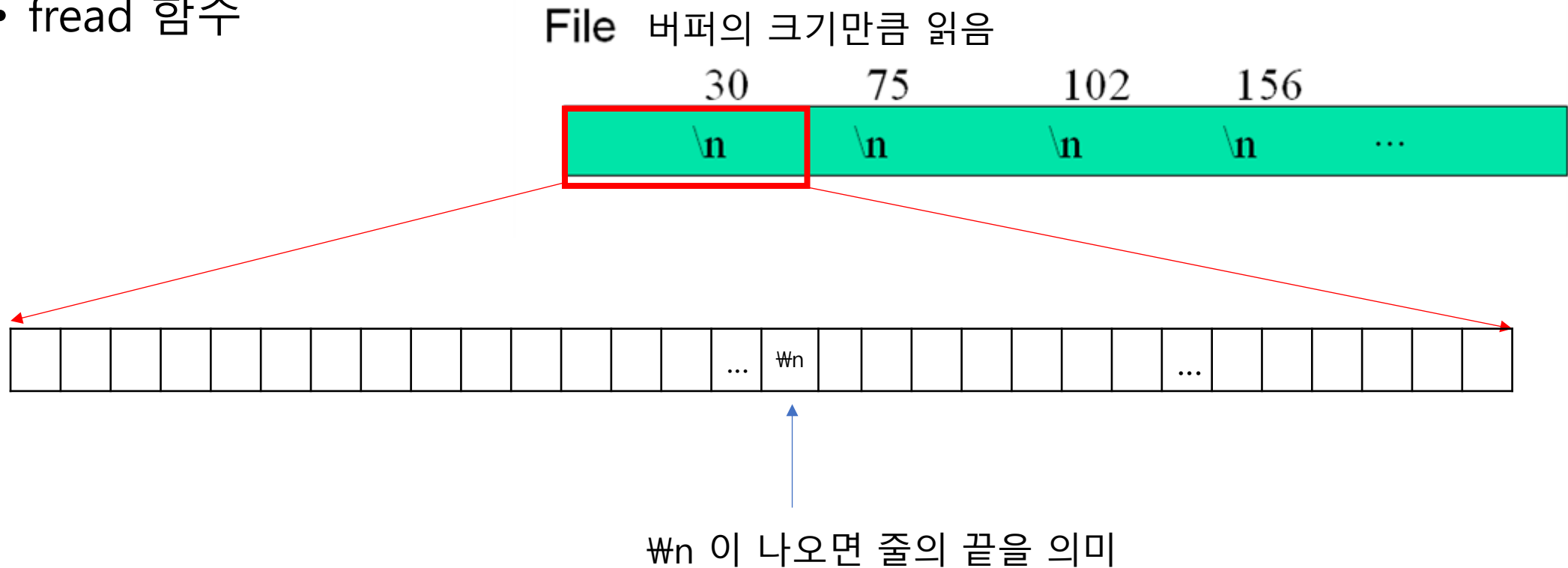
File 버퍼의 크기만큼 읽음



char 를 하나씩 비교

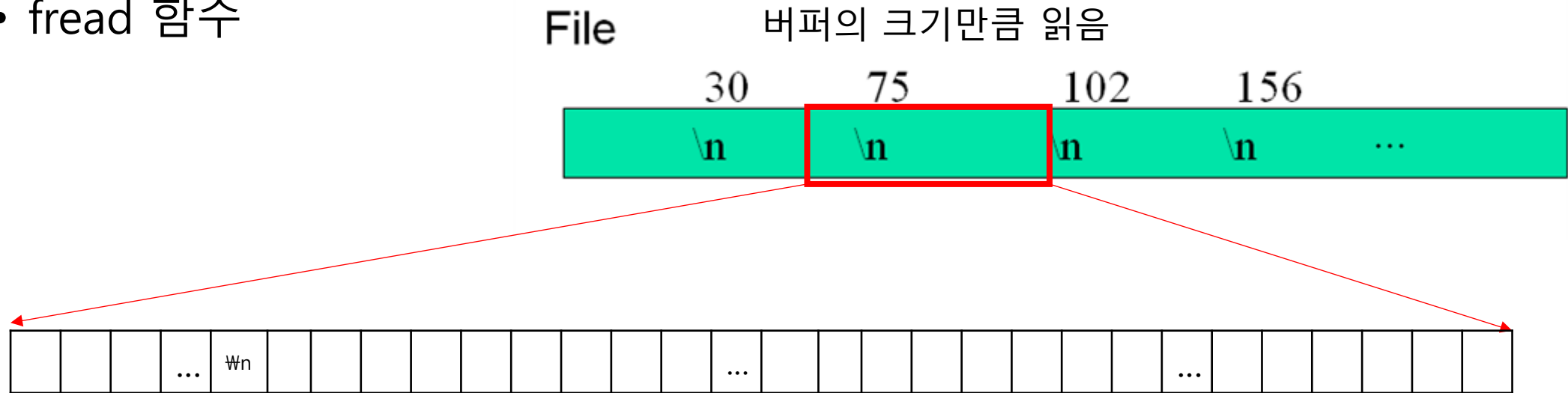
# 라인 테이블 만들기

- fread 함수



# 라인 테이블 만들기

- fread 함수



# 줄 단위 출력

```
user@localhost:~  
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)  
[user@localhost ~]$ ./lineprint hello.txt  
>> 1,2  
[1] hi hello everyone  
[2] my name is apple  
>> 1-2  
[1] hi hello everyone  
[2] my name is apple  
>> 2  
[2] my name is apple  
>> *  
[1] hi hello everyone  
[2] my name is apple  
[3] nice to meet you apple  
>> █
```

프롬프트로 입력 기다리기  
- q(quit을 의미)문자를 입력하기 전까지 프롬프트로 계속 입력 받기



# 줄 단위 출력 : 힌트

```
user@localhost:~  
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)  
[user@localhost ~]$ ./lineprint hello.txt  
>> 1,2 —  
[1] hi hello everyone  
[2] my name is apple  
>> 1-2  
[1] hi hello everyone  
[2] my name is apple  
>> 2  
[2] my name is apple  
>> *  
[1] hi hello everyone  
[2] my name is apple  
[3] nice to meet you apple  
>> █
```

프롬프트로 입력받는 부분

1	,	2	NULL	...			
---	---	---	------	-----	--	--	--

하나씩 읽어들이어서

- 한 줄 번호
- 줄 번호 리스트
- 줄의 범위
- 모든 줄

의 4가지 케이스 중 무엇인지 판별

# 줄 단위 출력 : 힌트

The screenshot shows a code editor with two tabs: 'hello.txt' and 'lineprint.c'. The 'hello.txt' tab is active and contains the following text:

```
hi hello everyone1
my name is apple1
nice to meet you apple1
hi hello everyone3
my name is apple3
nice to meet you apple3
hi hello everyone4
my name is apple4
nice to meet you apple4
hi hello everyone5
my name is apple5
nice to meet you apple5
hi hello everyone6
my name is apple6
nice to meet you apple6
hi hello everyone7
my name is apple7
nice to meet you apple7
```

The 'lineprint.c' tab is also visible. Below the code editor, a terminal window titled 'user@localhost:~' is open. It shows the command `./lineprint hello.txt` being executed. The terminal output is as follows:

```
[user@localhost ~]$ ./lineprint hello.txt
>> 10-11
[10] hi hello everyone5
[11] my name is apple5
>> 2-9
[2] my name is apple1
[3] nice to meet you apple1
[4] hi hello everyone3
[5] my name is apple3
[6] nice to meet you apple3
[7] hi hello everyone4
[8] my name is apple4
[9] nice to meet you apple4
>> 1, 10, 12
[1] hi hello everyone1
[10] hi hello everyone5
[12] nice to meet you apple5
>> *
[1] hi hello everyone1
[2] my name is apple1
[3] nice to meet you apple1
[4] hi hello everyone3
[5] my name is apple3
[6] nice to meet you apple3
```

→ 프롬프트로 입력받는 부분

1	0	-	1	1	NULL	...	
---	---	---	---	---	------	-----	--

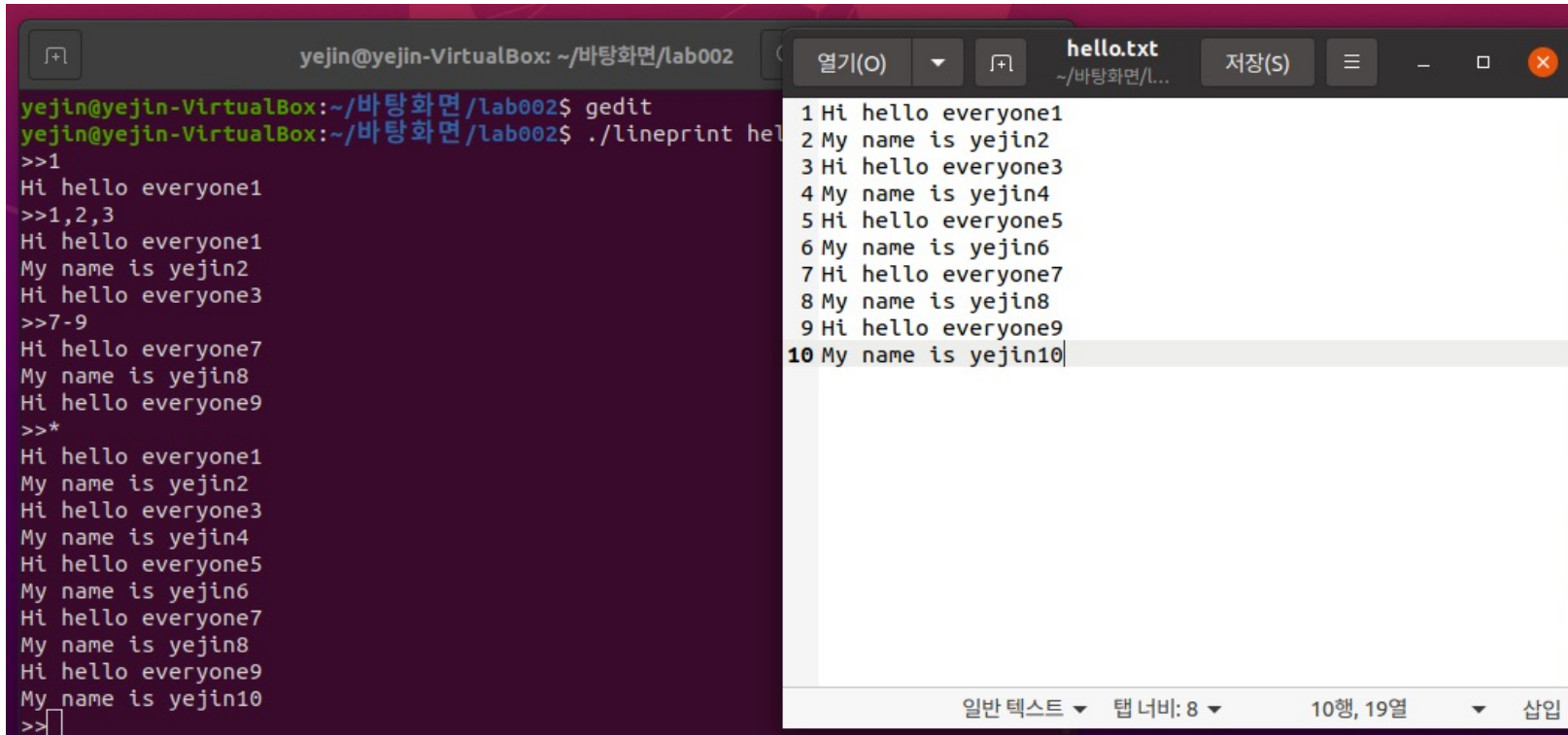
어떻게 프롬프트로 받은  
두 자리 수 이상의 수를  
줄 번호로 인식할지 고려  
해서 프로그램 작성하기

# 참고) strchr, strtok 사용하기

P.196

- strchr : 문자열 내에서 처음 나타난 특정 문자 c에 대한 포인터를 반환한다. 없으면 NULL을 반환한다.
- strtok : 문자열 내의 문자들을 구분자로 사용하여 문자열을 토큰들로 나누어준다.

# 캡처 예시



The screenshot shows a terminal window on the left and a text editor window on the right. The terminal window title is 'yejin@yejin-VirtualBox: ~/바탕화면/lab002'. The text editor window title is 'hello.txt ~/바탕화면/l...'. The terminal shows the following commands and output:

```
yejin@yejin-VirtualBox:~/바탕화면/lab002$ gedit
yejin@yejin-VirtualBox:~/바탕화면/lab002$ ./lineprint hel
>>1
Hi hello everyone1
>>1,2,3
Hi hello everyone1
My name is yejin2
Hi hello everyone3
>>7-9
Hi hello everyone7
My name is yejin8
Hi hello everyone9
>>*
Hi hello everyone1
My name is yejin2
Hi hello everyone3
My name is yejin4
Hi hello everyone5
My name is yejin6
Hi hello everyone7
My name is yejin8
Hi hello everyone9
My name is yejin10
>>
```

The text editor shows the following content in 'hello.txt':

```
1 Hi hello everyone1
2 My name is yejin2
3 Hi hello everyone3
4 My name is yejin4
5 Hi hello everyone5
6 My name is yejin6
7 Hi hello everyone7
8 My name is yejin8
9 Hi hello everyone9
10 My name is yejin10
```

The status bar at the bottom of the text editor shows '일반 텍스트', '탭 너비: 8', '10행, 19열', and '삽입'.

## 보여져야 하는 내용

- 명령 줄 인수로 넣어준 파일 내용(ex. hello.txt 최소 10줄 이상)
- 종료(q) 전까지는 계속 입력 받기
- 프롬프트로 총 4가지를 입력 받은 결과를 캡처
  - 1) 한 줄 번호
  - 2) 줄 번호 리스트(최소 3개 이상 입력)
  - 3) 줄의 범위 (최소 3 줄 이상)
  - 4) 모든 줄

# 라인 에디터로 확장

1. 줄 단위 출력 프로그램을 간단한 편집 기능을 갖는 라인 에디터로 확장

2. 편집 명령어 설계

- 프린트
  - p 줄번호
- 줄 삭제
  - d 줄번호
- 줄 추가
  - a 줄번호
  - 줄번호 이후에 추가할 줄 입력
- 줄 대치
  - s 줄번호
  - 해당 줄을 대치할 줄 입력