

# 리눅스

## ■ 1. 리눅스가 무엇인지 ?

## ■ 2. 리눅스 기본 명령어

- <a href="#">cd 명령어</a>	<a href="#">문제 1 ~ 7</a>
- <a href="#">touch 명령어</a>	<a href="#">문제 8 ~ 10</a>
- <a href="#">mkdir 명령어</a>	<a href="#">문제 11</a>
- <a href="#">rm 명령어</a>	<a href="#">문제 12 ~ 14</a>
- <a href="#">rmdir명령어</a>	<a href="#">문제 15 ~ 16</a>
- <a href="#">alias명령어</a>	<a href="#">문제 17 ~ 18</a>
- <a href="#">cat명령어</a>	
- <a href="#">redirection명령어</a>	<a href="#">문제 20 ~ 23</a>
- <a href="#">more 명령어</a>	
- <a href="#">head 명령어</a>	<a href="#">문제 24</a>
- <a href="#">wc 명령어</a>	<a href="#">문제 25 ~ 28</a>
- <a href="#">grep 명령어</a>	<a href="#">문제 29 ~ 33</a>
- <a href="#">awk 명령어</a>	<a href="#">문제 34 ~ 41</a>
- <a href="#">sort 명령어</a>	<a href="#">문제 42 ~ 54</a>
- <a href="#">uniq 명령</a>	<a href="#">문제 55 ~ 62</a>
- <a href="#">echo 명령어</a>	<a href="#">문제 63 ~ 72</a>
- <a href="#">diff 명령어</a>	<a href="#">문제 73</a>
- <a href="#">find 명령어</a>	<a href="#">문제 74 ~ 76</a>
- <a href="#">tar 명령어</a>	<a href="#">문제 77 ~ 80</a>
- <a href="#">ln(link) 명령어</a>	<a href="#">문제 81</a>
- <a href="#">sed 명령어</a>	<a href="#">문제 82</a>
- <a href="#">cp 명령어</a>	<a href="#">문제 83</a>
- <a href="#">mv 명령어</a>	<a href="#">문제 84</a>

## ■ 3. vi 편집기 명령어

- <a href="#">편집기를 시작하는 명령어</a>	
- <a href="#">vi 편집기 내에서 커서 이동</a>	<a href="#">문제 85</a>
- <a href="#">vi 편집기의 입력 명령어</a>	
- <a href="#">vi 편집기의 삭제 명령어</a>	<a href="#">문제 86</a>
- <a href="#">취소 명령어</a>	
- <a href="#">복사/붙여넣기 명령어</a>	<a href="#">문제 87 ~ 90</a>
- <a href="#">vi 편집기 내에서 문자를 변경하는 명령어</a>	<a href="#">문제 91 ~ 93</a>
- <a href="#">리눅스 셸 스크립트를 디버깅 모드로 실행하는 방법</a>	<a href="#">문제 94 ~ 96</a>

## ■ 4. 권한 관리

- <a href="#">권한을 설정하는 숫자</a>	<a href="#">문제 97 ~ 104</a>
- <a href="#">chown 명령어</a>	<a href="#">문제 105 ~ 106</a>
	<a href="#">문제 107 ~ 110</a>

## ■ 5. 디스크 관리

- <a href="#">du 명령어</a>	<a href="#">문제 111</a>
- <a href="#">df 명령어</a>	
- <a href="#">sar 명령어</a>	<a href="#">문제 112 ~ 117</a>

## ■ 6. 프로세서 관리

- [jobs 명령어](#)
- [top명령어](#)
- [ps 명령어](#)
- [kill 명령어](#)

[문제 118 ~ 120](#)  
[문제 121](#)  
[문제 122 ~ 125](#)  
[문제 126](#)

## ■ 7. 쉘 스크립트 작성법

- [쉘 스크립트를 작성하고 실행하는 방법](#)
- [변수 사용법](#)
- [변수의 숫자 계산하는 방법](#)
- [파라미터 변수](#)
- [리눅스 쉘에서 if문 사용방법](#)
- [if 문 조건문에 들어가는 비교 연산자](#)
- [리눅스에서 사용하는 연산자](#)
- [loop 문](#)
- [sed 명령어로 공백라인을 지우는 방법 \(데이터 정제\)](#)
- [while loop문](#)

[문제 127](#)  
  
[문제 128 ~ 129](#)  
[문제 130 ~ 131](#)  
  
[문제 132 ~ 133](#)  
[문제 134 ~ 135](#)  
[문제 136 ~ 145](#)  
[문제 146 ~ 156](#)  
[문제 157 ~ 163](#)

## ■ [파일과 관련된 조건](#)

[문제 164 ~ 169](#)

## ■ [데이터 분석을 위해 만들어야할 리눅스 쉘 스크립트](#)

### ■ [오라클의 data를 csv 로 생성하는 방법](#)

[문제 170](#)

- [리눅스 sql에서 ed로 vi 편집기가 열리게 하는 방법](#)
- [리눅스의 emp테이블의 데이터를 csv로 생성하는 방법](#)

[문제 171 ~ 177](#)

### ■ [case문](#)

[문제 178 ~ 186](#)

### ■ [유저 생성 및 관리](#)

- [useradd명령어](#)
- [passwd 명령어](#)
- [usermod 명령어](#)
- [userdel 명령어](#)
- [groupadd 명령어](#)
- [groupmod 명령어](#)
- [gpasswd 명령어](#)

[문제 187](#)  
[문제 188 ~ 190](#)  
[문제 191 ~ 194](#)  
[문제 195 ~ 196](#)

[문제 197](#)

## ■ [문제 모음](#)



## \* 리눅스를 왜 배워야 하는가 ?

1. 현업의 대용량 데이터는 대부분 window가 아니라 리눅스 나 유닉스 시스템에 저장되어 있다.
2. 하둡이 리눅스 시스템에 설치 된다.
3. 딥러닝을 구현하는 환경의 대부분이 리눅스 환경이다.

## \* 리눅스가 무엇인가?

유닉스가 너무 고가여서 리눅스 오픈소스를 핀란드의 리누즈 토발즈 학생이 1991년 11월에 개발을 했다.

리눅스 토발즈가 리눅스의 커널(자동차 엔진)을 개발하고 소스를 무료로 공개하고 전세계의 많은 개발자들이 이 오픈소스를 가져다가 더 좋게 개선을 해서 다시 인터넷에 올리고 하는 작업을 반복하다보니 리눅스 os가 유닉스 보다 더 가볍고 안정적이게 되었다.

GNU프로젝트 → 누구든지 배포된 오픈소스를 가져다가 개발할 수 있고 돈을 벌 목적으로 상용화를 할 수도 있는데 한가지 지켜야 할 약속은 이 소스를 가져다가 더 좋게 수정했으면 그 코드를 인터넷에 올려줘야 하는 약속이 있다.

레드햇(상용버전) -----> Cent os  
유료                      무료

1. Oracle linux
2. Cento os
3. ubuntu

## \* 리눅스 시스템 접속 및 환경 구성하는 방법

## VirtualBox-5.0.14-105127-Win 설치

비주열 창에서 11gWS2 실행

안된다면 <https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html> 에서

## putty-64bit-0.70-installer.msi 깔아봐

putty 시작프로그램에서 찾아서 실행

호스트에 192.168.19.77 입력하고 실행

test123/test 우리 gpu pc 서버에 접속하는 것

파이썬 접속 명령어 : python3

나가기 : exit()

## ■ 2. 리눅스 기본 명령어

### ★ 1. cd 명령어

#### 문제 1 ~ 7

"Change Directory 명령어로 디렉토리를 이동하는 명령어"

예제 : `$ pwd` <----- 현재 내가 있는 디렉토리 확인

↓

print working directory의 약자

```
test123@woody-dev:~$ pwd
/home/test123
```

`$ ls` <----- 현재 디렉토리에 있는 폴더와 파일을 확인

↓

list 의 약자

```
test123@woody-dev:~$ ls
labs                <----- 파란색이면 폴더
```

`$ cd labs` <----- labs디렉토리로 이동

```
test123@woody-dev:~/labs$ pwd
/home/test123/labs
```

### ★ 2. touch 명령어

#### 문제 8 ~ 10

"파일의 용량이 0인 파일을 생성하는 명령어"

예 : `$ touch a1.txt`

`$ ls -l a1.txt`

```
test123@woody-dev:~/test3$ touch a3.txt
test123@woody-dev:~/test3$ ls
a1.txt  a3.txt
test123@woody-dev:~/test3$ ls -l a3.txt
-rw-rw-r-- 1 test123 test123 0 12월 14 11:58 a3.txt
```

### ★ 3. mkdir 명령어

#### 문제 11

"디렉토리를 만드는 명령어"

예제 :

```
heaven@buzz:~$ mkdir oracle_test
heaven@buzz:~$ ls -dir oracle_test
286449 oracle_test
heaven@buzz:~$ cd oracle_test
heaven@buzz:~/oracle_test$ pwd
/home/heaven/oracle_test
```

★ 4. rm 명령어

문제 12 ~ 14

"파일이나 디렉토리를 삭제하는 명령어"

★★★★★ 주의 사항 !!!!!

리눅스나 유닉스는 휴지통이 없기 때문에 삭제할 때 특히 주의해야 한다 !!

삭제할 때 특히 주의해야 한다 !!!

백업이 없으면 절대 복구가 불가능하다 !!

예제 :

```
heaven@buzz:~/oracle_test/test/test2/test3$ touch bbb.txt
heaven@buzz:~/oracle_test/test/test2/test3$ ls -l bbb.txt
-rw-rw-r-- 1 heaven heaven 0 Dec 14 05:38 bbb.txt
heaven@buzz:~/oracle_test/test/test2/test3$ rm bbb.txt
heaven@buzz:~/oracle_test/test/test2/test3$ ls -l bbb.txt
ls-l: command not found
```

★ 5. rmdir 명령어

문제 15 ~ 16

"디렉토리를 삭제하는 명령어"

예 :

```
heaven@buzz:~/oracle_test$ mkdir ddd
heaven@buzz:~/oracle_test$ ls
ddd
heaven@buzz:~/oracle_test$ rmdir ddd
heaven@buzz:~/oracle_test$ ls
```

★ 6. alias명령어

문제 17 ~ 18

"자주 수행하는 명령어들을 쉽게 사용할 수 있도록 설정하는 명령어"

\* 오라클 db에 접속

```
heaven@buzz:~$ sqlplus / as sysdba
```

\* python 접속

```
heaven@buzz:~$ python3
```

```
heaven@buzz:~$ alias p='python3'
```

```
heaven@buzz:~$ p
```

★ 7. cat 명령어

"파일의 내용을 화면에 출력하는 명령어"

예제 :

```
heaven@buzz:~$ cat 파일명.txt
heaven@buzz:~$ vi emp.txt
```

★ 8. redirection 명령어

문제 20 ~ 23

"화면에 출력되는 결과를 파일로 저장하는 명령어"

>> : 없으면 파일을 생성하고 있으면 기존 파일 뒤에 덧붙이겠다.  
> : 파일을 생성하겠다. 있으면 그냥 덮어쓰겠다

예 : \$ cat emp.txt >> emp50.txt

지금 보고있는 emp.txt의 화면의 결과를 emp50.txt 로 생성한다.

★ 9. more 명령어

" 1페이지가 넘는 문서의 내용을 화면에 출력할 때 페이지 단위로 볼 수 있는 명령어"

예제 :

[orcl:~]\$ more winter.txt

전진키 : 스페이스

후진키 : b

페이지 단위로 내려가는 것 : f

★ 10. head 명령어

문제 24

"문서의 처음 몇줄을 화면에 출력하는 명령어"

예제 :

[orcl:~]\$ [출력줄수] 파일명

[orcl:~]\$ head -20 winter.txt

"Final Shooting Draft

by

Jennifer Lee

9/23/13

Property of the Walt Disney Animation Studios

OPEN ON: ICE.

We' re underwater looking up at it. A saw cuts through,  
heading right for us.

EXT. SNOW-CAPPED MOUNTAINS ? DUSK

ICE HARVESTERS, dressed in traditional Sami clothing, score a  
frozen lake. They SING.

" The Frozen Heart (Ice Worker' s Song)"

ICE HARVESTERS

BORN OF COLD AND WINTER AIR

AND MOUNTAIN RAIN COMBINING,

THIS ICY FORCE BOTH FOUL AND FAIR

HAS A FROZEN HEART WORTH MINING.

The men drag giant ice blocks through channels of water.

ICE HARVESTERS (CONT' D)

★ 11. wc 명령어

문제 25 ~ 28

" 파일안에 단어의 개수 또는 라인수를 출력하는 명령어 "

예 : \$ wc -l 파일명  
\$ wc -l emp.txt

14 emp.txt

옵션 :    -l    : 라인수 출력  
         -w    : 단어의 개수 출력  
         -c    : 문자수를 출력

★ 12. grep 명령어

문제 29 ~ 33

" 파일안에 포함된 특정 단어나 구문을 검색하는 명령어 "

예 :  
\$ grep '찾고 싶은 단어' 파일명  
\$ grep -i 'scott' emp.txt

설명 : emp.txt 에서 scott이라는 단어가 포함된 라인을 보여달라  
-i 옵션 : 대소문자를 구분하지 않겠다.

★ 13. awk 명령어

문제 34 ~ 41

" 특정 컬럼을 출력하고자 할 때 사용하는 명령어 "

예 :  
\$ awk '패턴 {action}' 대상 파일명  
  
\$ awk '{print \$2, \$6}' emp.txt  
  
\$ awk '\$3=="SALESMAN" {print \$2, \$3}' emp.txt

※ 리눅스의 연산자 3가지

1. 산술 연산자 : +, -, \*, /
2. 비교 연산자 : >, <, >=, <=, ==, !=
3. 논리 연산자 : &&, ||, !

★ 14. sort 명령어

문제 42 ~ 54

"data를 특정 컬럼을 기준으로 정렬하는 명령어"

예제 :

```
[orcl:~]$ sort -k 6 emp.txt
```

※ 설명 : emp.txt의 모든 데이터를 출력하는데 6번째 컬럼의 데이터가 작은 값부터 출력하시오 !

★ 15. uniq 명령

문제 55 ~ 62

"중복된 라인을 제거하는 명령어"

예 : \$ uniq 옵션 파일명

★ 16. echo 명령어

문제 63 ~ 72

"출력하고자 하는 글자를 출력할때 사용하는 명령어"

예제 :

```
[orcl:~]$ echo '야호~'
```

★ 17. diff 명령어

문제 73

"두 파일간의 차이점을 찾아서 알려주는 명령어"

예 :

```
[orcl:~]$ diff emp.txt emp7.txt
14a15
> 7839 KING      PRESIDENT      0 81/11/17      5000      0      10
```

★ 18. find 명령어

문제 74 ~ 76

"검색하고자 하는 파일을 찾을 때 사용하는 명령어"

예 : \$ find 디렉토리 -name 파일명 -print

          ↑                  ↑  
          검색할 디렉토리  검색할 파일명

```
[orcl:~]$ find /home/oracle -name 'emp.txt' -print
/home/oracle/emp.txt
```

※ 설명 : /home/oracle 디렉토리 밑에 emp.txt라는 파일이 있는지 검색하시오 !



★ 19. tar 명령어

문제 77 ~ 80

"파일을 압축하고 압축 해제하는 명령어 "

예제 :

1. 압축할 때

```
$ tar cvf 압축파일명 압축파일대상
```

2. 압축을 해제할 때

```
$ tar xvf 압축파일명 압축해제할 위치
```

옵션 :

- c : compress, 여러개의 파일을 하나로 만들어라
- v : view , 압축되는 과정을 보여달라
- f : file , 생성되는 파일명을 지정
- x : extract , 묶여있는 파일을 풀어줘라
- C : 압축이 풀릴 위치를 지정

★ 20. ln(link) 명령어

문제 81

"윈도우의 바로가기 기능과 유사함"

" 내가 자주 열어봐야 하는 파일이 있다면 바로가기를 생성해 놓으면 편하게 열어볼 수 있다."

예제 :

```
$ ln -s "링크를 걸 파일위치와 파일명" "바로가기 이름"
```

```
[orcl:~]$ mkdir -p ./test22/test23/test24/test25/test26/test27
[orcl:~]$ cd ./test22/test23/test24/test25/test26/test27
[orcl:test27]$ pwd
/home/oracle/test22/test23/test24/test25/test26/test27
```

```
[orcl:test27]$ vi salgrade.txt
[orcl:test27]$ cat salgrade.txt
1 700 1200
2 1201 1400
3 1401 1800
4 1801 3000
5 3001 5000
```

★ 21. sed 명령어

문제 82

"grep 명령어는 파일의 특정 내용을 검색하는 기능을 갖는다면 sed 명령어는 검색 뿐만아니라 내용을 변경할 수 있다."

예제 :

```
[orcl:~]$ sed 's/KING/yyy/' emp.txt
7839 yyy      PRESIDENT      0 81/11/17      5000      0      10
7698 BLAKE    MANAGER      7839 81/05/01      2850      0      30
7782 CLARK    MANAGER      7839 81/05/09      2450      0      10
```

★실제로 emp.txt 가 변경된 것은 아니다.

★ 22. cp 명령어

문제 83

"파일을 복사하는 명령어"

예제 : \$ cp 파일명 복사할 파일명

\$ cp 위치/파일명 위치/복사할 파일명

[orcl:~]\$ cp emp.txt emp400.txt

★ 23. mv 명령어

문제 84

"파일의 이름을 바꾸거나 파일을 다른 디렉토리로 이동하는 명령어"

예제 : \$ mv 기존 파일명 새로운 파일명

[orcl:~]\$ mv emp400.txt emp500.txt

[orcl:~]\$ mv emp.txt ./backup/

[orcl:~]\$ ls emp.txt

ls: emp.txt: No such file or directory

### ■ 3. vi 편집기 명령어

- vi 편집기란 ? 리눅스안에서 사용할 수 있는 문서 편집기  
vi(Visual Editor)의 뜻에서 유래가 됨

- vi편집기의 명령모드 3가지

1. command 모드

vi의 기본 모드이며 vi 를 실행하면 바로 보이는 화면  
방향키로 왔다갔다 할 수 있는 화면

2. Edit 모드

a, i, o 등을 누르면서 내용을 입력 또는 삭제하는 명령모드

3. last line 모드

입력 모드에서 저장 종료, 강제종료 등의 명령어를 입력하는 모드  
:wq! 저장하고 종료  
:q! 저장 안하고 종료 (단축키 :zq)

★ 편집기를 시작하는 명령어

1. \$ vi 파일명 : 읽기, 쓰기가 가능한 상태로 파일이 열린다.
2. \$ view 파일명 : 읽기만 가능한 상태로 파일이 열린다.
3. \$ vi -r 파일명 : 손상된 파일을 복구하는 명령어

※ vi 편집기를 열 때 정상적으로 안열리고 이상한 경고 메세지들이 나오면서 열릴 때 해결방법은?

- swap파일을 삭제한다.

예 : \$ rm .emp.txt.swp

★ vi 편집기 내에서 커서 이동

문제 85

\$ vi winter.txt

1. j : 아래로 이동
2. k : 위로이동
3. h : 왼쪽으로 이동
4. l : 오른쪽으로 이동
5. gg: 맨 위로 이동
6. G : 맨 아래로 이동
7. :set nu : 파일 내의 텍스트에 번호표시
8. :set nonu : 번호 안보이게 하는 방법

★ vi 편집기의 입력 명령어

1. i : 커서의 왼쪽부터 입력하겠다.
2. a : 커서의 오른쪽부터 입력하겠다.
3. o : 커서의 다음라인부터 입력하겠다.
4. O : 커서의 윗라인부터 입력하겠다.

★ vi 편집기의 삭제 명령어

문제 86

1. x : 철자 하나 삭제
2. dd : 한행 삭제
3. dw : 커서에 있는 단어 삭제
4. :5,10 d : 5 ~ 10번째 행 삭제
5. D : 커서 오른쪽 행 삭제

★ 취소 명령어

u

★ 복사/붙여넣기 명령어

문제 87 ~ 90

1. yy : 하나의 행을 복사
2. p : 붙여넣기
3. yG : 현재행 부터 파일 끝까지 복사
4. :1,2 co 3 : 1~2행을 3행 다음으로 복사
5. :1,2 m 3 : 1~2행을 3행 다음으로 이동

★ vi 편집기 내에서 문자를 변경하는 명령어

문제 91 ~ 93

1. :/문자 -----> 문자 검색

예 : \$ vi winter.txt  
:/ELSA  
알파벳 n을 누르면서 계속 검색  
shift + n을 누르면서 다시 뒤로 검색

2. :s/기존 문자/변경할 문자  
지금 커서가 있는 현재 행의 기존문자를 변경할 문자로 변경하겠다.
3. :%s/기존문자/변경할 문자/g # g는 global  
문서 전체의 기존문자를 변경할 문자로 변경하겠다.

★ 리눅스 쉘 스크립트를 디버깅 모드로 실행하는 방법

문제 94 ~ 96

\$ sh -x cp\_emp.sh

## ■ 4. 권한 관리

### 문제 97 ~ 104

리눅스에서 파이썬의 배치파일을 실행하거나 R배치 파일을 실행하거나 하둡을 설치하고 운영을 할 때 여러가지 문제들이 발생하는데 그 중에 많은 문제들이 권한에 관련한 오류들이 많다. 그래서 권한관리를 잘 알아둬야 한다.

\* 권한관리에 관한 명령어 3가지

1. chmod : change mode
2. chown : change file attributes
3. chattr : change ownership of a file

\* 권한관리 표

번호	권한	대표문자	파일	디렉토리
1	읽기권한	r	읽고, copy	디렉토리에서 ls가능
2	쓰기권한	w	수정	디렉토리에 파일 생성 가능
3	실행권한	x	실행	디렉토리에서 cd로 접근 가능

\* ls -l 로 특정 파일을 조회했을 때 나오는 권한부분을 해석

```
$ ls -l emp.txt
-rw-r--r-- 1 oracle oinstall 1366 Dec 19 10:47 emp.txt
  ↑      ↑      ↑
  권한   파일소유자 그룹
```

\* 권한 부분을 좀 더 상세하게 분석하시오 !

```
-          rw-          r--          r--
↑          ↑          ↑          ↑
디렉토리인지  파일의 소유자의  그룹의 권한  기타유저
파일인지를 나타냄  권한
```

★ 권한을 설정하는 숫자

### 문제 105 ~ 106

- 0 ---> 퍼미션을 허용하지 않겠다.
- 1 ---> 실행(x) 퍼미션을 허용하겠다.
- 2 ---> 쓰기(w) 퍼미션을 허용하겠다.
- 4 ---> 읽기(r) 퍼미션을 허용하겠다.

예 :

```
$ chmod u+rw,g+rw,o+r emp.txt
```

↓ 위의 명령어를 숫자로 실행하면 아래와 같이 간단해진다

```
$ chmod 764 emp.txt
```

★ chown 명령어

[문제 107 ~ 110](#)

"파일이나 디렉토리의 소유자를 변경하는 명령어"

```
[orcl:~]$ su -
```

```
Password:
```

```
[root@edydr1p0 ~]# cd /home/oracle
```

```
[root@edydr1p0 oracle]# ls -l emp.txt
```

```
-r--r--r-- 1 oracle oinstall 1370 Dec 19 11:46 emp.txt
```

```
[root@edydr1p0 oracle]# whoami
```

```
root
```

```
[root@edydr1p0 oracle]# chown root:root emp.txt
```

```
          ↑   ↑  
        유저명 그룹명
```

```
[root@edydr1p0 oracle]# ls -l emp.txt
```

```
-r--r--r-- 1 root root 1370 Dec 19 11:46 emp.txt
```

## ■ 5. 디스크 관리

\* 디스크 관리 명령어 3가지

1. df 명령어
2. du 명령어
3. sar 명령어

★ du 명령어

[문제 111](#)

"현재 파일/디렉토리의 디스크 사용량을 표시하는 명령어"

예제 :

```
[orcl:~]$ du -sh
55M      .
[orcl:~]$ du -s
55896    .
```

```
[orcl:~]$ du -s *.txt
4        dept.txt
4        emp.txt
4        emp1.txt
4        emp10.txt
4        emp11.txt
:
:
```

★ df 명령어

" 현재 파일 시스템의 총 사용율을 확인하는 명령어"

예 :

```
[orcl:~]$ df -hP
Filesystem              Size  Used Avail Use% Mounted on
/dev/mapper/VolGroup00-LogVol00  190G  44G  137G  25% /
/dev/sda1                99M   18M   76M  20% /boot
tmpfs                   1014M  442M  572M  44% /dev/shm
```

★ sar 명령어

[문제 112 ~ 117](#)

"disk의 i/o 성능을 모니터링하는 명령어 "

예제 :

```
[orcl:~]$ sar 1 100
Linux 2.6.18-164.el5 (edydr1p0.us.oracle.com) 12/19/18

15:41:26      CPU      %user      %nice      %system      %iowait      %steal      %idle
15:41:27      all        0.00        0.00        0.00        0.00        0.00      100.00
15:41:28      all        0.50        0.00        0.50        0.00        0.00      99.00
15:41:29      all        0.00        0.00        0.50        0.00        0.00      99.50
```

⋮

오라클에서 악성 SQL을 수행하거나 시스템에서 무한루프를 돌리면 CPU 사용율과 %user의 사용율이 올라간다.

#### \* 컬럼 소개

%user -> oracle 유저와 같은 일반 유저가 사용하는 disk i/o

%nice -> cpu를 양보하는 친절도

%system-> system을 사용하는 disk i/o

%iowait-> i/o를 일으키면서 얼마나 대기하는지

%idle -> 작업을 안하고 있는 idle한 상태

%steal -> ?



## ■ 6. 프로세서 관리

\* 프로세서 관리 명령어

1. ps 명령어
2. top 명령어
3. kill 명령어
4. job 명령어

★ jobs 명령어

[문제 118 ~ 120](#)

"동작중인 작업의 상태를 확인하는 명령어"

예 : \$ jobs 옵션

- 상태 정보 4가지
  1. running : 실행중
  2. stopped : 일시중단중
  3. Done : 종료
  4. terminated : 강제종료

\$ vi hhh.txt

```
select ename, sal, job, deptno
  from emp
 where
```

ctrl + z <----- 하던 작업 취소

[1]	+	Stopped	vim hhh.txt
↑	↑	↑	↑
job번호	현재job	일시정지	동작중인 job

[orcl:~]\$ jobs

[1]+ Stopped vim hhh.txt

※ 현재 진행중인 job을 확인할 수 있다.

[orcl:~]\$ fg <----- 현재job으로 접속하는 명령어(for ground)

[orcl:~]\$ sar 1 1000 >> sar\_20181220.txt &

[2] 5580

[orcl:~]\$ jobs <----- 동작중인 작업의 상태를 확인하는 명령어

[1]+ Stopped vim hhh.txt

[2]- Running sar 1 1000 >> sar\_20181220.txt &

★ top명령어

[문제 121](#)

" 지금 현재 작동중인 프로세서들의 cpu 사용율과 메모리 사용율을 확인하는 명령어 "

★ ps 명령어

[문제 122 ~ 125](#)

" 현재 시스템에서 수행되고 있는 프로세서의 정보를 표시하는 명령어 "

예제 :

\$ ps 옵션 프로세서 아이디

옵션 :

- e : 현재 실행중인 모든 프로세서
- f : 실제 유저명, 개시시간등을 표시
- l : 프로세서의 상태, 우선도 등과 같은 상세한 정보
- p : 프로세서 아이디

★ kill 명령어

[문제 126](#)

\$ kill -9 프로세서 번호

옵션 : -9 : 프로세서를 죽인다.

-6 : 프로세서를 중단한다.

-2 : 프로세서에게 종료하도록 지시한다.

## ■ 7. 셸 스크립트 작성법

### ▷ 셸(shell)이란 무엇인가?

shell이란 운영체제에서 제공하는 명령을 실행하는 프로그램이다.

### ▷ 셸(shell) 스크립트 란?

인터프리터(통역사) 역할을 하는것으로 시스템에서 지원하는 명령어들의 집합을 묶어서 프로그램화한것을 말한다.

### ▷ 셸(shell)의 종류

1. Bourne shell
2. C shell
3. Korn shell
4. bash shell

예 : `#!/bin/bash` ---> 셸 중에 bash 셸을 쓰겠다.

### ▷ 셸 스크립트 프로그래밍이란?

1. C언어와 유사한 프로그래밍
2. 변수, 반복문(loop문), 제어문(if문)이 사용가능
3. 별도로 컴파일 하지 않고 텍스트 파일 형태로 바로 실행이 가능하다.
4. vi 나 gedit으로 작성이 가능
5. 리눅스의 많은 부분이 쉘스크립트로 작성이 되어있다.

### ★ 셸 스크립트를 작성하고 실행하는 방법

#### 문제 127

```
$ vi a.sh
```

```
echo "host name : " $HOSTNAME
```

```
$ sh a.sh
```

```
host name : edydr1p0.us.oracle.com
```

```
[orcl:~]$ ./a.sh
```

```
-bash: ./a.sh: Permission denied
```

### ★ 변수 사용법

1. 모든 변수는 '문자열(string)' 으로 취급된다.
2. 변수 이름은 대소문자를 구분한다.
3. 변수에 값을 대입할 때는 '=' 좌우에 공백이 없어야한다.
4. 변수에 들어간 문자를 출력하려면 변수 앞에 \$ 를 붙이고 echo명령어로 출력하면 된다.

예 : `$ myvar="Hi !!!!!!!"`

```
$ echo $myvar
```

★ 변수의 숫자 계산하는 방법

문제 128 ~ 129

1. 변수에 대입한 값은 모두 문자열로 취급이 된다.
2. 변수에 들어있는 값을 숫자로 해서 + \* / - 등의 연산을 하려면 expr을 사용해야 한다.
3. 수식에 괄호 또는 곱하기(\*)를 사용하려면 그 앞에 반드시 (w)를 붙여야 한다.

예 :

```
[orcl:~]$ num1=100
[orcl:~]$ num2=200
[orcl:~]$ echo $num1
100
[orcl:~]$ echo $num2
200
[orcl:~]$ echo $num2 + $num2
200 + 200
[orcl:~]$ expr $num1 + $num2
300
```

★ 파라미터 변수

문제 130 ~ 131

1. 파라미터 변수는 \$0, \$1, \$2, ...의 형태를 가진다.
2. 전체 파라미터는 \$\*로 표현한다.

예제 :

```
[orcl:~]$ vi b.sh
echo "실행파일 이름은 $0 입니다."
echo "첫번째 파라미터 값은 $1 입니다."
echo "두번째 파라미터 값은 $2 입니다."
echo "전체 파라미터 값은 $* 입니다."
```

```
[orcl:~]$ sh b.sh 10 20
실행파일 이름은 b.sh 입니다.
첫번째 파라미터 값은 10 입니다.
두번째 파라미터 값은 20 입니다.
전체 파라미터 값은 10 20 입니다.
```

★ 리눅스 셸에서 if문 사용방법

```
if [ 조건 ]; then
    실행문
elif [ 조건 ]; then
    실행문
else
    실행문
fi
```

※ if 문 사용시 주의사항 ! "공백처리를 잘해줘야 한다."

★ if 문 조건문에 들어가는 비교 연산자

문제 132 ~ 133

- 문자열 비교

1. "문자열1" = "문자열2" : 두 문자열이 같으면 true
2. "문자열1" != "문자열2" : 두 문자열이 같지 않으면 true

- 숫자열 비교

1. 숫자1 -eq 숫자2 : 두 숫자가 같으면 True equal
2. 숫자1 -ne 숫자2 : 두 숫자가 같지 않으면 True not equal
3. 숫자1 -gt 숫자2 : 숫자1이 숫자2보다 크다면 True greater
4. 숫자1 -ge 숫자2 : 숫자1이 숫자2보다 크거나 같다면 True greater or equal
5. 숫자1 -lt 숫자2 : 숫자1이 숫자2보다 작다면 True little
6. 숫자1 -le 숫자2 : 숫자1이 숫자2보다 작거나 같으면 True le - little or equal
7. !숫자1 : 숫자1이 거짓이라면 True

예제 :

```
[orcl:~]$ vi if1.sh
#!/bin/bash

if [ 100 -eq 200 ]; then
    echo "100 is equal to 200"
else
    echo "100 is not equal to 200"
fi

[orcl:~]$ sh if1.sh
100 is not equal to 200
```

★ 리눅스에서 사용하는 연산자

문제 134 ~ 135

※ 리눅스의 연산자 3가지

1. 산술 연산자 : +, -, \*, /
2. 비교 연산자 : >, <, >=, <=, ==, !=
3. 논리 연산자 : &&, ||, !

논리 연산자인 and 는 && 또는 -a를 사용한다.  
논리 연산자인 or 는 || 또는 -o를 사용한다.

예제 :

```
if [ $sal -lt 2000 ] && [ $job=="SALESMAN" ]; then

    또는

if [ $sal -lt 2000 -a $job=="SALESMAN" ]; then
```

★ loop 문

문제 136 ~ 145

\* for loop문 문법  
for 변수 in 값1,값2,값3  
do  
    반복할 문장  
done

예제 :

```
[orcl:~]$ vi for.sh
#!/bin/bash

hap=0
for i in {1..10}
do
    hap=`expr $hap + $i`
done
echo "1부터 10까지의 합은 " $hap

[orcl:~]$ sh for.sh
1부터 10까지의 합은 55
```

-우분투버전

```
#!/bin/bash

hap=0
for i in $(seq 1 10)
do
    hap=$(( $hap + $i ))
done
echo "1부터 10까지의 합은 " $hap
```

★ sed 명령어로 공백라인을 지우는 방법 (데이터 정제 작업)

문제 146 ~ 156

1. emp.txt 를 열어서 중간에 공백라인을 하나 넣으시오 !
2. emp.txt 안에 공백라인이 있는지 검색한다.┆  
[orcl:~]\$ sed -n /^\$/p emp.txt | wc -l
3. emp.txt 안에 공백라인을 삭제한다.

```
[orcl:~]$ sed '/^$/d' emp.txt >> emp_backup.txt
[orcl:~]$ rm emp.txt
[orcl:~]$ mv emp_backup.txt emp.txt
[orcl:~]$ sed -n /^$/p emp.txt | wc -l
0
```

※ 설명 : -n 옵션은 sed 옆의 식에 해당하는 데이터만 출력해라 라는 뜻이다.

※ 스페이스바 넣은 공백까지 다 지우는 방법

1. emp.txt를 열어서 중간에 스페이스바로 공백을 넣는데 여러번 엔터를 쳐서 넣으시오 !
2. emp.txt 에서 여러번 엔터를 넣은 스페이스바 공백이 있는지 확인한다.  
[orcl:~]\$ cat emp.txt
3. emp.txt에서 여러번의 엔터를 넣은 스페이스바 공백을 지운 상태로 출력하시오 !

```
[orcl:~]$ sed '/^ */d' emp.txt
```

```
4. [orcl:~]$ sed '/^ */d' emp.txt >> emp_backup.txt
```

```
5. [orcl:~]$ rm emp.txt
```

```
[orcl:~]$ mv emp_backup.txt emp.txt
```

### ★ while loop문

[문제 157 ~ 163](#)

문법 :       조건에 만족할 때만 루프문을 반복 수행한다.



while [ 조건 ]   ← 한칸씩 띄어쓰기를 해야한다.

do

    실행문

done

예제 :

```
[orcl:~]$ vi test.sh
```

```
i=1
```

```
while [ $i -le 10 ]
```

```
do
```

```
    echo $i
```

```
    i=`expr $i + 1`
```

```
done
```

```
[orcl:~]$ sh test.sh
```

```
1
```

```
2
```

```
3
```

```
4
```

```
5
```

```
6
```

```
7
```

```
8
```

```
9
```

```
10
```

## ■ 파일과 관련된 조건

[문제 164 ~ 169](#)

1. -d 파일명 : 파일이 디렉토리이면 true
2. -e 파일명 : 파일이 존재하면 true
3. -f 파일명 : 파일이 일반 파일이면 true
4. -r 파일명 : 파일이 읽기 가능하면 true
5. -s 파일명 : 파일이 크기가 0이 아니면 true
6. -w 파일명 : 파일이 쓰기가 가능한 상태면 true
7. -x 파일명 : 파일이 실행가능한 상태면 true

예제 :

```
[orcl:~]$ vi test.sh
#!/bin/bash
```

```
fname=/home/oracle/emp.txt
if [ -e $fname ]; then
    cat $fname
else
    echo "파일이 없습니다."
fi
```

```
[orcl:~]$ sh test.sh
```

7839 KING	PRESIDENT	0 81/11/17	5000	0	10
7698 BLAKE	MANAGER	7839 81/05/01	2850	0	30



## ■ 데이터 분석을 위해 만들어야할 리눅스 셸 스크립트

1. 데이터를 분리하는 스크립트
2. 오라클 --> R 또는 파이썬 에서 분석할 수 있도록 text파일로 생성하는 작업
3. 자동화 스크립트

## ■ 오라클의 data를 csv 로 생성하는 방법

### 문제 170

1. 오라클에 접속을 한다

```
[orcl:~]$ sqlplus / as sysdba
```

```
SQL> show user
```

```
USER is "SYS"
```

2. scott 계정과 hr계정의 lock을 해제하고 패스워드를 tiger로 변경하시오 !

```
SQL> alter user scott
```

```
2 account unlock;
```

```
User altered.
```

```
SQL> alter user scott
```

```
2 identified by tiger;
```

```
User altered.
```

```
SQL> connect scott/tiger
```

```
Connected.
```

```
SQL> select * from emp;
```

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7369	SMITH	CLERK	7902	17-DEC-80	800		20
7499	ALLEN	SALESMAN	7698	20-FEB-81	1600	300	30
7521	WARD	SALESMAN	7698	22-FEB-81	1250	500	30
			:				
			:				
7902	FORD	ANALYST	7566	03-DEC-81	3000		20
7934	MILLER	CLERK	7782	23-JAN-82	1300		10

### ★ 리눅스 sqlplus 에서 ed 썼을 때 vi 편집기가 열리게 하는 방법

```
[orcl:~]$ cd $ORACLE_HOME/sqlplus/admin
```

```
[orcl:admin]$ pwd
```

```
/u01/app/oracle/product/11.2.0/dbhome_1/sqlplus/admin
```

```
[orcl:admin]$ vi glogin.sql          #sqlplus의 환경설정 파일
```

```
--
```

```
-- Copyright (c) 1988, 2005, Oracle. All Rights Reserved.
```

```
--
```

```
-- NAME
```

```
-- glogin.sql
```

```
--
```

```
-- DESCRIPTION
```

```
-- SQL*Plus global login "site profile" file
```

```
--
```

```
-- Add any SQL*Plus commands here that are to be executed when a
```

```
-- user starts SQL*Plus, or uses the SQL*Plus CONNECT command.
```

```
--
```

```
-- USAGE
```

```
-- This script is automatically run
--
define _editor='vi'

[orcl:admin]$ cd
[orcl:~]$ sqlplus scott/tiger

SQL> show user
USER is "SCOTT"

SQL> select job, sum(sal)
2   from emp
3   group by job
4   order by sum(sal) des;
order by sum(sal) des
                *
```

ERROR at line 4:  
ORA-00933: SQL command not properly ended

```
SQL> ed
Wrote file afiedt.buf

1 select job, sum(sal)
2   from emp
3   group by job
4*  order by sum(sal) desc
```

```
SQL> /
```

JOB	SUM(SAL)
MANAGER	8275
ANALYST	6000
SALESMAN	5600
PRESIDENT	5000
CLERK	4150

★ 리눅스의 emp테이블의 데이터를 csv로 생성하는 방법  
[문제 171 ~ 177](#)

```
SQL> spool on
SQL> spool result.txt
SQL> select * from dept;
```

DEPTNO	DNAME	LOC
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON

```
SQL> spool off
```

```
SQL> ed result.txt
```

```
SQL> select * from dept;
```

DEPTNO	DNAME	LOC
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON

SQL> spool off

※ sqlplus에서 spool 명령어를 이용하면 데이터를 txt나 csv파일로 생성할 수 있다.

```
[orcl:~]$ vi create_emp_data.sql
set colsep ','          # 데이터를 , 로 구분하겠다.
set pagesize 20          # 결과의 세로 사이즈 조절
set linesize 120         # 결과의 가로 사이즈 조절
set sqlprompt ""         # SQL> <----- 안나오게 하는 명령어
set feedback off        # 몇 건이 출력되었습니다. 안나오게 하는 명령어
set echo off
set underline off       # 컬럼명 밑에 언더라인 사라지게
```

spool emp.csv

select \* from emp;

spool off

exit

[orcl:~]\$ sqlplus scott/tiger

SQL> @create\_emp\_data.sql

[orcl:~]\$ cat emp.csv

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7369	SMITH	CLERK		7902-17-DEC-80	800		20
7499	ALLEN	SALESMAN		7698-20-FEB-81	1600	300	30
7521	WARD	SALESMAN		7698-22-FEB-81	1250	500	30
7566	JONES	MANAGER		7839-02-APR-81	2975		20
7654	MARTIN	SALESMAN		7698-28-SEP-81	1250	1400	30
7698	BLAKE	MANAGER		7839-01-MAY-81	2850		30
7782	CLARK	MANAGER		7839-09-JUN-81	2450		10
7788	SCOTT	ANALYST		7566-19-APR-87	3000		20
7839	KING	PRESIDENT		17-NOV-81	5000		10
7844	TURNER	SALESMAN		7698-08-SEP-81	1500	0	30
7876	ADAMS	CLERK		7788-23-MAY-87	1100		20
7900	JAMES	CLERK		7698-03-DEC-81	950		30
7902	FORD	ANALYST		7566-03-DEC-81	3000		20
7934	MILLER	CLERK		7782-23-JAN-82	1300		10

## ■ case문

### 문제 178 ~ 186

"if 문과 유사한 문법인데 특정 셸 스크립트를 실행할 수 있도록 도와주는 문법"

예제 : \* 준비해야할 스크립트 2가지

1. 두 파일의 차이를 확인하는 diff.sh
2. 파일을 검색하는 find\_file.sh

## ■ 유저 생성 및 관리

### \* 유저 생성 및 관리 명령어

1. useradd : 유저 생성
2. usermod : 유저 수정
3. userdel : 유저 삭제
4. passwd : 패스워드 변경
5. groupadd : 그룹생성
6. groupmod : 그룹수정
7. groupdel : 그룹 삭제
8. su : 유저 변경

### \* root유저로 접속한다.

```
[orcl:~]$ su -  
Password: oracle  
[root@edydr1p0 ~]# whoami  
root  
[root@edydr1p0 ~]# id  
uid=0(root) gid=0(root) groups=0(root),1(bin),2(daemon),3(sys),4(adm),6(disk),10(wheel)
```

### ★ useradd명령어

[문제 187](#)

"유저를 생성하는 명령어"

예제 : # useradd 유저명

### ★ passwd 명령어

[문제 188 ~ 190](#)

"사용자의 비밀번호를 지정하거나 변경하는 명령어"

사용문법 : # passwd 유저명

```
[root@edydr1p0 ~]# passwd oracle2  
Changing password for user oracle2.  
New UNIX password: oracle  
BAD PASSWORD: it is based on a dictionary word  
Retype new UNIX password: oracle  
passwd: all authentication tokens updated successfully.
```

```
[orcl:~]$ su - oracle2  
Password: oracle
```

### ★ usermod 명령어

[문제 191 ~ 194](#)

"유저 계정의 정보를 변경하는 명령어"

\* 사용 문법 : # usermod [옵션] 유저명

\* 옵션 : -d : 사용자의 홈 디렉토리를 지정하는 옵션

★ userdel 명령어

문제 195 ~ 196

"유저 삭제하는 명령어"

예제 : # userdel oracle2

★ groupadd 명령어

"그룹을 생성하는 명령어"

\* 사용문법 : # groupadd -g 그룹번호 그룹명

예제 :

```
[root@edydr1p0 home]# groupadd -g 516 project1
[root@edydr1p0 home]# cat /etc/group
:
:
project1:x:516:
```

★ groupmod 명령어

"그룹을 변경하는 명령어 (group 이름, 번호를 변경)"

\* 사용문법 :

```
[root@edydr1p0 home]# groupmod -n project2 project1
[root@edydr1p0 home]# cat /etc/group
:
:
project2:x:516:
```

★ gpasswd 명령어

문제 197

"특정 그룹에 유저를 추가/제거하는 명령어"

\* 사용문법 : # gpasswd [옵션] 유저명 그룹명

\* 옵션 : -a : 특정 그룹에 새로운 멤버 추가  
-d : 특정 그룹에 기존 멤버 삭제

\* 예제 :

```
[root@edydr1p0 home]# useradd oracle9
[root@edydr1p0 home]# passwd oracle9
Changing password for user oracle9.
New UNIX password: oralc
BAD PASSWORD: it is based on a dictionary word
Retype new UNIX password: oracle
passwd: all authentication tokens updated successfully.

[root@edydr1p0 ~]# gpasswd -a oracle9 oinstall
                        ↑      ↑
                        유저명  그룹명
Adding user oracle9 to group oinstall
[root@edydr1p0 ~]# cat /etc/group
:
:
oinstall:x:504:oracle9
```

:  
:  
oracle9:x:517:



## ■ 문제 모음

문제 1. 다시 /home/test123으로 이동하시오 !

```
test123@woody-dev:~/labs$ cd ..
test123@woody-dev:~$ pwd
/home/test123
```

문제 2. 현재 디렉토리에서 / (루트) 디렉토리로 바로 이동하시오 !

```
test123@woody-dev:~$ cd /
test123@woody-dev:/$ pwd
/
```

문제 3. /home 밑으로 이동하시오 !

```
test123@woody-dev:/$ cd /home          # 그냥 cd home해도 됨
test123@woody-dev:/home$ pwd
/home

test123@woody-dev:/home$ ls
test123  woody
```

문제 4. /home/test123으로 이동하시오 !

```
test123@woody-dev:/home$ cd test123
test123@woody-dev:~$ pwd
/home/test123
```

※ 경로에는 크게 2가지 경로가 있다.

1. 절대경로 : cd "내가 가고자하는 정확한 위치"

예 : \$ pwd

\$ cd /home

2. 상대경로 : 나의 현재위치를 상대로 이동하겠다.

예 : cd .. 나의 현재위치를 상대로 상위 디렉토리로 이동하겠다.

cd <--- 자기 자신의 홈 디렉토리로 이동  
(/home/test123)

문제 5. 현재 디렉토리에서 test 라는 디렉토리를 만들고 test 디렉토리로 이동하시오 !

```
test123@woody-dev:~$ mkdir test3
test123@woody-dev:~$ pwd
/home/test123
test123@woody-dev:~$ cd test3
test123@woody-dev:~/test3$ pwd
/home/test123/test3
test123@woody-dev:~/test3$
```

문제 6. 지금 현재 디렉토리에서 상위 디렉토리의 상위디렉토리로 이동하시오 !

```
/home/test123/test3 ----> /home

test123@woody-dev:~/test3$ cd ../../
test123@woody-dev:/home$
test123@woody-dev:/home$ pwd
/home
```

문제 7. 그럼 다시 원래 있었던 위치로 이동하시오 !

```
test123@woody-dev:/home$ cd -
/home/test123/test3
test123@woody-dev:~/test3$ pwd
/home/test123/test3
```

문제 8. 자기 폴더(/home/test123/test3) 로 이동해서 아래의 파일들을 크기가 0으로 생성하시오

```
a.txt b.txt c.txt d.txt e.txt f.txt

test123@woody-dev:~/test3$ touch a.txt b.txt c.txt d.txt e.txt f.txt
test123@woody-dev:~/test3$ ls -l *.txta
-rw-rw-r-- 1 heaven heaven 0 Dec 14 01:45 a.txt
-rw-rw-r-- 1 heaven heaven 0 Dec 14 01:45 b.txt
-rw-rw-r-- 1 heaven heaven 0 Dec 14 01:45 c.txt
-rw-rw-r-- 1 heaven heaven 0 Dec 14 01:45 d.txt
-rw-rw-r-- 1 heaven heaven 0 Dec 14 01:45 e.txt
-rw-rw-r-- 1 heaven heaven 0 Dec 14 01:45 f.txt
```

문제 9. 리눅스 서버에 자기 홈 디렉토리를 생성하시오 !

```
heaven@buzz:~/test$ cd
heaven@buzz:~$ pwd
/home/heaven
```

문제 10. emp.txt를 리눅스 서버에 자기 홈 디렉토리에 올리시오 !

1. 도스창을 연다.
2. c:\User\Administrator> <---- 이 폴더에 emp.txt, dept.txt 를 가져다 둔다
3. ftp 192.168.56.104  
★★다시

문제 11. /home/ test014/oracle\_test 밑에 아래의 하위 디렉토리를 생성하시오 !

```
heaven@buzz:~/oracle_test$ mkdir test
heaven@buzz:~/oracle_test$ cd test
heaven@buzz:~/oracle_test/test$ mkdir test2
heaven@buzz:~/oracle_test/test$ cd test2
```

```
heaven@buzz:~/oracle_test/test/test2$ mkdir test3
heaven@buzz:~/oracle_test/test/test2$ cd test3
heaven@buzz:~/oracle_test/test/test2/test3$ pwd
/home/heaven/oracle_test/test/test2/test3
```

-다른방법

```
heaven@buzz:~/oracle_test$ mkdir -p test1/test2/test3/test4/test5
```

\$ man mkdir <---- mkdir 명령어의 메뉴얼 (q누르면 빠져나올 수 있음)

문제 12. 자기 홈디렉토리에 1.txt ~ 10.txt 를 생성하시오 !

```
heaven@buzz:~$ touch 1.txt
heaven@buzz:~$ touch 2.txt
heaven@buzz:~$ touch 3.txt
heaven@buzz:~$ touch 4.txt
heaven@buzz:~$ touch 5.txt
heaven@buzz:~$ touch 6.txt
heaven@buzz:~$ touch 7.txt
heaven@buzz:~$ ls
1.txt 2.txt 3.txt 4.txt 5.txt 6.txt 7.txt oracle_test test
```

문제 13. 확장자가 .txt인 파일들을 전부 삭제하시오 !

```
heaven@buzz:~$ rm *.txt
heaven@buzz:~$ ls
oracle_test test
```

문제 14. /home/oralce\_test밑에 있는 아래의 디렉토리 들을 다 삭제하시오

test1/test2/test3/test4/test5

```
heaven@buzz:~/oracle_test$ rm -rf *
```

※ 설명 : 현재 디렉토리 밑에 있는 모든 파일과 디렉토리를 다 삭제하겠다.

-r 옵션 : 현재 디렉토리 밑에 있는 모든 파일과 디렉토리를 다 삭제하겠다.

-f 옵션 : 삭제할때 원래는 삭제할까요? 라고 물어보는데 물어보지 않고 그냥 강제로 다 삭제해라 !

문제 15. 자기 홈 디렉토리 밑에 test9 라는 디렉토리를 만들고 test9 디렉토리밑에 test10 디렉토리를 만드시오 !

```
heaven@buzz:~$ mkdir test9
heaven@buzz:~$ ls
oracle_test test test9
heaven@buzz:~$ cd test9
heaven@buzz:~/test9$ mkdir test10
heaven@buzz:~/test9$ ls
test10
```

- 다른방법

```
heaven@buzz:~$ mkdir -p ./test9/test10
```

↑  
현재 디렉토리

문제 16. 자기 홈 디렉토리 밑에 있는 test9 와 test10 디렉토리를 전부 삭제하시오 !

```
heaven@buzz:~$ rm -r test9
heaven@buzz:~$ ls
oracle_test test
```

문제 17. 지금 설정되어있는 alias가 무엇이 있는지 조회하시오 !

```
heaven@buzz:~$ alias
```

문제 18. p alias를 지우시오 !

```
heaven@buzz:~$ unalias p
heaven@buzz:~$ p
p: command not found
```

문제 20. 현재 디렉토리에 확장자가 .txt 인 파일들이 무엇이 있는지 조회하시오 !

```
[orcl:~]$ ls -l *.txt
```

문제 21. ls -l \*.txt로 나온 결과를 all.txt라는 이름으로 저장하시오 !

```
[orcl:~]$ ls -l *.txt >> all.txt
[orcl:~]$ cat all.txt
-rw-r--r-- 1 oracle oinstall 1386 Dec 14 15:53 emp.txt
-rw-r--r-- 1 oracle oinstall 1386 Dec 14 16:19 emp50.txt
-rw-r--r-- 1 oracle oinstall 111181 Dec 14 15:43 winter.txt
```

문제 22. cat emp.txt로 나온 결과를 all.txt파일 뒤에 덧붙이시오 !

```
[orcl:~]$ cat emp.txt >> all.txt
[orcl:~]$ cat all.txt
-rw-r--r-- 1 oracle oinstall 1386 Dec 14 15:53 emp.txt
-rw-r--r-- 1 oracle oinstall 1386 Dec 14 16:19 emp50.txt
-rw-r--r-- 1 oracle oinstall 111181 Dec 14 15:43 winter.txt
7839 KING                PRESIDENT                81/11/17                5000                10
7698 BLAKE                MANAGER                  7839 81/05/01            2850                30
7782 CLARK                MANAGER                  7839 81/05/09            2450                10
7566 JONES                MANAGER                  7839 81/04/01            2975                20
7654 MARTIN              SALESMAN                  7698 81/09/10            1250                1400                30
7499 ALLEN                SALESMAN                  7698 81/02/11            1600                300                30
7844 TURNER              SALESMAN                  7698 81/08/21            1500                 0                30
7900 JAMES                CLERK                    7698 81/12/11             950                30
7521 WARD                SALESMAN                  7698 81/02/23            1250                500                30
7902 FORD                ANALYST                  7566 81/12/11            3000                20
7369 SMITH                CLERK                    7902 80/12/09             800                20
```

7788 SCOTT	ANALYST	7566 82/12/22	3000	20
7876 ADAMS	CLERK	7788 83/01/15	1100	20
7934 MILLER	CLERK	7782 82/01/11	1300	10

문제 23. ls -l \* 로 나온 결과를 all.txt 파일에 덮어써버리시오 !

```
[orcl:~]$ ls -l * > all.txt
[orcl:~]$ cat all.txt
```

문제 24. 겨울왕국 대본의 100줄까지의 내용을 winter\_head.txt로 저장하시오 !

```
[orcl:~]$ head -100 winter.txt >> winter_head.txt
```

문제 25. 겨울왕국 대본의 단어수가 어떻게 되는지 출력하시오

```
답 :
$wc -w winter.txt
$wc winter.txt

4240   19909   111181  winter.txt
라인수   단어수   바이트크기
```

문제 26. 자신의 홈 디렉토리 밑에 확장자가 .txt로 끝나는 파일이 몇 개가 있는지 카운트 하시오

```
$ ls -l *.txt
```

```
$ ls -l *.txt | wc -l
```

※ pipe 설명 : 앞의 명령어 | 뒤의 명령어

앞의 명령어의 표준출력을 뒤의 명령어의 표준입력으로 보냄으로써 명령어의 실행결과를 다음 명령어로 전달하는 기능

문제 27. 루트(/) 디렉토리 밑에 tmp 디렉토리 밑에 파일이 몇 개가 있는지 조회하시오

```
답 : $ cd /tmp
      $ ls -l | wc -l
```

문제 28. 루트(/) 디렉토리 밑에 tmp 디렉토리 밑에 있는 모든 파일들을 삭제하시오

답:

```
$ rm -rf *
```

vi emp.txt

엔터 붙여넣기 esc 쉬프트 z z

문제 29. 직업이 salesman인 사원들의 모든 행을 출력하시오

```
[orcl:~]$ grep -i 'salesman' emp.txt
```

문제 30. 직업이 salesman인 사원들이 전부 몇 명인지 출력하시오

```
$ grep -i 'salesman' emp.txt | wc -l
4
```

문제 31. 직업이 salesman인 사원들의 이름과 월급을 출력하시오

```
$ grep -i 'salesman' emp.txt | awk '{print $2,$6}'
```

↑

컬럼선택하는 리눅스 명령어

```
MARTIN 1250
ALLEN 1600
TURNER 1500
WARD 1250
```

문제 32. 부서번호가 10번인 사원들의 이름과 월급과 부서번호를 출력하시오 !

```
[orcl:~]$ awk '{print $2,$6,$8}' emp.txt | grep -iw '10'
CLARK 2450 10
MILLER 1300 10
```

문제 33. 아래와 같은 상황에서 부서번호 10번인 사원들만 출력하려면 어떻게 해야 하는가 ?

```
[orcl:~]$ awk '$8=="10" {print $2,$5,$6,$8}' emp.txt | grep -iw '10'
KING 81/11/17 5000 10
CLARK 81/05/09 2450 10
MILLER 82/01/11 1300 10
```

문제 34. (점심시간 문제) 월급이 3000이상인 사원들의 이름과 월급을 출력하시오 !

```
[orcl:~]$ awk '$6 >= 3000 {print $2,$6}' emp.txt
KING 5000
FORD 3000
SCOTT 3000
```

문제 35. 직업이 salesman이 아닌 사원들의 이름과 직업을 출력하시오 !

```
[orcl:~]$ awk '$3 != toupper("salesman") {print $2, $3}' emp.txt
```

문제 36. 직업이 salesman인 사원들의 이름과 월급을 출력하는데 월급이 1500이상인 사원들만 출력하시오 !

```
[orcl:~]$ awk '$3==toupper("salesman") && $6>=1500 {print $2,$6}' emp.txt
```

문제 37. 이름이 A로 시작하는 사원들의 이름과 월급을 출력하시오 !

```
[orcl:~]$ awk '{print $2, $6}' emp.txt | grep -i '^a'
```

문제 38. 이름과 직업과 월급을 출력하는데 이름이 A로 시작하는 사원들의 이름을 출력하시오 !

```
[orcl:~]$ awk '{print $2, $3, $6}' emp.txt | grep -i '^a'
```

```
[orcl:~]$ awk 'substr($2,1,1)="A" {print $2,$6,$3}' emp.txt
```

문제 39. 81년도에 입사한 사원들의 이름과 입사일을 출력하시오 !

```
[orcl:~]$ awk 'substr($5,1,2)="81" {print $2,$5}' emp.txt
```

문제 40. 입사일에 81을 포함하지 않는 사원들의 이름과 입사일을 출력하시오 !

```
[orcl:~]$ awk 'substr($5,1,2)!="81" {print $2,$5}' emp.txt
```

```
[orcl:~]$ awk '{print $2, $5}' emp.txt | grep -v 81
```

※ 설명 : -v는 포함되지 않는 것을 검색할 때 사용

문제 41. 겨울왕국 대본에서 elsa가 포함되어져 있는 라인들의 갯수가 전체 몇개인지 출력하시오 !

```
[orcl:~]$ grep -i 'elsa' winter.txt | wc -l
```

문제 42. 위의 결과를 다시 출력하는데 이름과 월급만 나오게 하시오 !

```
[orcl:~]$ sort -k 6 emp.txt | awk '{print $2, $6}'
```

문제 43. 이름과 월급을 출력하는데 월급이 높은것부터 출력하시오 !

```
[orcl:~]$ sort -r -k 6 emp.txt | awk '{print $2, $6}'
```

문제 44. 직업이 salesman인 사원들의 이름과 월급을 출력하는데 월급이 높은것 부터 출력하시오 !

```
[orcl:~]$ awk '$3==toupper("salesman") {print $2, $6,$3}' emp.txt | sort -rk 2
```

문제 45. 사원테이블의 월급의 토탈값을 출력하시오 !

```
[orcl:~]$ awk '{print $6}' emp.txt | awk '{sum += $1} END {print sum}'
```

문제 46. 직업이 salesman인 사원들의 토탈월급을 출력하시오 !

```
[orcl:~]$ awk '$3 ==toupper("salesman") {print $6}' emp.txt | awk '{sum += $1} END {print sum}'
```

문제 47. 아래와 같이 결과가 출력되게 하시오 !

KING의 월급은 5000입니다.  
BLAKE의 월급은 2850입니다.  
CLARK의 월급은 2450입니다.  
JONES의 월급은 2975입니다.

```
[orcl:~]$ awk '{print $2 "의 월급은 " $6 "입니다."}' emp.txt
```

문제 48. 아래와 같이 결과를 출력하시오 !

KING의 직업은 PRESIDENT입니다.  
BLAKE의 직업은 MANAGER입니다.  
CLARK의 직업은 MANAGER입니다.

```
[orcl:~]$ awk '{print $2 "의 직업은 " $3 "입니다."}' emp.txt
```

문제 49. 자기 홈 디렉토리에서 ls -lrt라고 조회한 결과를 분석하시오 !

```
[orcl:~]$ ls -lrt
```

설명 : -l : 리스트  
-r : reverse  
-t : 시간

```
-rw-r--r-- 1 oracle oinstall 1386 Dec 17 12:03 emp.txt
  ↑      ↑      ↑      ↑      ↑      ↑
  권한  링크 소유자  그룹  파일크기 최종변경시간
```

문제 50. 자신의 홈 디렉토리의 파일 리스트를 출력하는데 현재것부터 출력하시오 !

```
[orcl:~]$ ls -lt
```

문제 51. ls -lrt \*.txt 를 했을 때 파일 크기에 해당하는 부분만 출력하시오 !

```
[orcl:~]$ ls -lrt *.txt | awk '{print $5}'
```

문제 52. 위에서 출력된 숫자들을 다 sum해서 출력하시오 !

```
[orcl:~]$ ls -lrt *.txt | awk '{print $5}' | awk '{sum += $1} END {print sum}'
```



문제 53. 아래와 같이 결과를 출력하시오 !

```
The size of total text file under current directory is 112563 bytes
```

```
[orcl:~]$ ls -lrt *.txt | awk '{print $5}' | awk '{sum += $1} END {print "The size of total text file under current directory is " sum " bytes" }'
```

문제 54. 위의 스크립트를 쉘 스크립트로 저장해서 실행되게 하시오 !

```
ls -lrt *.txt | awk '{print $5}' | awk '{sum += $1} END {print "The size of total text file under current directory is " sum " bytes" }'  
[orcl:~]$ vi size.sh  
[orcl:~]$ sh size.sh  
The size of total text file under current directory is 112563bytes
```

문제 55. emp.txt에서 직업만 출력하시오 !

```
[orcl:~]$ awk '{print $3}' emp.txt
```

문제 56. 위의 결과를 abcd순으로 정렬해서 출력하시오 !

```
[orcl:~]$ awk '{print $3}' emp.txt | sort
```

문제 57. 위의 결과를 출력할 때 중복 제거해서 출력하시오 !

```
[orcl:~]$ awk '{print $3}' emp.txt | sort | uniq
```

문제 58. 부서번호를 출력하는데 중복제거해서 출력하시오 !

```
[orcl:~]$ awk '{print $8}' emp.txt | sort | uniq
```

문제 59. emp.txt 를 복사해서 emp7.txt 를 생성하고 emp7.txt에 KING에 대한 데이터를 복사해서 맨 아래에 하나를 입력하고 저장하시오 !

```
[orcl:~]$ cp emp.txt emp7.txt  
[orcl:~]$ vi emp7.txt  
맨 위에(KING)있는 라인에 커서를 대고 yy눌르면 한라인 복사 후 맨밑에서 p눌르면 복사됨
```

문제 60. emp7.txt 에서 중복된 라인만 출력하시오 !

```
[orcl:~]$ sort emp7.txt | uniq -d
```

문제 61. emp7.txt 에서 중복된 라인을 제거하고 출력하시오 !

```
[orcl:~]$ sort emp7.txt | uniq -u
※ KING아예 출력 안됨
```

문제 62. emp7.txt에서 중복된 라인을 제거할 때 다 없어지지말고 하나는 남겨놓게 하시오 !

```
[orcl:~]$ sort emp7.txt | uniq
```

문제 63. 직업을 물어보게 하고 직업을 입력하면 해당 직업을 갖는 사원들의 이름과 직업이 출력되게 하시오 !  
(vi 편집기 명령어를 모르니까 일단 메모장에 작업하고 복사해서 리눅스에 붙여넣기로 하세요 !)

```
#!/bin/bash                                # 리눅스 스크립트 중에서 shell 스크립트를 사용하겠다.

echo "          "                        # 공백 출력
echo -n "직업을 입력하세요 ~ "          # -n 빼보고 실행해 본다 (엔터 키값???)
read job                                # 위에서 입력하고 엔터를 친 값이 job변수에 들어간다.
echo "          "
grep -i $job emp.txt | awk '{print $2, $3}'
```

위의 스크립트를 붙여넣고 저장하고 나와서 ( \$vi job2.sh 에다가)

문제 64. 부서번호를 물어보게 하고 부서번호를 입력하면 해당 부서번호에서 근무하는 사원들의 이름과 월급과 부서번호가 출력되게 하시오 !

답 1 :

```
#!/bin/bash

echo "          "
echo -n "부서번호를 입력하세요 ~ "
read deptno
echo "          "
awk '{print $2,$6,$8}' emp.txt | grep -iw $deptno
```

답 2 :

```
#!/bin/bash

echo "          "
echo -n "부서번호를 입력하세요 ~ "
read deptno
echo "          "
awk '$8==$deptno'{print $2,$6,$8}' emp.txt | grep -iw $deptno
```

문제 65. 이름을 물어보게하고 이름을 입력하면 해당 사원의 월급이 아래와 같이 출력되게 하는 쉘 스크립트를 작성하시오 !

```
vi find_sal.sh
```

```
#!/bin/bash

echo "          "
echo -n "이름을 입력하세요~"
```

```

read ename
echo "      "

grep -i $ename emp.txt | awk '{print $2,"의 월급은 ",$6"입니다."}'

[orcl:~]$ sh find_sal.sh

이름을 입력하세요~
scott

SCOTT 의 월급은 3000입니다.

```

문제 66. dept.txt를 리눅스 시스템에 자기 홈 디렉토리에 올리시오 !

```

10 ACCOUNTING      NEWYORK
20 RESEARCH        DALLAS
30 SALES            CHICAGO
40 OPERATIONS      BOSTON

```

```

-----
vi  팁
dd 누르면 줄제거
쉬프트 zz 저장하고 나가기
a누르고 복붙
-----

```

문제 67. 부서번호를 물어보게하고 부서번호를 입력하면 해당 부서번호의 부서위치가 출력되게 하는 쉘 스크립트를 작성하시오 !

```

[orcl:~]$ vi find_loc.sh

#!/bin/bash

echo "      "
echo -n "부서번호를 입력하세요 ~"
read deptno
echo "      "

grep $deptno dept.txt | awk '{print $3}'

[orcl:~]$ sh find_loc.sh

부서번호를 입력하세요 ~
10

NEWYORK

```

문제 68. 이름을 물어보게 하고 이름을 입력하면 해당 사원의 부서위치가 출력되게 하시오 !

```

[orcl:~]$ vi find_loc2.sh

#!/bin/bash

echo "      "

```

```
echo -n "이름을 입력하세요 ~"
read ename
echo "      "
deptno=`grep -i $ename emp.txt | awk '{print $8}'`
grep -i $deptno dept.txt | awk '{print $3"에서 근무하는 직원입니다."}'
```

#여기서 `가 포인트!(공백제거)

```
[orcl:~]$ sh find_loc2.sh
```

이름을 입력하세요 ~scott

DALLAS에서 근무하는 직원입니다.

문제 69. 이름을 물어보게하고 이름을 입력하면 해당 사원의 부서명이 출력되게 하시오 !

```
[orcl:~]$ vi find_dname.sh
```

```
#!/bin/bash
```

```
echo "      "
echo -n "이름을 입력하세요 ~"
read ename
echo "      "
deptno=`grep -i $ename emp.txt | awk '{print $8}'`
grep -i $deptno dept.txt | awk '{print $2"에서 근무하는 직원입니다."}'
```

```
[orcl:~]$ sh find_dname.sh
```

이름을 입력하세요 ~scott

RESEARCH에서 근무하는 직원입니다.

문제 70. 직업을 물어보게 하고 직업을 입력하면 해당 직업인 직원들의 월급이 출력되게 하시오 !

```
[orcl:~]$ vi job_sal.sh
```

```
#!/bin/bash
```

```
echo "      "
echo -n "직업을 입력하세요 ~"
read job
echo "      "
sal=`grep -i $job emp.txt | awk '{print $6}'`
echo "$sal"
```

```
[orcl:~]$ sh job_sal.sh
```

직업을 입력하세요 ~salesman

1250  
1600  
1500  
1250

문제 71. 이번에는 이름도 같이 출력하시오 !

```
[orcl:~]$ vi job_sal2.sh

#!/bin/bash

echo "      "
echo -n "직업을 입력하세요 ~"
read job
echo "      "
sal=`grep -i $job emp.txt | awk '{print $2, $6}'`
echo "$sal"

[orcl:~]$ sh job_sal2.sh

직업을 입력하세요 ~salesman

MARTIN 1250
ALLEN 1600
TURNER 1500
WARD 1250
```

문제 72. 아래와 같이 부서위치를 물어보게 하고 부서위치를 입력하면 해당 사원의 이름과 부서위치가 출력되게 하시오 !

문제 73. 아래와 같이 두개의 파일 이름을 각각 물어보게 하고 결과 파일의 내용의 차이가 출력되게 하시오 !

```
[orcl:~]$ vi diff.sh

#!/bin/bash

echo "      "
echo -n "비교할 첫번째 파일명을 입력하세요 ~"
read file1
echo -n "비교할 두번째 파일명을 입력하세요 ~"
read file2
echo "      "
diff $file1 $file2

[orcl:~]$ sh diff.sh

비교할 첫번째 파일명을 입력하세요 ~emp.txt
비교할 두번째 파일명을 입력하세요 ~emp7.txt

14a15
> 7839 KING          PRESIDENT          0 81/11/17    5000          0          10
```

문제 74. /home/oracle 밑에 test100과 test200 이라는 디렉토리를 생성하고 그 밑에 emp.txt를 각각 복사하시오 !

```
[orcl:~]$ cd
[orcl:~]$ mkdir test100 test200
[orcl:~]$ cp emp.txt ./test100/emp.txt
[orcl:~]$ cp emp.txt ./test200/emp.txt
```

↑  
현재 디렉토리

문제 75. /home/oracle 밑에 있는 emp.txt 를 검색하시오 !

```
[orcl:~]$ find /home/oracle -name 'emp.txt' -print
/home/oracle/test100/emp.txt
/home/oracle/test200/emp.txt
/home/oracle/emp.txt

[orcl:~]$ find /home/oracle -maxdepth 1 -name 'emp.txt' -print
/home/oracle/emp.txt
```

문제 76. (점심시간 문제) 파일을 검색하는 쉘 스크립트를 아래와 같이 생성하시오 !

```
[orcl:~]$ vi find_file.sh

#!/bin/bash

echo "      "
echo -n "현재 디렉토리 밑에 검색할 파일명을 입력하세요 ! "
read file
echo -n "depth는 어떻게 할까요? "
read depth
echo "      "
dir=`pwd`
find $dir -maxdepth $depth -name $file -print

[orcl:~]$ sh find_file.sh

현재 디렉토리 밑에 검색할 파일명을 입력하세요 ! emp.txt
depth는 어떻게 할까요? 1

./emp.txt
```

문제 77. /home/oracle 밑에 test77이라는 디렉토리를 만들고 그 밑에 emp.txt 를 복사해서 아래의 파일명으로 붙여넣으시오 !

```
emp1.txt emp2.txt emp3.txt emp4.txt

[orcl:~]$ cd
[orcl:~]$ mkdir test77
[orcl:~]$ cp emp.txt ./test77/emp1.txt
[orcl:~]$ cp emp.txt ./test77/emp2.txt
[orcl:~]$ cp emp.txt ./test77/emp3.txt
[orcl:~]$ cp emp.txt ./test77/emp4.txt
[orcl:~]$ ls -l ./test77/*.txt
-rw-r--r-- 1 oracle oinstall 1386 Dec 18 14:00 ./test77/emp1.txt
-rw-r--r-- 1 oracle oinstall 1386 Dec 18 14:00 ./test77/emp2.txt
-rw-r--r-- 1 oracle oinstall 1386 Dec 18 14:00 ./test77/emp3.txt
-rw-r--r-- 1 oracle oinstall 1386 Dec 18 14:00 ./test77/emp4.txt
```

문제 78. /home/oracle 밑에 test77 밑에 있는 확장자 .txt 파일들을 all.tar라는 이름으로 압축하시오 !

```
[orcl:~]$ tar cvf ./test77/*.txt
./test77/emp2.txt
./test77/emp3.txt
./test77/emp4.txt
```

문제 79. /home/oracle 밑에 test79 라는 디렉토리를 생성하고 그 밑에 all.tar 압축파일의 압축을 해제하시오 !

```
[orcl:~]$ mkdir test79
[orcl:~]$ tar xvf all.tar -C /home/oracle/test79
./test77/emp1.txt
./test77/emp2.txt
./test77/emp3.txt
./test77/emp4.txt
```

문제 80. 겨울왕국 대본인 winter.txt를 winter1.txt ~ winter3.txt로 copy를 하고서 winter1.txt ~ winter3.txt 를 압축해서 winter.tar라는 이름의 압축파일을 생성하시오 !

```
[orcl:~]$ mkdir test80
[orcl:~]$ cp winter.txt /home/oracle/test80/winter1.txt
[orcl:~]$ cp winter.txt /home/oracle/test80/winter2.txt
[orcl:~]$ cp winter.txt /home/oracle/test80/winter3.txt
[orcl:~]$ tar cvf winter.tar /home/oracle/test80
tar: Removing leading `/' from member names
/home/oracle/test80/
/home/oracle/test80/winter2.txt
/home/oracle/test80/winter1.txt
/home/oracle/test80/winter3.txt
```

문제 81. 아래의 디렉토리 밑에 있는 salgrade.txt를 나의 홈 디렉토리 에서도 편하게 열어볼 수 있도록 바로가기를 나의 홈 디렉토리에 만드시오 !

```
/home/oracle/test22/test23/test24/test25/test26/test27
```

답 :

```
[orcl:~]$ ln -s /home/oracle/test22/test23/test24/test25/test26/test27/salgrade.txt
salgrade.txt
```

문제 82. 위의 변경된 내용대로 출력한 내용을 emp9.txt로 저장하시오 !

```
[orcl:~]$ sed 's/KING/yyy/' emp.txt >> emp9.txt
```

문제 83. /home/oracle 밑에 있는 확장자 .txt파일들을 전부 /home/oracle/backup이라는 폴더를 만들고 거기에 복사하시오 !

```
[orcl:~]$ cd
[orcl:~]$ mkdir backup
[orcl:~]$ cp /home/oracle/*.txt /home/oracle/backup/
[orcl:~]$ ls -l /home/oracle/backup/
total 140
-rw-r--r-- 1 oracle oinstall 102 Dec 18 14:41 dept.txt
-rw-r--r-- 1 oracle oinstall 1386 Dec 18 14:41 emp.txt
```

```
-rw-r--r-- 1 oracle oinstall 1386 Dec 18 14:41 emp400.txt
-rw-r--r-- 1 oracle oinstall 1485 Dec 18 14:41 emp7.txt
-rw-r--r-- 1 oracle oinstall 1385 Dec 18 14:41 emp9.txt
-rw-r--r-- 1 oracle oinstall 65 Dec 18 14:41 salgrade.txt
-rw-r--r-- 1 oracle oinstall 111177 Dec 18 14:41 winter.txt
```

문제 84. emp.txt를 다시 자기 홈 디렉토리로 이동시키시오 !

```
[orcl:~]$ mv ./backup/emp.txt ./
[orcl:~]$ ls emp.txt
emp.txt
```

문제 85. 겨울 왕국 대본의 1 ~ 200번째 까지의 내용을 winter200.txt 라는 이름으로 저장하시오 !

```
[orcl:~]$ vi winter.txt

: 1,200 w winter200.txt
"winter200.txt" [New] 200L, 6350C written
```

문제 86. 겨울왕국 대본을 열어서 위에 한행만 남기고 전체 대본을 다 지우시오 !

```
[orcl:~]$ vi winter.txt

:2,4240 d
```

문제 87. 겨울 왕국 대본의 전체 라인수를 먼저 확인하고 겨울왕국 대본 전체를 복사해서 아래에 붙여넣고 겨울 왕국 대본의 라인수가 2배가 되었는지 확인하시오 !

문제 88. 구구단 2단을 수행하는 파이썬 코드를 작성하고 아래와 같이 리눅스에서 실행하면 결과가 출력되게 하시오 !

```
[orcl:~]$ vi p_2_dan.py
for i in range(1, 10):
    print('2 x %d = %d'%(i,i*2))

[orcl:~]$ python p_2_dan.py
2 x 1 = 2
2 x 2 = 4
2 x 3 = 6
2 x 4 = 8
2 x 5 = 10
2 x 6 = 12
2 x 7 = 14
2 x 8 = 16
2 x 9 = 18
```

문제 89. 아래와 같이 숫자를 물어보게 하고 숫자를 입력하면 세모가 출력되게 하시오 !

숫자를 입력하세요 ~ 7



```
#-*- coding: utf-8 -*-
```

```
a = int(input('숫자를 입력하세요 ~'))  
for i in range(1,a):  
    print ((a-i)*' '+'i*'★ ')
```

문제 90. (리눅스의 파이썬을 이용해서 구현) 스마일 게이트의 알고리즘 시험문제  
5!를 코드로 구현하시오 !(코드는 파이썬, c언어, java)

```
[orcl:~]$ vi factorial.py
```

```
a = int(input('input number : '))  
b = 1  
for i in range(1,a+1):  
    b *= i  
print(b)
```

```
[orcl:~]$ python factorial.py  
input number : 5  
120
```

문제 91. BLAKE의 직업을 MANAGER에서 SALESMAN으로 변경하시오 !

```
:s/MANAGER/SALESMAN
```

문제 92. emp.txt에서 직업이 MANAGER인 직원들은 전부 SALESMAN으로 변경하시오 !

```
:%s/MANAGER/SALESMAN/g
```

문제 93. emp.txt를 emp1.txt ~ emp20.txt로 복사하시오 !

```
[orcl:~]$ vi cp_emp.sh
```

```
#!/bin/bash  
  
i=1  
while [ $i -le 20 ]  
do  
    cp emp.txt emp$i.txt  
    let i=i+1  
done
```

```
[orcl:~]$ sh cp_emp.sh
```

문제 94. emp1.txt ~ emp20.txt 의 내용을 변경하는데 SALESMAN을 jjj로 변경하시오 !

```
[orcl:~]$ vi emp*.txt
```

```
:argdo %s/SALESMAN/jjj/g |update
```

문제 95. winter.txt 대본을 100개로 복사하고 winter.txt ~ winter100.txt 로 복사한다.  
winter.txt ~ winter100.txt 의 내용중에 by 를 bybyby로 한번에 변경하시오 !

```
[orcl:~]$ vi cp_winter.sh

#!/bin/bash

i=1
while [ $i -le 100 ]
do
    cp winter.txt winter$i.txt
    let i=i+1
done

[orcl:~]$ sh cp_winter.sh

[orcl:~]$ vi winter*.txt
:argdo %s/by/bybyby/g |update
```

- 숫자열 비교

1. 숫자1 -eq 숫자2 : 두 숫자가 같으면 True
2. 숫자1 -ne 숫자2 : 두 숫자가 같지 않으면 True
3. 숫자1 -gt 숫자2 : 숫자1이 숫자2보다 크다면 True
4. 숫자1 -ge 숫자2 : 숫자1이 숫자2보다 크거나 같다면 True
5. 숫자1 -lt 숫자2 : 숫자1이 숫자2보다 작다면 True
6. 숫자1 -le 숫자2 : 숫자1이 숫자2보다 작거나 같으면 True
7. !숫자1 : 숫자1이 거짓이라면 True

문제 96. winter1.txt ~ winter100.txt를 전부 삭제하시오 !

```
$ vi rm_winter.sh

#!/bin/bash

i=1
while [ $i -le 100 ]
do
    rm winter$i.txt
    let i=i+1
done

$ sh rm_winter.sh
```

문제 97. emp.txt 의 소유자의 권한을 읽기만 가능하도록 수정하시오 !

```
[orcl:~]$ chmod u-w emp.txt
[orcl:~]$ ls -l emp.txt
-r--r--r-- 1 oracle oinstall 1366 Dec 19 10:47 emp.txt
```

※ 권한관리 단축어 정리

- |             |           |
|-------------|-----------|
| 1. u : 유저   | 1. r : 읽기 |
| 2. g : 그룹   | 2. w : 쓰기 |
| 3. o : 기타유저 | 3. x : 실행 |

설명 : read only 파일도 !(느낌표) 를 붙이고 저장을 하면 강제종료되어서 저장된다.  
수정하고 --> :wq!

문제 98. emp.txt의 소유자의 권한에 읽기 권한도 없애시오 !

```
[orcl:~]$ chmod u-r emp.txt
[orcl:~]$ ls -l emp.txt
----r--r-- 1 oracle oinstall 1370 Dec 19 11:46 emp.txt
```

문제 99. emp.txt 의 소유자의 권한에 읽기, 쓰기, 실행권한을 넣으시오

```
[orcl:~]$ chmod u+rwX emp.txt
[orcl:~]$ ls -l emp.txt
-rwxr--r-- 1 oracle oinstall 1370 Dec 19 11:46 emp.txt
```

문제 100. emp.txt 에 대해서 소유자, 그룹, 기타 유저에 대해서 읽기, 쓰기, 실행권한을 모두 넣으시오 !

```
[orcl:~]$ chmod ugo+rwX emp.txt
[orcl:~]$ ls -l emp.txt
-rwxrwxrwx 1 oracle oinstall 1370 Dec 19 11:46 emp.txt
```

문제 101. oracle 유저(소유자)는 emp.txt에 대해서 무조건 읽기만 할 수 있게 하고 쓰지는 못하게 하며 그룹과 다른 기타 유저에 대해서는 읽기, 쓰기, 실행을 다 못하게 하시오 !

```
[orcl:~]$ chmod u-wx,go-rwx emp.txt
[orcl:~]$ ls -l emp.txt
-r----- 1 oracle oinstall 1370 Dec 19 11:46 emp.txt
```

문제 102. (점심시간 문제) 리눅스에서 파이썬으로 아래의 그림을 그리시오 !

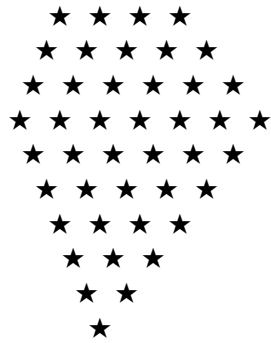
```
[orcl:~]$ vi star.py

#-*- coding: utf-8 -*-

a = int(input('숫자 입력 : '))
for i in range(1,a+1):
    print ((a-i)*' ' + i*'★')
for i in range(a-1, 0, -1):
    print((a-i)*' ' + i*'★')

[orcl:~]$ python star.py
숫자를 입력하세요 7

  ★
 ★ ★
★ ★ ★
```



문제 103. oracle 유저가 emp.txt에 대해서 chmod명령어 자체를 수행하지 못하게 하시오 !

root 유저 (최상위 유저)

↓

oracle, test01 ~ test014

\* 내가 누구인지 확인하는 방법 ?

\$ whoami

\* 최상위 계정인 root 유저로 접속하는 방법?

\$ su -

패스워드 : oracle

[orcl:~]\$ su -

Password:

[root@edydr1p0 ~]# whoami

root

[root@edydr1p0 ~]# pwd

/root

[root@edydr1p0 ~]# cd /home/oracle

[root@edydr1p0 oracle]# pwd

/home/oracle

[root@edydr1p0 oracle]# ls -l emp.txt

-r----- 1 oracle oinstall 1370 Dec 19 11:46 emp.txt

[root@edydr1p0 oracle]# chmod +i emp.txt

\* 다시 oracle 유저로 접속해서 chmod 명령어를 emp.txt 에 수행해 본다.

[root@edydr1p0 oracle]# su - oracle

[orcl:~]\$ chmod u-r emp.txt

chmod: changing permissions of `emp.txt': Operation not permitted

문제 104. 다시 oracle 계정이 emp.txt 파일을 chmod로 권한수정할 수 있도록 하게 하시오 !

[orcl:~]\$ su -

Password:

[root@edydr1p0 ~]# cd /home/oracle

[root@edydr1p0 oracle]# pwd

/home/oracle

[root@edydr1p0 oracle]# chmod -i emp.txt

[root@edydr1p0 oracle]# su - oracle

[orcl:~]\$ chmod u-r emp.txt

[orcl:~]\$ ls -l emp.txt

----- 1 oracle oinstall 1370 Dec 19 11:46 emp.txt

문제 105. emp.txt에 대해서 아래와 같이 권한이 설정이 될 수 있도록 하시오 !

```
$ ls -l emp.txt
-r--r--r--

[orcl:~]$ chmod 444 emp.txt
[orcl:~]$ ls -l emp.txt
-r--r--r-- 1 oracle oinstall 1370 Dec 19 11:46 emp.txt
```

문제 106. winter.txt에 대해서 권한을 아래와 같이 설정이 될 수 있도록 하시오 !

```
$ ls -l winter.txt
-rwxr--r--

[orcl:~]$ chmod 744 winter.txt
[orcl:~]$ ls -l winter.txt
-rwxr--r-- 1 oracle oinstall 222718 Dec 19 11:15 winter.txt
```

문제 107. 다시 emp.txt의 소유자를 oracle로 변경하고 그룹은 oinstall로 변경하시오 !

```
$ su -
$ cd /home/oracle
$ chown oracle:oinstall emp.txt
$ exit
$ ls -l emp.txt
-r--r--r-- 1 oracle oinstall 1370 Dec 19 11:46 emp.txt
```

문제 108. /home/oracle밑에 bbb라는 디렉토리를 만들고 bbb라는 디렉토리 밑에 /home/oracle밑에 있는 확장자 .txt 파일들을 전부 copy하시오 !

```
[orcl:~]$ mkdir bbb
[orcl:~]$ cp *.txt ./bbb/
[orcl:~]$ cd bbb
[orcl:bbb]$ ls
dept.txt  emp10.txt  emp13.txt  emp16.txt  emp19.txt  emp3.txt  emp500.txt  emp8.txt  winter.txt
emp.txt   emp11.txt  emp14.txt  emp17.txt  emp2.txt   emp4.txt  emp6.txt   emp9.txt  winter200.txt
emp1.txt  emp12.txt  emp15.txt  emp18.txt  emp20.txt  emp5.txt  emp7.txt   salgrade.txt
```

문제 109. /home/oracle 밑에 bbb 디렉토리의 소유자를 root유저로 변경하시오

```
[orcl:~]$ ls -ld /home/oracle/bbb
drwxr-xr-x 2 oracle oinstall 4096 Dec 19 14:41 /home/oracle/bbb
```

↑  
디렉토리의 소유자가 oracle  
디렉토리의 소유자를 바꾼다는 것은 디렉토리 안의 데이터들의 소유자도 바꾼다는 것!

```
[orcl:~]$ su -
Password:
[root@edydr1p0 ~]# cd /home/oracle
[root@edydr1p0 oracle]# chown -R root:root /home/oracle/bbb
-R은 그 디렉토리 안에있는 모든파일들의 소유자를 싹 변경 하겠다.
```

만약 -R 옵션을 안붙이면 디렉토리 소유자만 변경되고 파일들은 변경 안된다.

```
[root@edydr1p0 oracle]# ls -ld /home/oracle/bbb
drwxr-xr-x 2 root root 4096 Dec 19 14:41 /home/oracle/bbb
[root@edydr1p0 bbb]# ls -lrt
total 328
-rw-r--r-- 1 root root 6366 Dec 19 14:41 winter200.txt
-rwxr--r-- 1 root root 222718 Dec 19 14:41 winter.txt
-rw-r--r-- 1 root root 65 Dec 19 14:41 salgrade.txt
-rw-r--r-- 1 root root 1386 Dec 19 14:41 emp9.txt
:
```

문제 110. 다시 /home/oracle/bbb 디렉토리의 소유자를 oracle 로 변경하시오 !  
(소유자 : oracle, 그룹 : oinstall )

```
[root@edydr1p0 bbb]# chown -R oracle:oinstall /home/oracle/bbb
```

문제 111. 위의 txt 파일의 총 크기를 출력하시오 !

```
[orcl:~]$ du -s *.txt | awk '{sum = sum + $1} END {print sum}'
328
```

또는

```
[orcl:~]$ du -c *.txt
4      dept.txt
4      emp.txt
:
:
4      salgrade.txt
224    winter.txt
8      winter200.txt
328    total
```

문제 112. 나의 프로세서의 cpu를 양보하는 친절도 확인하는 방법 ?

```
[orcl:~]$ ps
  PID TTY          TIME CMD
 4996 pts/1    00:00:00 bash<---- 4996 번호 확인
 7682 pts/1    00:00:00 vim
 8779 pts/1    00:00:00 bash
10614 pts/1    00:00:00 ps
```

```
[orcl:~]$ top -p 7787
  PID USER      PR  NI  VIRT  RES  SHR S %CPU %MEM    TIME+  COMMAND
 4996 oracle   16   0  4676 1448 1164 S   0.0   0.1   0:00.23 bash
```

문제 113. sar 1 1000 결과화면을 sar\_20181219.txt 로 저장하시오 !

```
[orcl:~]$ sar 1 1000 >> sar_20181219.txt
```

문제 114. 위의 명령어가 그냥 백그라운드로 실행되게 하시오 !

```
[orcl:~]$ sar 1 1000 >> sar_20181219.txt &  
[1] 10778      <----- job 번호
```

문제 115. sar\_20181219.txt 파일의 사이즈가 점점 커지고 있는지 확인하시오 !

```
[orcl:~]$ ls -l sar_20181219.txt  
-rw-r--r-- 1 oracle oinstall 16842 Dec 19 16:37 sar_20181219.txt  
[orcl:~]$ ls -l sar_20181219.txt  
-rw-r--r-- 1 oracle oinstall 17088 Dec 19 16:37 sar_20181219.txt  
:  
:
```

문제 116. sar\_20181219.txt의 뒷부분의 데이터를 실시간으로 볼 수 있게 하시오 !

```
[orcl:~]$ tail -f sar_20181219.txt  
16:39:25      all      0.00      0.00      0.50      0.00      0.00      99.50  
16:39:26      all      0.50      0.00      1.00      0.00      0.00      98.51  
16:39:27      all      0.00      0.00      1.00      0.00      0.00      99.00
```

문제 117. (리눅스에서 파이썬으로 구현하세요)

리스트의 메소드를 이용하지 말고 loop문으로 직접 코드를 작성해서 아래의 결과를 출력하시오 !  
(오늘의 마지막 문제 댓글로 올리세요 !)

```
a = raw_input('input string : ')  
b = ''  
for i in range(1, len(a)+1):  
    b += a[-i]  
print b
```

문제 118. jobs 명령어를 수행하면 아래와 같이 결과가 출력되게 상황을 만드시오 !

```
[orcl:~]$ jobs  
[1]  Stopped                  vim hhh.txt  
[2]  Running                  sar 1 1000 >> sar_20181220.txt &  
[3]- Stopped                  vim hhh2.txt  
[4]+ Stopped                  vim hhh3.txt  
  
[orcl:~]$ vi hhh2.txt  
  
[3]+ Stopped                  vim hhh2.txt  
[orcl:~]$ vi hhh3.txt  
  
[4]+ Stopped                  vim hhh3.txt
```

문제 119. 위의 일시 중단된 3번 job으로 fg하려면 어떻게 해야 하는가?

```
[orcl:~]$ fg 3  
vim hhh2.txt
```

```

[3]+ Stopped          vim hhh2.txt
[orcl:~]$ jobs
[1] Stopped          vim hhh.txt
[2] Running          sar 1 1000 >> sar_20181220.txt &
[3]+ Stopped          vim hhh2.txt
[4]- Stopped          vim hhh3.txt

```

※ 설명 : + --> 현재 job  
 - --> 현재 job 이전에 현재 job 있었던 job

문제 120. 4번 job에 fg하고 vi로 작업했던 작업을 정상적으로 저장하고 나오시오 !

```

[orcl:~]$ fg 4
vim hhh3.txt

ctrl + z z

[orcl:~]$ jobs
[1]- Stopped          vim hhh.txt
[2] Running          sar 1 1000 >> sar_20181220.txt &
[3]+ Stopped          vim hhh2.txt

```

문제 121. 터미널창을 하나 더 열고 오라클에 접속해서 악성 SQL을 하나 수행하시오 !

```

[orcl:~]$ sqlplus / as sysdba

SQL> select count(*)
2      from sh.sales, sh.sales, sh.sales;

  PID USER      PR  NI  VIRT  RES  SHR S %CPU %MEM    TIME+  COMMAND
  ---
4918 oracle   15   0  197m  20m  11m S   0.7   1.0   0:12.87 oraagent.bin
4292 oracle   15   0  334m  29m  18m S   0.3   1.4   0:05.56 ohasd.bin
5118 oracle   15   0  410m  14m  13m S   0.3   0.7   0:00.19 oracle
5777 oracle   15   0  2336 1060  796 R   0.3   0.1   0:00.62 top
    1 root     15   0  2072  624  532 S   0.0   0.0   0:00.57 init

```

문제 122. 현재 리눅스 시스템에서 실행중인 모든 프로세서를 출력하시오 !

```

[orcl:~]$ ps -ef

```

문제 123. 터미널 창을 하나 더 열고 sar 1 1000 명령어를 수행하시오

문제 124. 지금 죽은 터미널 창에서 vi 편집기 명령어로 아래의 작업을 수행하고 있으시오 !

```

$ vi ggg.txt

select ename, sal,

```



문제 125. 지금 작업을 vim으로 수행하고 있는 프로세서들을 출력하시오 !

```
[orcl:~]$ ps -ef | grep vim
oracle   6036   5935   0 10:39 pts/2    00:00:00 vim ggg.txt
oracle   6047   5271   0 10:40 pts/1    00:00:00 grep vim
```

```
[orcl:~]$ ps -ef | grep vim | grep -v grep
oracle   6036   5935   0 10:39 pts/2    00:00:00 vim ggg.txt
    ※ 자기 자신을 제외하고 출력(-v 검색 안되게 하겠다.)
```

```
[orcl:~]$ kill -9 6036
```

문제 126. kill 옵션 중 -2 옵션을 테스트 하시오 !

```
- 터미널창 1
$ sar 1 10000
```

```
- 터미널창 2
$ ps -ef |grep sar    <----- 프로세서 번호 확인
$ kill -2 프로세서 번호
```

문제 127. a.sh에 소유자가 실행할 수 있도록 실행권한을 넣으시오 !

```
[orcl:~]$ chmod u+x a.sh
[orcl:~]$ ll a.sh
-rwxr--r-- 1 oracle oinstall 31 Dec 20 11:41 a.sh
[orcl:~]$ ./a.sh
host name : edydr1p0.us.oracle.com
```

설명 : ./파일명.sh으로 실행하려면 실행권한이 있어야 된다.  
sh 파일명.sh 은 파일의 실행권한이 없어도 실행된다.

문제 128. num1 과 num2의 곱을 구하시오 !

```
[orcl:~]$ num1=100
[orcl:~]$ num2=200
[orcl:~]$ expr $num1 W* $num2
20000
```

문제 129. (점심시간 문제) 아래의 계산식을 구현하시오 !

```
($num2 + 200) W* $num1

[orcl:~]$ expr W( $num2 + 200 W) W* $num1
40000
```

문제 130. 아래와 같이 두 정수를 파라미터 값으로 입력 했을 때 나눈 나머지 값이 출력되게 하시오!

```
[orcl:~]$ vi f.sh
```

```
num1=$1
num2=$2
num3=`expr $num1 + $num2`
echo "$num1와 $num2를 더하면 $num3 입니다."
```

```
[orcl:~]$ sh f.sh 18 24
18와 24를 더하면 42 입니다.
```

※ 설명 : 변수=`리눅스 명령어`  
리눅스 명령어에 의해서 수행된 결과가 변수에 입력 되어야 한다면 역 따옴표를 사용해야 한다.

문제 131. 아래의 두 수를 각각 물어보게 하고 아래의 두 수를 곱한 결과가 출력되게 하시오 !

```
[orcl:~]$ vi two_number.sh 12 54
num3=`expr $1 W* $2`
echo "$1 와 $2 를 곱하면 $num3 입니다."
```

```
[orcl:~]$ sh two_number.sh 12 54
12 와 54 를 곱하면 648 입니다.
```

문제 132. 위의 스크립트를 파라미터 변수를 이용해서 아래와 같이 실행될 수 있게 하시오 !

```
[orcl:~]$ vi if1.sh
#!/bin/bash

if [ $1 -eq $2 ]; then
    echo "$1 is equal to $2"
else
    echo "$1 is not equal to $2"
fi
```

```
[orcl:~]$ sh if1.sh 100 200
100 is not equal to 200
[orcl:~]$ sh if1.sh 100 100
100 is equal to 100
```

문제 133. 아래와 같이 실행되게 쉘 스크립트를 작성하시오 !

```
[orcl:~]$ sh if2.sh 100 200
100 is little than 200
[orcl:~]$ sh if2.sh 200 100
200 is greater than 100
[orcl:~]$ sh if2.sh 200 200
200 is equal to 200
```

```
[orcl:~]$ vi if2.sh
#!/bin/bash

if [ $1 -gt $2 ]; then
    echo "$1 is greater than $2"
elif [ $1 -lt $2 ]; then
    echo "$1 is little than $2"
else
```

```
    echo "$1 is equal to $2"
fi
```

문제 134. 아래와 같이 쉘 스크립트를 실행하면 아래와 같은 결과가 출력되게 하시오 !

```
[orcl:~]$ sh find_sal.sh scott
SCOTT 의 월급은 3000입니다.

[orcl:~]$ vi find_sal.sh
ename=$1
grep -i $ename emp.txt | awk '{print $2,"의 월급은",$6"입니다."}'
```

문제 135. 아래와 같이 실행했을 때 해당 사원의 부서번호가 30번이고 월급이 2000 보다 작다면  
"월급 인상 대상자 입니다." 라는 메시지가 나오게 하고 그렇지 않다면  
"월급 인상 대상자가 아닙니다." 라는 메시지가 출력되게 하시오 !

```
[orcl:~]$ vi find_sal.sh
#!/bin/bash

ename=$1
sal=`grep -i $ename emp.txt | awk '{print $6}'`
deptno=`grep -i $ename emp.txt | awk '{print $8}'`
if [ $sal -lt 2000 -a $deptno -eq 30]; then
    echo "월급 인상 대상자 입니다."
else
    echo "월급인상 대상자가 아닙니다."
fi

[orcl:~]$ sh find_sal.sh miller
월급인상 대상자가 아닙니다.
[orcl:~]$ sh find_sal.sh martin
월급 인상 대상자 입니다.
```

문제 136. 위의 코드를 수정해서 구구단 2단을 출력하시오 !

```
[orcl:~]$ vi for.sh
#!/bin/bash

gop=1
for i in {1..9}
do
    gop=`expr 2 W* $i`
    echo "2 x $i = $gop"
done

[orcl:~]$ sh for.sh
2 x 1 = 2
2 x 2 = 4
2 x 3 = 6
2 x 4 = 8
2 x 5 = 10
2 x 6 = 12
2 x 7 = 14
2 x 8 = 16
```

$$2 \times 9 = 18$$

문제 137. 구구단 2단부터 9단까지 출력하시오 !

```
[orcl:~]$ vi for.sh
#!/bin/bash

gop=1
for i in {2..9}
do
    for j in {1..9}
    do
        gop=`expr $i W* $j`
        echo "$i x $j = $gop"
    done
done

[orcl:~]$ sh for.sh
2 x 1 = 2
2 x 2 = 4
2 x 3 = 6
2 x 4 = 8
    :
9 x 7 = 63
9 x 8 = 72
9 x 9 = 81
```

문제 138. 아래의 별표를 출력하시오 !

```
[orcl:~]$ vi star.sh
star=""
for i in {1..5}
do
    star="$star★"
    echo $star
done

[orcl:~]$ sh star.sh
★
★★
★★★
★★★★
★★★★★
```

문제 139. 위의 코드를 수정해서 숫자를 물어보게 하고 숫자를 입력하면 해당 숫자만큼 ★이 출력되게 하시오 !

```
[orcl:~]$ vi start.sh
echo -n "input number : "
read num
star=""
for i in `eval echo {1..$num}`
do
    star="$star★"
    echo $star
```

```
done
```

```
[orcl:~]$ sh start.sh
input number : 4
★
★★
★★★
★★★★
```

※ 설명 : eval 명령어는 문자열을 명령문으로 인식하고 실행하는 명령어

```
[orcl:~]$ eval echo {1..10}
1 2 3 4 5 6 7 8 9 10
```

문제 140. (오늘의 마지막 문제) 숫자를 물어보게 하고 아래와 같이 결과를 출력하시오 !

```
[orcl:~]$ vi star.sh
echo -n "input number : "
read num
star=""
for i in `eval echo {$num..1}`
do
    for j in `eval echo {1..$i}`
    do
        star="$star★"
    done
    echo $star
    star=""
done
```

```
[orcl:~]$ sh star.sh
input number : 5
★★★★★
★★★★★
★★★★★
★★★
★★
★
```

문제 141. emp.txt 를 for loop문을 이용해서 emp1.txt ~ emp100.txt 로 복사하시오 !

```
[orcl:~]$ vi cp_emp.sh
#!/bin/bash

i=1
while [ $i -le 100 ]
do
    cp emp.txt emp$i.txt
    let i=i+1
done

[orcl:~]$ sh cp_emp.sh
```

문제 142. 짝공하고 자리를 바꿔서 emp1.txt ~ emp100.txt 중 하나 또는 여러개의 파일을 랜덤으로 골라서 데이터를 변경하고 자리로 돌아오시오 !

```
$ clear
```

— 짝공이 100개의 파일중에 무엇을 변경했는지 한번에 알아내는 방법

문제 143. emp3.txt 를 열어서 3000 을 3900으로 변경하시오 !

```
[orcl:~]$ vi emp3.txt
: %s/3000/3900/g
```

문제 144. emp.txt와 emp3.txt 파일 둘간의 데이터의 차이가 있는지 확인하시오 !

```
[orcl:~]$ diff emp.txt emp3.txt
10c10
< 7902 FORD          ANALYST          7566 81/12/11      3000          0          20
---
> 7902 FORD          ANALYST          7566 81/12/11      3900          0          20
12c12
< 7788 SCOTT         ANALYST          7566 82/12/22      3000          0          20
---
> 7788 SCOTT         ANALYST          7566 82/12/22      3900          0          20
[orcl:~]$ diff --brief emp.txt emp3.txt
Files emp.txt and emp3.txt differ
```

문제 145. 짝공이 emp1.txt ~ emp100.txt 중 어떠한 것을 변경했는지 한번에 알아내시오 !

```
[orcl:~]$ vi dif.sh
#!/bin/bash

for i in {1..100}
do
    diff --brief emp.txt emp$i.txt
done

[orcl:~]$ sh dif.sh
Files emp.txt and emp3.txt differ
Files emp.txt and emp70.txt differ
```

문제 146. 짝공과 자리를 바꿔서 emp1.txt ~ emp100.txt 중 몇개를 열어서 공백을 막 넣어주고 자리로 돌아오시오 !

문제 147. 짝공이 어떠한 파일에 공백을 넣었는지 확인하시오 !

```
[orcl:~]$ vi a.sh
for i in {1..100}
do
    result=`sed -n '/^ *$/p' emp$i.txt | wc -l`
    if [ $result -ne 0 ]; then
```

```

        echo emp$i
    fi
done

[orcl:~]$ sh a.sh
emp21
emp45
emp64
emp92

```

문제 148. (점심시간 문제) 짝공이 공백을 입력한 파일들의 공백을 다 지우시오 !

```

[orcl:~]$ vi a.sh
#!/bin/bash

for i in {1..100}
do
    result=`sed -n '/^ *$/p' emp$i.txt | wc -l`
    if [ $result -ne 0 ]; then
        sed '/^ */d' emp$i.txt >> empback$i.txt
        rm emp$i.txt
        mv empback$i.txt emp$i.txt
        echo emp$i.txt "가 수정되었습니다.."
    fi
done

[orcl:~]$ sh a.sh
emp21.txt 가 수정되었습니다.
emp45.txt 가 수정되었습니다.
emp64.txt 가 수정되었습니다.
emp92.txt 가 수정되었습니다.

```

문제 149. (bc 카드 공채 문제) 아래의 쉘을 구현하시오 !

```

[orcl:~]$ vi factorial.sh
num=1
num1=$1
for i in `eval echo {1..$1}`
do
    num=`expr $num W* $i`
done
echo $num

[orcl:~]$ sh factorial.sh 5
120

```

문제 150. (알고리즘 문제) 아래의 power 함수를 쉘로 구현하시오 !

```

SQL> select power(2, 3) from dual;
      8

[orcl:~]$ vi power.sh
num=1
for i in `eval echo {1..$2}`

```

```
do
    num=`expr $num W* $1`
done
echo $num
```

```
[orcl:~]$ sh power.sh 2 3
8
```

```
※ 디버깅 하고 싶을때는 ? (-x)
[orcl:~]$ sh -x power.sh 3 4
+ num=1
++ eval echo '{1..4}'
+++ echo 1 2 3 4
+ for i in `eval echo {1..$2}`
++ expr 1 '*' 3
+ num=3
+ for i in `eval echo {1..$2}`
++ expr 3 '*' 3
+ num=9
+ for i in `eval echo {1..$2}`
++ expr 9 '*' 3
+ num=27
+ for i in `eval echo {1..$2}`
++ expr 27 '*' 3
+ num=81
+ echo 81
81
```

문제 151. (알고리즘 문제) 최대 공약수를 출력하시오 !

```
[orcl:~]$ vi gcd.sh
num1=$1
num2=$2
dummy=0
i=1
while [ $i -eq 1 ]
do
    if [ $num1 -lt $num2 ]; then
        dummy=$num1
        num1=$num2
        num2=$dummy
    fi
    dummy=`expr $num1 % $num2`
    num1=$num2
    num2=$dummy
    if [ $num2 -eq 0 ]; then
        i=2
    fi
done
echo $num1
```

```
[orcl:~]$ sh gcd.sh 16 24
8
```



문제 152. 직업을 중복제거 해서 출력하시오 !

```
[orcl:~]$ awk '{print $3}' emp.txt | sort | uniq
ANALYST
CLERK
MANAGER
PRESIDENT
SALESMAN
```

문제 153. 위의 결과를 변수에 담아서 for loop 문으로 변수의 내용을 출력하시오 !

```
[orcl:~]$ vi a.sh
job=`awk '{print $3}' emp.txt | sort | uniq`

for i in $job
do
    echo $i
done

[orcl:~]$ sh a.sh
ANALYST
CLERK
MANAGER
PRESIDENT
SALESMAN
```

문제 154. 직업이 SALESMAN 인 사원들의 토탈월급을 출력하시오 !

```
[orcl:~]$ grep -i 'salesman' emp.txt | awk '{print $6}' | awk '{sum = sum+$1} END {print sum}'
5600
```

문제 155. 문제 153번과 문제 154번 코드를 조합해서 아래의 결과를 출력하시오 !

```
[orcl:~]$ vi a.sh
job=`awk '{print $3}' emp.txt | sort | uniq`

for i in $job
do
    echo $i `grep -i $i emp.txt | awk '{print $6}' | awk '{sum = sum+$1} END {print sum}'`
done

[orcl:~]$ sh a.sh
ANALYST 6000
CLERK 4150
MANAGER 8275
PRESIDENT 5000
SALESMAN 5600
```

문제 156. 부서번호, 부서번호별 평균 월급을 출력하시오 !

```
[orcl:~]$ vi a.sh
deptno=`awk '{print $8}' emp.txt | sort | uniq`
```

```

for i in $deptno
do
    sumsal=`grep -i $i emp.txt | awk '{print $6}' | awk '{sum = sum+$1} END {print sum}'`
    cnt=`grep $i emp.txt | wc -l`
    avgsal=`expr $sumsal / $cnt`
    echo $i $avgsal
done

[orcl:~]$ sh a.sh
10 3700
20 2175
30 2783

```

문제 157. 구구단 2단을 출력하시오 ! (while loop문으로)

```

[orcl:~]$ vi a.sh
#!/bin/bash

i=1
while [ $i -le 9 ]
do
    gop=`expr $i W* 2`
    echo "2 x $i = $gop"
    i=`expr $i + 1`
done

[orcl:~]$ sh a.sh
2 x 1 = 2
2 x 2 = 4
2 x 3 = 6
2 x 4 = 8
2 x 5 = 10
2 x 6 = 12
2 x 7 = 14
2 x 8 = 16
2 x 9 = 18

```

문제 158. (while loop문의 중첩)

구구단을 2단부터 9단까지 출력하시오 !

```

[orcl:~]$ vi a.sh
#!/bin/bash

i=1
j=2
while [ $j -le 9 ]
do
    while [ $i -le 9 ]
    do
        gop=`expr $i W* $j`
        echo "$j x $i = $gop"
        i=`expr $i + 1`
    done
    echo "-----"
    i=1
    j=`expr $j + 1`
done

```

done

```
[orcl:~]$ sh a.sh
2 x 1 = 2
2 x 2 = 4
2 x 3 = 6
      :
      :
-----
      :
      :
9 x 7 = 63
9 x 8 = 72
9 x 9 = 81
```

문제 159. 아래와 같이 ★를 while loop 문으로 구현하시오 !

```
[orcl:~]$ vi star.sh
#!/bin/bash

echo -n "input number : "
read num
star=""
i=1
while [ $i -le $num ]
do
    j=1
    while [ $j -le $i ]
    do
        star="$star███~X~E"
        j=`expr $j + 1`
    done
    echo $star
    star=""
    i=`expr $i + 1`
done

[orcl:~]$ sh star.sh
input number : 5
★
★★
★★★
★★★★
★★★★★
```

문제 160. 아래의 power함수를 while loop문으로 구현하시오 !

```
SQL> select power(2, 3) from dual;
8
```

```
[orcl:~]$ vi power.sh
#!/bin/bash

num=1
i=1
while [ $i -le $2 ]
do
```

```

        num=`expr $num W* $1`
        i=`expr $i + 1`
    done
    echo $num

[orcl:~]$ sh power.sh 2 3
8

```

문제 161. while loop문으로 factorial을 구현하시오 !

```

[orcl:~]$ vi factorial.sh
#!/bin/bash

num=1
i=1
while [ $i -le $1 ]
do
    num=`expr $num W* $i`
    i=`expr $i + 1`
done
echo $num

[orcl:~]$ sh factorial.sh 5
120

```

문제 162. log 함수를 while loop문으로 구현하시오 !

```

[orcl:~]$ vi log.sh
#!/bin/bash

log_val=1
i=$1
while [ $i -lt $2 ]
do
    i=`expr $i W* $1`
    log_val=`expr $log_val + 1`
done
echo $log_val

[orcl:~]$ sh log.sh 2 16
4

```

문제 163. (점심시간 문제) 최대공약수를 while loop문으로 구현하시오 !

```

[orcl:~]$ vi gcd.sh
num1=$1
num2=$2
dummy=0
i=1
while [ $i -eq 1 ]
do
    if [ $num1 -lt $num2 ]; then
        dummy=$num1
        num1=$num2
        num2=$dummy
    fi
    i=`expr $i + 1`
done
echo $dummy

```

```

fi

dummy=`expr $num1 % $num2`
num1=$num2
num2=$dummy
if [ $num2 -eq 0 ]; then
    i=2
fi
done

echo $num1

```

```

[orcl:~]$ sh gcd.sh 24 16
8
[orcl:~]$ sh gcd.sh 52 16
4

```

문제 164. 위의 스크립트를 수정해서 파일명을 물어보게 하고 파일이 존재하면 파일의 내용이 출력되게 하고 파일이 없으면 "파일이 없습니다." 라는 메시지가 출력되게 하시오 !

```

[orcl:~]$ vi a.sh
#!/bin/bash

echo "보고싶은 파일명을 입력하세요 ~ "
read file_name

fname=/home/oracle/$file_name
if [ -e $fname ]; then
    cat $fname
else
    echo "파일이 없습니다."
fi

[orcl:~]$ sh a.sh
input file name :
dept.txt
10 ACCOUNTING      NEWYORK
20 RESEARCH        DALLAS
30 SALES            CHICAGO
40 OPERATIONS      BOSTON

```

문제 165. emp.txt 에서 직업이 salesman인 사원들의 data만 가지고 salesman.txt 라는 파일을 생성하시오 !

```

[orcl:~]$ grep -i 'salesman' emp.txt >> salesman.txt
[orcl:~]$ cat salesman.txt
7654 MARTIN          SALESMAN          7698 81/09/10      1250      1400      30
7499 ALLEN           SALESMAN          7698 81/02/11      1600      300      30
7844 TURNER          SALESMAN          7698 81/08/21      1500      0        30
7521 WARD            SALESMAN          7698 81/02/23      1250      500      30

```

문제 166. 위의 스크립트를 가지고 shell을 생성하는데 직업을 물어보게하고 직업을 입력하면 해당 직업의 data를 가지고 txt파일이 만들어지게 하시오 !

★라이나 생명에서 데이터 분석을 했던 데이터 분석 내용  
1. 하둡, 오라클, 리눅스 쉘에서 데이터를 필터링을 하고

- 예 : 라이나 생명 고객중 40대만 따로 분리해서 text파일을 생성  
 2. 40대의 데이터를 가지고 군집분석(k-means) : R, 파이썬  
 3. 40대의 데이터를 가지고 연관분석(아프리오리 알고리즘) : R, 파이썬

```
[orcl:~]$ vi make_job.sh
#!/bin/bash
```

```
echo "직업을 입력하세요 ~ "
read job
```

```
grep -i $job emp.txt >> $job.txt
cat $job.txt
```

```
[orcl:~]$ sh make_job.sh
7902 FORD ANALYST 7566 81/12/11 3000 0 20
7788 SCOTT ANALYST 7566 82/12/22 3000 0 20
```

문제 167. 문제 166번 코드를 수정해서 이미 생성된 직업.txt 가 존재한다면 해당 파일은 이미 있습니다.라는 메시지가 출력되게 하고 없으면 직업.txt 가 생성되게 하시오 !

```
[orcl:~]$ vi make_job.sh
#!/bin/bash

echo "직업을 입력하세요 ~ "
read job

if [ -e $job.txt ]; then
    echo "해당 직업의 파일은 이미 있습니다."
else
    grep -i $job emp.txt >> $job.txt
    cat $job.txt
fi
```

```
[orcl:~]$ sh make_job.sh
직업을 입력하세요 ~
analyst
해당 직업의 파일은 이미 있습니다.
```

문제 168. 위의 문제처럼 직업을 일일이 물어보지 않아도 그냥 알아서 모든 직업에 대한 파일이 아래와 같이 생성 되게끔 쉘 스크립트를 작성하시오 !

```
[orcl:~]$ vi make_job.sh
#!/bin/bash

for i in `awk '{print $3}' emp.txt | sort | uniq`
do
    if [ -e $i.txt ]; then
        echo "해당직업의 파일은 이미 있습니다."
    else
        grep -i $i emp.txt >> $i.txt
        cat $i.txt
        echo "$i 파일이 생성되었습니다."
    fi
done
```

```
[orcl:~]$ sh make_job.sh
7902 FORD ANALYST 7566 81/12/11 3000 0 20
7788 SCOTT ANALYST 7566 82/12/22 3000 0 20
ANALYST 파일이 생성 되었습니다.
7900 JAMES CLERK 7698 81/12/11 950 0 30
7369 SMITH CLERK 7902 80/12/09 800 0 20
7876 ADAMS CLERK 7788 83/01/15 1100 0 20
7934 MILLER CLERK 7782 82/01/11 1300 0 10
CLERK 파일이 생성 되었습니다.
7698 BLAKE MANAGER 7839 81/05/01 2850 0 30
7782 CLARK MANAGER 7839 81/05/09 2450 0 10
7566 JONES MANAGER 7839 81/04/01 2975 0 20
MANAGER 파일이 생성 되었습니다.
7839 KING PRESIDENT 0 81/11/17 5000 0 10
PRESIDENT 파일이 생성 되었습니다.
7654 MARTIN SALESMAN 7698 81/09/10 1250 1400 30
7499 ALLEN SALESMAN 7698 81/02/11 1600 300 30
7844 TURNER SALESMAN 7698 81/08/21 1500 0 30
7521 WARD SALESMAN 7698 81/02/23 1250 500 30
SALESMAN 파일이 생성 되었습니다.
```

문제 169. 위의 스크립트를 활용해서 부서번호별로 emp.txt가 쪼개지게끔 스크립트를 작성하시오 !

```
[orcl:~]$ vi make_deptno.sh
#!/bin/bash

for i in `awk '{print $8}' emp.txt | sort | uniq`
do
    if [ -e $i.txt ]; then
        echo "해당 부서번호의 파일은 이미 있습니다."
    else
        awk '$8==$i emp.txt >> $i.txt
        cat $i.txt
        echo "$i 파일이 생성되었습니다."
    fi
done
```

```
[orcl:~]$ sh make_deptno.sh
7839 KING PRESIDENT 0 81/11/17 5000 0 10
7782 CLARK MANAGER 7839 81/05/09 2450 0 10
7934 MILLER CLERK 7782 82/01/11 1300 0 10
10 파일이 생성되었습니다.
7566 JONES MANAGER 7839 81/04/01 2975 0 20
7902 FORD ANALYST 7566 81/12/11 3000 0 20
7369 SMITH CLERK 7902 80/12/09 800 0 20
7788 SCOTT ANALYST 7566 82/12/22 3000 0 20
7876 ADAMS CLERK 7788 83/01/15 1100 0 20
20 파일이 생성되었습니다.
7698 BLAKE MANAGER 7839 81/05/01 2850 0 30
7654 MARTIN SALESMAN 7698 81/09/10 1250 1400 30
7499 ALLEN SALESMAN 7698 81/02/11 1600 300 30
7844 TURNER SALESMAN 7698 81/08/21 1500 0 30
7900 JAMES CLERK 7698 81/12/11 950 0 30
7521 WARD SALESMAN 7698 81/02/23 1250 500 30
30 파일이 생성되었습니다.
```

문제 170. 직업과 직업별 토달월급을 출력하는데 직업별 토달월급이 큰 것부터 출력하시오 !

```
SQL> select job, sum(sal)
2   from emp
3   group by job
4   order by sum(sal) desc;
```

JOB	SUM(SAL)
MANAGER	8275
ANALYST	6000
SALESMAN	5600
PRESIDENT	5000
CLERK	4150

※ 만약 쿼리가 틀려서 수정해야 한다면 ?  
ed 명령어로 수정해야 하는데 리눅스에서는 vi 편집기가 열리도록 설정해야 한다.

문제 171. 테이블명을 물어보게하고 타이틀명을 입력하면 해당 테이블의 데이터를 csv파일로 생성하는 쉘스크립트를 작성하시오 !

```
[orcl:~]$ vi make_csv.sh
#!/bin/bash
echo "★★★★★★★★★★★★"
echo -n "테이블명을 입력하세요 : "
read table
echo "★★★★★★★★★★★★"
echo " "
sqlplus scott/tiger << EOF
set colsep ','
set pagesize 20
set linesize 120
set sqlprompt ""
set feedback off
set echo off
set underline off

spool $table.csv

select * from $table;

spool off

EOF
```

문제 172. dept.csv에서 1번과 2번 라인을 삭제하고 dept2.csv로 저장되게 하시오 !

```
[orcl:~]$ sed '1,2d' dept.csv >> dept2.csv
[orcl:~]$ sed '/spool/d' dept2.csv >> dept3.csv
[orcl:~]$ cat dept3.csv
```

DEPTNO,DNAME	,LOC
10,ACCOUNTING	,NEW YORK
20,RESEARCH	,DALLAS
30,SALES	,CHICAGO
40,OPERATIONS	,BOSTON



문제 173. dept.csv를 지우고 dept3.csv를 dept.csv로 이름을 변경하시오 !

```
[orcl:~]$ rm dept.csv
[orcl:~]$ mv dept3.csv dept.csv
[orcl:~]$ cat dept.csv
```

```
DEPTNO,DNAME      ,LOC
    10,ACCOUNTING  ,NEW YORK
    20,RESEARCH    ,DALLAS
    30,SALES       ,CHICAGO
    40,OPERATIONS  ,BOSTON
```

문제 174. 아래의 스크립트를 clean.sh 로 저장하시오 !

```
#!/bin/bash

sed '1,2d' /home/oracle/$1 > /home/oracle/$12
sed '/spool/d' /home/oracle/$12 > /home/oracle/$13
sed '/^$/d' /home/oracle/$13 > /home/oracle/$14
rm /home/oracle/$1
mv /home/oracle/$14 /home/oracle/$1
```

문제 175. make\_csv.sh를 이용해서 emp.csv를 생성하시오 !

```
[orcl:~]$ sh make_csv.sh
★★★★★★★★★★★★
테이블명을 입력하세요 : emp
★★★★★★★★★★★★
[orcl:~]$ sh clean.sh emp.csv
```

문제 176. `make_csv.sh` 에 `clean.sh` 스크립트를 수행하는 스크립트 맨 아래에 입력해서 스크립트를 합치시오 !

[illegible]

```
spool off
```

```
EOF
```

```
sh clean.sh $table.csv
```

문제 177. (오늘의 마지막 문제) hr 계정이 가지고 있는 모든 테이블들을 전부 csv 로 생성하시오 !

```
[orcl:~]$ sqlplus / as sysdba
```

```
SQL> alter user hr  
2 account unlock;
```

```
User altered.
```

```
SQL> alter user hr  
2 identified by hr;
```

```
User altered.
```

```
SQL> connect hr/hr  
Connected.
```

```
SQL> select table_name from user_tables;
```

```
TABLE_NAME  
-----  
REGIONS  
DEPARTMENTS  
COUNTRIES  
EMPLOYEES  
LOCATIONS  
JOB_HISTORY  
JOBS
```

문제 178. 문제 144번 코드를 가지고 아래의 쉘 스크립트를 작성하시오 !

```
[orcl:~]$ vi diff.sh  
#!/bin/bash
```

```
echo "  
echo -n "비교할 첫번째 파일명을 입력하세요 ~"  
read file1  
echo -n "비교할 두번째 파일명을 입력하세요 ~"  
read file2  
echo "  
diff $file1 $file2  
diff --brief $file1 $file2  
[orcl:~]$ sh diff.sh
```

```
비교할 첫번째 파일명을 입력하세요 ~emp.txt  
비교할 두번째 파일명을 입력하세요 ~emp3.txt
```

```
1c1
```

```
< 7839 KING
```

```
PRESIDENT
```

```
0 81/11/17
```

```
5000
```

```
0
```

```
10
```

```
---
```

문제 76. (점심시간 문제) 파일을 검색하는 쉘 스크립트를 아래와 같이 생성하시오 !

```
[orcl:~]$ vi find_file.sh
```

```
#!/bin/bash
```

```
echo " "
echo -n "현재 디렉토리 밑에 검색할 파일명을 입력하세요 ! "
read file
echo -n "depth는 어떻게 할까요? "
read depth
echo " "
dir=`pwd`
find $dir -maxdepth $depth -name $file -print
```

```
[orcl:~]$ sh find_file.sh
```

```
현재 디렉토리 밑에 검색할 파일명을 입력하세요 ! emp.txt
depth는 어떻게 할까요? 1
```

```
./emp.txt
```

문제 179. 아래의 두개의 스크립트를 이용해서 case문으로 아래의 두개의 스크립트를 쉽게 실행할 수 있도록 total.sh 스크립트를 생성하시오 !

1. 두 파일의 차이를 확인하는 diff.sh
2. 파일을 검색하는 find\_file.sh

```
[orcl:~]$ chmod ugo+rx diff.sh
[orcl:~]$ chmod ugo+rx find_file.sh
[orcl:~]$ vi total.sh
#!/bin/bash
```

```
echo " "
echo "1. 두파일의 차이를 확인하려면 1번을
      2. 특정 파일을 검색하려면 2번을
"
```

```
echo -n "번호를 입력하세요 ~"
read choice
case $choice in
1)
    /home/oracle/diff.sh ;;
2)
    /home/oracle/find_file.sh ;;
esac
```

문제 180. 겨울왕국 스크립트에서 elsa를 찾는 쉘 스크립트를 가져와서 아래의 스크립트로 완성하시오 !

```
[orcl:~]$ vi find_word.sh
#!/bin/bash
```

```

echo -n "스크립트 이름을 입력하세요 !"
read file
echo -n "스크립트에서 찾을 단어를 입력하세요 !"
read word
grep -iow $word $file | wc -l

[orcl:~]$ sh find_word.sh
스크립트 이름을 입력하세요 !winter.txt
스크립트에서 찾은 단어를 입력하세요 !elsa
658

```

문제 181. 위에서 만든 find\_word.sh를 total.sh에 3번으로 추가시키시오!

```

[orcl:~]$ chmod ugo+rx find_word.sh
[orcl:~]$ vi total.sh
#!/bin/bash

echo "
"
echo "1. 두파일의 차이를 확인하려면 1번을
      2. 특정 파일을 검색하려면 2번을
      3. 찾고자 하는 단어의 갯수를 검색하려면 3번을
"
echo -n "번호를 입력하세요 ~"
read choice
case $choice in
    1)
        /home/oracle/diff.sh ;;
    2)
        /home/oracle/find_file.sh ;;
    3)
        /home/oracle/find_word.sh ;;
esac

```

문제 182. scott으로 접속해서 csv파일을 생성하는 쉘 스크립트를 아래의 이름으로 생성하시오

```

$ sh make_scott_csv.sh

테이블명을 입력하세요~ emp

emp.csv가 생성되게 하시오

[orcl:~]$ vi make_scott_csv.sh
#!/bin/bash
echo "★★★★★★★★★★★★"
echo -n "테이블명을 입력하세요 : "
read table
echo "★★★★★★★★★★★★"
echo " "
sqlplus hr/hr << EOF
set colsep ' '
set pagesize 2000
set linesize 200
set sqlprompt ""
set feedback off
set echo off
set underline off

```

```

spool $table.csv

select * from $table;

spool off

EOF

sh clean.sh $table.csv

```

문제 183. scott과 hr로 자동으로 접속해서 csv를 만들 수 있도록 위의 스크립트를 이용해서 아래의 스크립트를 생성하시오 !

```

[orcl:~]$ vi make_csv.sh
#!/bin/bash
echo -n "유저명을 입력하세요 ~"
read user
echo -n "패스워드를 입력하세요 ~"
read pass
echo "★★★★★★★★★★★★"
echo -n "테이블명을 입력하세요 : "
read table
echo "★★★★★★★★★★★★"
echo " "
sqlplus $user/$pass << EOF
set colsep ','
set pagesize 2000
set linesize 200
set sqlprompt ""
set feedback off
set echo off
set underline off

spool $table.csv

select * from $table;

spool off

EOF

sh clean.sh $table.csv

```

문제 184. make\_csv.sh 를 total.sh 의 4번에 추가 시키시오 !

1. 두파일의 차이를 확인하려면 1번을
2. 특정 파일을 검색하려면 2번을
3. 찾고자 하는 단어의 갯수를 검색하려면 3번을
4. 오라클 데이터를 csv로 생성하려면 4번을

```

[orcl:~]$ chmod ugo+rx make_csv.sh
[orcl:~]$ vi total.sh
#!/bin/bash

echo " "
echo "1. 두파일의 차이를 확인하려면 1번을

```

```

2. 특정 파일을 검색하려면 2번을
3. 찾고자하는 단어의 갯수를 검색하려면 3번을
4. 오라클 데이터를 csv로 생성하려면 4번을
"

echo -n "번호를 입력하세요 ~"
read choice
case $choice in
    1)
        /home/oracle/diff.sh ;;
    2)
        /home/oracle/find_file.sh ;;
    3)
        /home/oracle/find_word.sh ;;
    4)
        /home/oracle/make_csv.sh ;;
esac

```

문제 185. 위에서 생성한 emp.csv에서 이름이 scott인 사원의 이름과 월급을 출력하는 python코드를 작성하시오 !

```

[orcl:~]$ vi find_ename.py
import csv

a = raw_input("input ename : ").upper()
file = open("/home/oracle/emp.csv")
emp_csv = csv.reader(file)
for i in emp_csv:
    if a == i[1].strip():
        print i[1], i[5]

[orcl:~]$ python find_ename.py
input ename : scott
SCOTT          3000

```

문제 186. (점심시간 문제) 문제 185번 아래의 파이썬 실행문장을 total.sh의 5번에 추가하시오 !

```

[orcl:~]$ chmod ugo+rxw find_ename.py
[orcl:~]$ vi total.sh
#!/bin/bash

echo "
echo "1. 두파일의 차이를 확인하려면 1번을
2. 특정 파일을 검색하려면 2번을
3. 찾고자하는 단어의 갯수를 검색하려면 3번을
4. 오라클 데이터를 csv로 생성하려면 4번을
5. emp.csv에서 이름과 월급을 검색하려면 5번을
"

echo -n "번호를 입력하세요 ~"
read choice
case $choice in
    1)
        /home/oracle/diff.sh ;;
    2)
        /home/oracle/find_file.sh ;;
    3)
        /home/oracle/find_word.sh ;;
    4)

```

```

/home/oracle/make_csv.sh ;;
5)
python /home/oracle/find_ename.py ;;
esac

[orcl:~]$ sh total.sh

1. 두파일의 차이를 확인하려면 1번을
2. 특정 파일을 검색하려면 2번을
3. 찾고자하는 단어의 갯수를 검색하려면 3번을
4. 오라클 데이터를 csv로 생성하려면 4번을
5. emp.csv에서 이름과 월급을 검색하려면 5번을

번호를 입력하세요 ~5
input ename : scott
SCOTT          3000

```

문제 187. oracle2 라는 유저를 생성하시오 !

```

[root@edydr1p0 ~]# useradd oracle2

[root@edydr1p0 ~]# cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
news:x:9:13:news:/etc/news:
uucp:x:10:14:uucp:/var/spool/uucp:/sbin/nologin
operator:x:11:0:operator:/root:/sbin/nologin
games:x:12:100:games:/usr/games:/sbin/nologin
gopher:x:13:30:gopher:/var/gopher:/sbin/nologin
ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin
nobody:x:99:99:Nobody:/:/sbin/nologin
nscd:x:28:28:NSCD Daemon:/:/sbin/nologin
vcsa:x:69:69:virtual console memory owner:/dev:/sbin/nologin
rpc:x:32:32:Portmapper RPC user:/:/sbin/nologin
mailnull:x:47:47:/:/var/spool/mqueue:/sbin/nologin
smmsp:x:51:51:/:/var/spool/mqueue:/sbin/nologin
oprofile:x:16:16:Special user account to be used by OProfile:/home/oprofile:/sbin/nologin
pcap:x:77:77:/:/var/arpwatch:/sbin/nologin
ntp:x:38:38:/:etc/ntp:/sbin/nologin
dbus:x:81:81:System message bus:/:/sbin/nologin
avahi:x:70:70:Avahi daemon:/:/sbin/nologin
sshd:x:74:74:Privilege-separated SSH:/var/empty/sshd:/sbin/nologin
rpcuser:x:29:29:RPC Service User:/var/lib/nfs:/sbin/nologin
nfsnobody:x:65534:65534:Anonymous NFS User:/var/lib/nfs:/sbin/nologin
haldaemon:x:68:68:HAL daemon:/:/sbin/nologin
avahi-autoipd:x:100:103:avahi-autoipd:/var/lib/avahi-autoipd:/sbin/nologin
xfs:x:43:43:X Font Server:/etc/X11/fs:/sbin/nologin
gdm:x:42:42:/:/var/gdm:/sbin/nologin
sabayon:x:86:86:Sabayon user:/home/sabayon:/sbin/nologin
vboxadd:x:101:1:/:/var/run/vboxadd:/bin/false
oracle:x:500:504:/:home/oracle:/bin/bash

```

```

grid:x:501:504:./home/grid:/bin/bash
oracle2:x:502:510:./home/oracle2:/bin/bash
oracle2:x: 502: 510: ./home/oracle2:/bin/bash
  ↑      ↑      ↑      ↑      ↑      ↑
  사용자 패스워드 유저id 그룹id 홈디렉토리 로그인 기본셸

```

문제 188. oracle2 접속한 그 홈디렉토리에서 aaa라는 폴더를 만드시오 !

```

[oracle2@edydr1p0 ~]$ mkdir aaa
[oracle2@edydr1p0 ~]$ ls
aaa

```

문제 189. 리눅스 시스템의 사용자와 그 사용자들의 패스워드를 조회하시오 !

```

[oracle2@edydr1p0 ~]$ su -
Password: oracle
[root@edydr1p0 ~]# cat /etc/shadow
root:$1$4F01uWcs$YZQbV6x2xd0ulzCWdYvCx.:15606:0:99999:7:::
bin:!:15606:0:99999:7:::
daemon:!:15606:0:99999:7:::
adm:!:15606:0:99999:7:::
lp:!:15606:0:99999:7:::
sync:!:15606:0:99999:7:::
shutdown:!:15606:0:99999:7:::
halt:!:15606:0:99999:7:::
mail:!:15606:0:99999:7:::
news:!:15606:0:99999:7:::
uucp:!:15606:0:99999:7:::
operator:!:15606:0:99999:7:::
games:!:15606:0:99999:7:::
gopher:!:15606:0:99999:7:::
ftp:!:15606:0:99999:7:::
nobody:!:15606:0:99999:7:::
nscd:!:15606:0:99999:7:::
vcsa:!:15606:0:99999:7:::
rpc:!:15606:0:99999:7:::
mailnull:!:15606:0:99999:7:::
smmsp:!:15606:0:99999:7:::
oprofile:!:15606:0:99999:7:::
pcap:!:15606:0:99999:7:::
ntp:!:15606:0:99999:7:::
dbus:!:15606:0:99999:7:::
avahi:!:15606:0:99999:7:::
sshd:!:15606:0:99999:7:::
rpcuser:!:15606:0:99999:7:::
nfsnobody:!:15606:0:99999:7:::
haldaemon:!:15606:0:99999:7:::
avahi-autoipd:!:15606:0:99999:7:::
xfs:!:15606:0:99999:7:::
gdm:!:15606:0:99999:7:::
sabayon:!:15606:0:99999:7:::
vboxadd:!:15606:0:99999:7:::
oracle:$1$vo/QDy68$Jdjx8NAYgCM6hbMGW/TPv1:15606:0:99999:7:::
grid:$1$n.P.1Cs1$egWWHThIy3Zo00DJSddG0/:15606:0:99999:7:::
oracle2:$1$Fbpy10VB$.nfamRCE2JDZ4eIbws0V1.:17891:0:99999:7:::

```



문제 190. oracle3이라는 유저를 생성하고 oracle3으로 접속을하면 자동으로 /home/oracle3밑에 emp.txt  
와 dept.txt 와 winter.txt가 생성되어있게 하시오 !

```
[root@edydr1p0 ~]# cat /etc/default/useradd
# useradd defaults file
GROUP=100
HOME=/home
INACTIVE=-1
EXPIRE=
SHELL=/bin/bash
SKEL=/etc/skel      <---- 이 위치에 특정 파일을 가져다 놓으면 유저가 생성될 때마다 자동으로 그
CREATE_MAIL_SPOOL=yes      유저의 홈디렉토리에 특정파일이 생성이 된다.
```

```
[root@edydr1p0 ~]# cp /home/oracle/*.txt /etc/skel/
[root@edydr1p0 ~]# useradd oracle3
[root@edydr1p0 ~]# passwd oracle3
Changing password for user oracle3.
New UNIX password: oracle
BAD PASSWORD: it is based on a dictionary word
Retype new UNIX password: oracle
passwd: all authentication tokens updated successfully.
[root@edydr1p0 ~]# su - oracle3
[oracle3@edydr1p0 ~]$ ls
10.txt      emp16.txt  emp29.txt  emp41.txt  emp53.txt  emp66.txt  emp79.txt  emp91.txt
20.txt      emp17.txt  emp2.txt   emp42.txt  emp54.txt  emp67.txt  emp7.txt   emp92.txt
30.txt      emp18.txt  emp30.txt  emp43.txt  emp55.txt  emp68.txt  emp80.txt  emp93.txt
ANALYST.txt emp19.txt  emp31.txt  emp44.txt  emp56.txt  emp69.txt  emp81.txt  emp94.txt
b.txt       emp1.txt   emp32.txt  emp45.txt  emp57.txt  emp6.txt   emp82.txt  emp95.txt
CLERK.txt   emp20.txt  emp33.txt  emp46.txt  emp58.txt  emp70.txt  emp83.txt  emp96.txt
dept.txt    emp21.txt  emp34.txt  emp47.txt  emp59.txt  emp71.txt  emp84.txt  emp97.txt
emp100.txt  emp22.txt  emp35.txt  emp48.txt  emp5.txt   emp72.txt  emp85.txt  emp98.txt
emp10.txt   emp23.txt  emp36.txt  emp49.txt  emp60.txt  emp73.txt  emp86.txt  emp99.txt
emp11.txt   emp24.txt  emp37.txt  emp4.txt   emp61.txt  emp74.txt  emp87.txt  emp9.txt
emp12.txt   emp25.txt  emp38.txt  emp500.txt emp62.txt  emp75.txt  emp88.txt  emp.txt
emp13.txt   emp26.txt  emp39.txt  emp50.txt  emp63.txt  emp76.txt  emp89.txt  hhh2.txt
emp14.txt   emp27.txt  emp3.txt   emp51.txt  emp64.txt  emp77.txt  emp8.txt   hhh3.txt
emp15.txt   emp28.txt  emp40.txt  emp52.txt  emp65.txt  emp78.txt  emp90.txt  hhh.txt
```

문제 191. oracle7이라는 유저를 생성하시오 !

```
[root@edydr1p0 ~]# useradd oracle7

[root@edydr1p0 ~]# passwd oracle7
Changing password for user oracle7.
New UNIX password: oracle
BAD PASSWORD: it is based on a dictionary word
Retype new UNIX password: oracle
passwd: all authentication tokens updated successfully.

[root@edydr1p0 ~]# su - oracle7
[oracle7@edydr1p0 ~]$ pwd
/home/oracle7
```

문제 192. oracle7의 홈디렉토리를 /home/oracle777으로 변경하시오 !

```
[oracle7@edydr1p0 ~]$ su -
```

```

Password: oracle
[root@edydr1p0 ~]# mkdir -p /home/oracle777
[root@edydr1p0 ~]# chown -R oracle7:oinstall /home/oracle777    <- 소유자 변경
[root@edydr1p0 ~]# cat /etc/passwd
:
:
oracle7:x:504:512::/home/oracle7:/bin/bash

[root@edydr1p0 ~]# usermod -d /home/oracle777 oracle7

[root@edydr1p0 ~]# su - oracle7
-bash-3.2$ pwd
/home/oracle777

```

문제 193. oracle7 유저도 oracle2유저처럼 접속했을 때에 리눅스 화면이 아래와 같게 하시오 !

```

[root@edydr1p0 ~]# cp /home/oracle2/.bash_profile /home/oracle777/    <--- 유저의 환경설정 파일
[root@edydr1p0 ~]# cp /home/oracle2/.bashrc /home/oracle777/
[root@edydr1p0 ~]# su - oracle7
[oracle7@edydr1p0 ~]$

```

문제 194. oracle7으로 접속할때 마다 아래의 메세지가 출력되게 하시오 !

저한테 왜 그러셨어요

```

[oracle7@edydr1p0 ~]$ vi .bash_profile
# .bash_profile

# Get the aliases and functions
if [ -f ~/.bashrc ]; then
    . ~/.bashrc
fi

# User specific environment and startup programs

PATH=$PATH:$HOME/bin

export PATH

echo "저한테 왜그러셨어요"

```

문제 195. /home/oracle2 디렉토리를 삭제하시오 !

```

[root@edydr1p0 ~]# rm -rf /home/oracle2

```

문제 196. oracle7 유저를 삭제하고 oracle7의 홈 디렉토리도 삭제하시오 !(/home/oracle777)

```

[root@edydr1p0 home]# userdel oracle7
[root@edydr1p0 home]# rm -rf /home/oracle777
[root@edydr1p0 home]# rm -rf /home/oracle7

```

문제 197. oracle9를 oinstall그룹에서 제거하시오 !

```
[root@edydr1p0 ~]# gpasswd -d oracle9 oinstall
Removing user oracle9 from group oinstall
[root@edydr1p0 ~]# cat /etc/group
:
:
oinstall:x:504:
```