컴퓨터시스템일반 4단원 심화 문제

운영체제(OS), 리눅스, 셸 명령어, 파일 시스템, 사용자와 그룹, 패키지 관리, 시스템 관리

1. [3점] 운영체제, 커널, 셸의 관계에 대한 설명으로 옳은 것만을 고른 것은?

ㄱ. 커널은 CPU, 메모리, 저장 장치 같은 자원을 관리하고 하드웨어와 직접 맞닿아 동작한다.
ㄴ. 셸은 사용자의 명령을 해석해 커널에 전달하는 역할을 하므로, 하드웨어 자원 배분 자체를 담당한다고 볼 수 있다.
ㄷ. CLI는 명령을 알아야 한다는 부담이 있지만 반복 작업과 원격 관리에서 효율적으로 쓰일 수 있다.
ㄹ. GUI는 셸이 아니므로 운영체제와 사용자를 연결하는 인터페이스의 한 종류로 볼 수 없다.

① ㄱ, ㄴ
② ㄱ, ㄷ
③ ㄴ, ㄷ
④ ㄴ, ㄹ
⑤ ㄱ, ㄷ, ㄹ

2. [3점] 다음 설명 중 리눅스와 윈도우의 차이를 가장 정확하게 이해한 것은?

① 리눅스는 무료로만 배포되므로 서버에서 성능이나 안정성의 장점을 갖는다고 볼 수 없다.
② 윈도우와 달리 리눅스는 단일 루트 디렉터리 `/` 아래에 파일 시스템이 구성되는 방식이 일반적이다.
③ 리눅스의 장점은 GUI 프로그램 호환성뿐이므로 CLI 학습은 서버 관리와 직접 관련이 없다.
④ 리눅스는 오픈 소스이므로 사용자 권한, 파일 권한, 패키지 관리 같은 관리 체계가 필요 없다.
⑤ 리눅스 배포판은 모두 같은 셸만 사용하므로 bash, zsh, fish의 구분은 운영체제 학습과 무관하다.

3. [3점] 다음 중 셸과 배포판의 연결 및 특징에 대한 판단으로 옳은 것만을 고른 것은?

ㄱ. bash는 리눅스에서 널리 쓰이는 대표적인 셸이다.
ㄴ. zsh는 macOS에서 기본 셸로 쓰이는 경우가 있으며, bash와 전혀 다른 운영체제 커널이다.
ㄷ. fish는 초보자가 쓰기 편한 기능을 강조하는 셸로 소개될 수 있다.
ㄹ. Red Hat, Ubuntu, Debian은 셸의 종류가 아니라 리눅스 배포판으로 분류된다.

① ㄱ, ㄴ
② ㄱ, ㄷ
③ ㄴ, ㄹ
④ ㄱ, ㄷ, ㄹ
⑤ ㄴ, ㄷ, ㄹ

4. [3점] Docker 컨테이너와 가상 머신(VM)을 비교한 설명으로 옳은 것만을 고른 것은?

ㄱ. 컨테이너는 일반적으로 호스트의 커널을 공유하므로 VM보다 가볍고 빠르게 실행될 수 있다.
ㄴ. VM은 게스트 운영체제를 포함해 가상화하므로 컨테이너보다 격리 수준은 높지만 무거워질 수 있다.
ㄷ. 컨테이너는 호스트 커널을 공유하므로 보안 격리가 VM보다 항상 강하다고 단정할 수 있다.
ㄹ. 컨테이너 안에서 만든 데이터는 별도 저장소를 연결하지 않으면 컨테이너 삭제 시 사라질 수 있다.

① ㄱ, ㄴ
② ㄱ, ㄷ
③ ㄷ, ㄹ
④ ㄱ, ㄴ, ㄹ
⑤ ㄴ, ㄷ, ㄹ

5. [3점] 이미 내려받은 `ubuntu` 이미지를 이용해 새 컨테이너를 만들고, 바로 대화형 터미널로 접속하려 한다. 가장 적절한 명령은?

① `docker pull --name lab ubuntu`
② `docker images -it --name lab ubuntu`
③ `docker run -it --name lab ubuntu`
④ `docker exec -it --name lab ubuntu`
⑤ `docker ps -a --name lab ubuntu`

6. [3점] 다음 명령을 실행했다.

`docker run --name clock -d busybox sh -c "while true; do date; sleep 1; done"`

이후 이 컨테이너의 시간 출력이 계속 이어지는지 확인하려고 할 때 가장 적절한 명령은?

① `docker logs -f clock`
② `docker images -f clock`
③ `docker run -f clock`
④ `docker rmi -f clock`
⑤ `docker exec -f clock`

7. [3점] 컨테이너 `web1`은 중지되어 있고, 이미지는 삭제하지 않은 상태이다. 기존 컨테이너를 다시 실행한 뒤 내부 셸에 접속하는 흐름으로 가장 적절한 것은?

① `docker rmi web1` 다음 `docker exec -it web1 /bin/bash`
② `docker start web1` 다음 `docker exec -it web1 /bin/bash`
③ `docker rm web1` 다음 `docker start web1`
④ `docker pull web1` 다음 `docker logs web1`
⑤ `docker images web1` 다음 `docker stop web1`

8. [3점] Docker 명령의 대상이 바르게 연결된 것만을 고른 것은?

ㄱ. `docker rm` - 컨테이너 삭제
ㄴ. `docker rmi` - 이미지 삭제
ㄷ. `docker ps -a` - 실행 중인 컨테이너만 표시
ㄹ. `docker images` - 내려받았거나 생성된 이미지 목록 표시

① ㄱ, ㄴ
② ㄴ, ㄷ
③ ㄱ, ㄴ, ㄹ
④ ㄱ, ㄷ, ㄹ
⑤ ㄴ, ㄷ, ㄹ

9. [3점] 프롬프트가 `student@cs01:~/project$`일 때 해석으로 옳은 것만을 고른 것은?

ㄱ. 현재 사용자는 `student`이다.
ㄴ. `cs01`은 현재 작업 디렉터리 이름이다.
ㄷ. `~/project`는 사용자의 홈 디렉터리 아래 `project`에 있음을 뜻한다.
ㄹ. `$`는 일반 사용자 권한의 셸임을 나타내며, `#`였다면 보통 관리자 권한을 의미한다.

① ㄱ, ㄴ
② ㄱ, ㄷ
③ ㄴ, ㄹ
④ ㄱ, ㄷ, ㄹ
⑤ ㄴ, ㄷ, ㄹ

10. [3점] 셸 명령의 일반 형식 `명령어 [옵션] [인자]`를 기준으로 볼 때, 설명이 가장 정확한 것은?

① `ls -al /etc`에서 `-al`은 인자이고 `/etc`는 옵션이다.
② `man ls`에서 `man`은 옵션이고 `ls`는 명령어이다.
③ `grep -i error app.log`에서 `-i`는 옵션이고 `error`, `app.log`는 인자로 볼 수 있다.
④ `cd ..`에서 `..`는 상위 디렉터리로 가는 옵션이다.
⑤ `date +%Y-%m-%d`는 옵션이 없으므로 출력 형식을 지정할 수 없다.

11. [3점] 현재 디렉터리는 `/home/student/project/src`이고, 직전 작업 디렉터리는 `/etc`였다고 하자. 아래 명령을 차례로 실행한 뒤 현재 디렉터리로 옳은 것은?

`cd ..`
`cd -`

① `/etc`

② `/home/student/project`
③ `/home/student/project/src`
④ `/home/student`
⑤ `/`

12. [3점] 파일과 디렉터리 명령에 대한 설명으로 옳은 것만을 고른 것은?

ㄱ. `touch memo.txt`는 파일이 없으면 빈 파일을 만들 수 있다.
ㄴ. `rmdir logs`는 `logs` 안에 파일이 있어도 디렉터리만 남기고 내용만 삭제한다.
ㄷ. `mv a.txt b.txt`는 같은 디렉터리 안에서 파일 이름 변경에 사용될 수 있다.
ㄹ. `rm -r old`는 디렉터리와 그 내부 항목을 재귀적으로 삭제할 수 있으므로 주의해야 한다.
① ㄱ, ㄴ
② ㄱ, ㄷ
③ ㄴ, ㄹ
④ ㄱ, ㄷ, ㄹ
⑤ ㄴ, ㄷ, ㄹ

13. [3점] Vim 사용에 대한 설명으로 옳은 것만을 고른 것은?

ㄱ. 일반 모드에서 `i`, `a`, `o`를 누르면 각각 다른 방식으로 입력 모드에 들어갈 수 있다.
ㄴ. 입력 모드에서 일반 모드로 돌아가려면 보통 `Esc`를 사용한다.
ㄷ. `:wq`는 저장 후 종료이고, `:q!`는 저장하지 않고 강제 종료할 때 쓰인다.
ㄹ. 명령행 모드는 파일 편집이 끝난 뒤 리눅스 셸을 반드시 종료시키는 모드이다.
① ㄱ, ㄴ
② ㄴ, ㄷ
③ ㄱ, ㄴ, ㄷ
④ ㄱ, ㄷ, ㄹ
⑤ ㄴ, ㄷ, ㄹ

14. [3점] 검색과 출력 관련 명령의 연결로 옳지 않은 것을 고른 것은?
① `grep -i error app.log` - 대소문자를 구분하지 않고 `error` 검색
② `grep -v root /etc/passwd` - `root`가 들어간 줄만 골라 출력
③ `tail -f app.log` - 파일 끝에 추가되는 내용을 계속 확인
④ `head -n 5 data.txt` - 앞에서 5줄 출력
⑤ `find /etc -type f -name "*.conf"` - `/etc` 아래의 일반 파일 중 이름이 `.conf`로 끝나는 파일 검색

15. [3점] 압축과 아카이브 명령에 대한 설명으로 옳은 것만을 고른 것은?

ㄱ. `gzip a.txt`는 일반적으로 `a.txt.gz`를 만들고 원본 `a.txt`는 남겨 둔다.
ㄴ. `gunzip a.txt.gz`는 gzip으로 압축된 파일을 해제할 때 쓸 수 있다.
ㄷ. `tar -cvf backup.tar a b`는 여러 파일을 하나의 tar 아카이브로 묶을 때 사용할 수 있다.
ㄹ. `tar -czvf backup.tar.gz a b`는 tar 아카이브 생성과 gzip 압축을 함께 수행하는 형태이다.
① ㄱ, ㄴ
② ㄱ, ㄷ
③ ㄴ, ㄷ
④ ㄴ, ㄷ, ㄹ
⑤ ㄱ, ㄴ, ㄷ, ㄹ

16. [3점] 환경과 날짜 관련 명령의 설명으로 가장 적절한 것은?
① `date +%Y-%m-%d`는 날짜 출력 형식을 지정할 수 있다.
② `echo $PATH`는 현재 작업 디렉터리 경로 하나만 출력한다.
③ `which ls`는 현재 실행 중인 모든 프로세스를 보여 준다.
④ `env`는 현재 디렉터리 안의 숨김 파일만 보여 준다.
⑤ `cal 2025`는 2025년의 패키지 설치 기록을 보여 준다.

17. [3점] 리눅스 파일 시스템에 대한 설명으로 옳은 것만을 고른 것은?

ㄱ. 리눅스의 최상위 디렉터리는 `/`이다.
ㄴ. 절대 경로는 현재 위치와 관계없이 `/`에서 시작하는 경로이다.
ㄷ. 상대 경로의 `.`은 상위 디렉터리, `..`은 현재 디렉터리를 뜻한다.
ㄹ. `/dev`, `/proc`, `/sys`처럼 실제 장치나 시스템 정보를 파일처럼 다루는 구조가 나타난다.
① ㄱ, ㄴ
② ㄱ, ㄷ
③ ㄴ, ㄹ
④ ㄱ, ㄴ, ㄹ
⑤ ㄴ, ㄷ, ㄹ

18. [3점] 현재 디렉터리가 `/home/user/docs`일 때 `/tmp/a.txt`를 가리키는 상대 경로로 가장 적절한 것은?
① `./tmp/a.txt`
② `../tmp/a.txt`
③ `../../tmp/a.txt`
④ `../../../tmp/a.txt`
⑤ `/home/user/docs/tmp/a.txt`

19. [3점] `ls -l`의 첫 글자가 나타내는 파일 종류 연결로 옳은 것만을 고른 것은?

ㄱ. `-` - 일반 파일
ㄴ. `d` - 디렉터리
ㄷ. `l` - 하드 디스크 전체를 뜻하는 블록 장치
ㄹ. `c` - 문자 장치
① ㄱ, ㄴ
② ㄱ, ㄷ
③ ㄴ, ㄹ
④ ㄱ, ㄴ, ㄹ
⑤ ㄴ, ㄷ, ㄹ

20. [3점] 주요 디렉터리의 용도 연결로 옳지 않은 것은?
① `/bin` - 기본 사용자 명령어가 위치하는 디렉터리
② `/etc` - 시스템 설정 파일이 위치하는 디렉터리
③ `/home` - 일반 사용자의 홈 디렉터리가 위치하는 디렉터리
④ `/dev` - 장치 파일이 위치하는 디렉터리
⑤ `/tmp` - 시스템의 영구 설정 파일만 저장하는 디렉터리

21. [3점] inode에 대한 설명으로 옳은 것만을 고른 것은?

ㄱ. inode는 파일의 소유자, 권한, 크기, 위치 등 메타데이터와 관련된다.
ㄴ. 파일 이름은 inode를 가리키는 항목으로 볼 수 있다.
ㄷ. `ls -i`는 파일의 inode 번호를 확인하는 데 쓰일 수 있다.
ㄹ. `stat`은 파일의 세부 정보를 확인하는 데 쓰일 수 있지만 inode 정보와는 관련이 없다.
① ㄱ, ㄴ
② ㄱ, ㄷ
③ ㄴ, ㄹ
④ ㄱ, ㄴ, ㄷ
⑤ ㄴ, ㄷ, ㄹ

22. [3점] 사용자와 그룹에 대한 설명으로 옳은 것만을 고른 것은?

ㄱ. `root`는 보통 UID 0을 갖는 관리자 계정이다.
ㄴ. 일반 사용자는 보통 UID 1000 이상으로 배정되는 경우가 많다.
ㄷ. 사용자는 반드시 하나 이상의 그룹에 속한다.
ㄹ. 새 사용자 계정을 만들면 같은 이름의 그룹은 절대 만들어지지 않는다.

① ㄱ, ㄴ
② ㄱ, ㄷ
③ ㄴ, ㄹ
④ ㄱ, ㄴ, ㄷ
⑤ ㄴ, ㄷ, ㄹ

23. [3점] `/etc/passwd`와 `/etc/group`에 대한 설명으로 가장 정확한 것은?
① `/etc/passwd`에는 사용자 이름, UID, GID, 홈 디렉터리, 로그인 셸 정보가 포함될 수 있다.
② `/etc/passwd`에는 모든 사용자의 실제 비밀번호 평문만 저장된다.
③ `/etc/group`에는 파일 권한의 숫자 모드만 저장된다.
④ `/etc/group`의 GID는 항상 UID와 같아야 하므로 별도 의미가 없다.
⑤ `/etc/passwd`는 그룹 목록 파일이고 `/etc/group`은 사용자 홈 디렉터리 목록 파일이다.

24. [3점] 사용자와 그룹 관리 명령의 연결로 옳은 것만을 고른 것은?

ㄱ. `adduser --ingroup dev dog` - `dev` 그룹을 기본 그룹으로 하여 `dog` 사용자 생성
ㄴ. `id dog` - `dog` 사용자의 UID, GID, 소속 그룹 확인
ㄷ. `su - dog` - `dog` 사용자 환경으로 전환
ㄹ. `deluser --remove-home dog` - `dog` 계정과 홈 디렉터리 삭제 시도
① ㄱ, ㄴ
② ㄴ, ㄷ
③ ㄱ, ㄷ, ㄹ
④ ㄴ, ㄷ, ㄹ
⑤ ㄱ, ㄴ, ㄷ, ㄹ

25. [3점] 사용자 `dog`가 `sudo apt update`를 실행했더니 `dog is not in the sudoers file`이라는 오류가 났다. 해결 방향으로 옳은 것만을 고른 것은?

ㄱ. 관리자 권한으로 `usermod -aG sudo dog`를 실행해 sudo 그룹에 추가할 수 있다.
ㄴ. `/etc/sudoers`를 직접 수정해야 한다면 `visudo`를 사용하는 것이 바람직하다.
ㄷ. `%sudo ALL=(ALL:ALL) ALL`에서 `%sudo`는 `sudo`라는 사용자 한 명만을 뜻한다.
ㄹ. sudo는 관리자 권한이 필요한 명령을 일시적으로 실행하게 해 주는 방식이다.
① ㄱ, ㄴ
② ㄱ, ㄷ
③ ㄴ, ㄹ
④ ㄱ, ㄴ, ㄹ
⑤ ㄴ, ㄷ, ㄹ

26. [3점] 소유권과 그룹 변경에 대한 설명으로 옳지 않은 것은?
① `chown dog a.txt`는 파일 소유자를 `dog`로 바꾸는 명령 형식이다.
② `chown dog:dev a.txt`는 소유자와 소유 그룹을 함께 바꿀 수 있다.
③ `chgrp dev a.txt`는 파일의 소유 그룹을 `dev`로 바꿀 수 있다.
④ `chown -R dog:dev project`는 디렉터리 내부까지 재귀적으로 적용하는 형태이다.
⑤ 일반 사용자는 자신이 소유하지 않은 모든 파일의 소유자를 자유롭게 바꿀 수 있다.

27. [3점] 권한 숫자 표기와 의미의 연결로 옳은 것만을 고른 것은?

ㄱ. `r=4`, `w=2`, `x=1`로 계산한다.
ㄴ. `755`는 소유자 `rwx`, 그룹 `r-x`, 기타 사용자 `r-x`이다.
ㄷ. `640`은 소유자 `rw-`, 그룹 `r--`, 기타 사용자 `---`이다.
ㄹ. `777`은 모든 사용자에게 모든 권한을 주므로 보안상 주의해야 한다.
① ㄱ, ㄴ
② ㄴ, ㄷ
③ ㄱ, ㄷ, ㄹ
④ ㄴ, ㄷ, ㄹ
⑤ ㄱ, ㄴ, ㄷ, ㄹ

28. [3점] 디렉터리 권한에 대한 설명으로 가장 적절한 것은?

① 디렉터리의 `r` 권한은 그 디렉터리 안의 파일 내용을 무조건 수정할 수 있음을 뜻한다.
② 디렉터리의 `w` 권한은 파일 생성, 삭제, 이름 변경과 관련되며 실제 사용에는 실행 권한도 중요하다.
③ 디렉터리의 `x` 권한은 그 디렉터리를 압축 파일로 바꾸는 권한이다.
④ 파일의 `x` 권한과 디렉터리의 `x` 권한은 모두 화면에 목록을 출력하는 권한이다.
⑤ 디렉터리에 `rwx`가 있어도 내부 파일 권한은 전혀 고려하지 않는다.

29. [3점] 현재 `report.sh`의 권한을 다음과 같이 바꾸려 한다.

- 소유자: 읽기, 쓰기 가능
- 그룹: 읽기만 가능
- 기타 사용자: 모든 권한 없음

가장 적절한 명령은?

① `chmod 600 report.sh`
② `chmod 640 report.sh`
③ `chmod 644 report.sh`
④ `chmod 750 report.sh`
⑤ `chmod 777 report.sh`

30. [3점] `data.txt`에서 기타 사용자의 모든 권한을 제거하고, 그룹에는 쓰기 권한을 추가하려 한다. 가장 적절한 명령은?

① `chmod o-rwx,g+w data.txt`
② `chmod o+rwx,g-w data.txt`
③ `chmod a=rw data.txt`
④ `chmod u-rwx,g+w data.txt`
⑤ `chmod o=w,g+rwx data.txt`

31. [3점] 패키지 형식과 관리 도구의 연결로 옳은 것만을 고른 것은?

ㄱ. RPM 계열은 Red Hat, Fedora, CentOS, Rocky Linux 등과 관련된다.
ㄴ. DEB 계열은 Debian, Ubuntu, Linux Mint 등과 관련된다.
ㄷ. `rpm`과 `dpkg`는 저장소 기반 의존성 해결을 항상 자동으로 처리하므로 `yum`, `dnf`, `apt`가 필요 없다.
ㄹ. `dnf`는 Red Hat 계열에서 `yum`의 후속 성격을 갖는 도구로 볼 수 있다.

① ㄱ, ㄴ
② ㄱ, ㄷ
③ ㄴ, ㄹ
④ ㄱ, ㄴ, ㄹ
⑤ ㄴ, ㄷ, ㄹ

32. [3점] APT 명령에 대한 설명으로 옳지 않은 것은?

① `sudo apt update`는 저장소의 패키지 목록 정보를 갱신한다.
② `sudo apt upgrade`는 설치된 패키지를 새 버전으로 업그레이드하는 데 쓰인다.
③ `sudo apt install hello`는 필요한 의존 패키지 설치를 함께 처리할 수 있다.
④ `sudo apt remove hello`는 패키지와 설정 파일을 모두 완전히 삭제한다.
⑤ `sudo apt autoremove`는 더 이상 필요하지 않은 의존 패키지를 정리할 수 있다.

33. [3점] `hello` 패키지를 찾고, 정보를 확인한 뒤 설치하려는 흐름으로 가장 적절한 것은?

① `apt show hello` -> `apt search hello` -> `apt remove hello`
② `apt search hello` -> `apt show hello` -> `sudo apt install hello`
③ `dpkg -l hello` -> `apt purge hello` -> `apt show hello`
④ `apt upgrade hello` -> `apt cache hello` -> `sudo apt install hello`
⑤ `sudo apt autoremove hello` -> `apt search hello` -> `dpkg -r hello`

34. [3점] 패키지 관리 상황에 대한 판단으로 옳은 것만을 고른 것은?

ㄱ. 저장소 목록을 갱신하지 않고도 설치가 될 수는 있지만, 최신 정보를 반영하려면 `apt update`가 필요하다.
ㄴ. `apt update`만 실행하면 이미 설치된 모든 패키지가 자동으로 새 버전으로 교체된다.
ㄷ. 설정 파일까지 지우려면 `apt purge`를 고려할 수 있다.
ㄹ. 설치 후 확인에는 `dpkg -l | grep hello` 같은 방식이 쓰일 수 있다.

① ㄱ, ㄴ
② ㄱ, ㄷ
③ ㄴ, ㄹ
④ ㄱ, ㄷ, ㄹ
⑤ ㄴ, ㄷ, ㄹ

35. [3점] 시스템 종료와 재부팅 명령에 대한 설명으로 가장 적절한 것은?

① `logout`은 현재 세션을 끝내는 명령이며, 시스템 전체 전원을 끄는 명령으로 보기는 어렵다.
② `shutdown -c`는 즉시 재부팅을 의미한다.
③ `shutdown -r now`는 예약된 종료를 취소한다.
④ `reboot`는 현재 셸만 종료하고 시스템은 계속 실행한다.
⑤ 서버 환경에서는 여러 사용자가 있어도 일반 사용자가 언제든 무조건 종료 명령을 실행해야 한다.

36. [3점] 종료·재부팅 명령의 연결로 옳은 것만을 고른 것은?

ㄱ. `shutdown now` - 즉시 종료
ㄴ. `shutdown -r now` - 즉시 재부팅
ㄷ. `shutdown -c` - 예약된 종료 취소
ㄹ. `exit` - 모든 사용자 세션과 시스템 전원을 동시에 종료

① ㄱ, ㄴ
② ㄱ, ㄷ
③ ㄴ, ㄹ
④ ㄱ, ㄴ, ㄷ
⑤ ㄴ, ㄷ, ㄹ

37. [3점] 다음 작업 목표와 명령의 연결 중 가장 적절하지 않은 것은?

① 현재 위치 확인 - `pwd`
② 홈 디렉터리로 이동 - `cd ~` 또는 `cd`
③ 숨김 파일까지 자세히 보기 - `ls -al`
④ 상위 디렉터리로 이동 - `cd -`
⑤ 명령어 사용법 확인 - `man 명령어`

38. [3점] `/home/dog/shared.txt`를 `dog`와 `cat`이 함께 읽고 쓰게 하되, 그 밖의 사용자는 접근하지 못하게 하려 한다. 두 사용자가 모두 `dev` 그룹에 속한다고 할 때 파일에 대한 설정으로 가장 적절한 것은?

① `chgrp dev shared.txt` 후 `chmod 660 shared.txt`
② `chgrp dev shared.txt` 후 `chmod 600 shared.txt`
③ `chown root:root shared.txt` 후 `chmod 777 shared.txt`
④ `chmod 666 shared.txt`만 실행
⑤ `chmod o+rw shared.txt` 후 `chmod g-rw shared.txt`

39. [3점] 관리자가 다음 요구를 처리하려고 한다.

- `/etc` 아래에서 이름이 `.conf`로 끝나는 일반 파일을 찾는다.
- `app.log` 끝에 새로 추가되는 내용을 실시간에 가깝게 확인한다.
- `data.txt`에서 `error`가 들어가지 않은 줄만 출력한다.

각 요구에 알맞은 명령 묶음으로 가장 적절한 것은?

① `grep /etc -type f -name "*.conf"`, `head -f app.log`, `grep error data.txt`
② `find /etc -type f -name "*.conf"`, `tail -f app.log`, `grep -v error data.txt`
③ `ls -R /etc "*.conf"`, `tail -n app.log`, `grep -i -r data.txt error`

④ `file /etc/*.conf`, `cat -f app.log`, `grep -r -v data.txt error`

⑤ `which /etc/*.conf`, `date -f app.log`, `grep --show error data.txt`

40. [3점] 다음 상황에서 가장 적절한 관리 흐름을 고른 것은?

Ubuntu 계열 실습 환경에서 `hello` 패키지를 설치해 실행 가능 여부를 확인한 뒤, 설정 파일까지 포함해 제거하고 남은 불필요 의존 패키지도 정리하려 한다.

① `sudo apt update` -> `sudo apt install hello` -> `dpkg -l | grep hello` -> `sudo apt purge hello` -> `sudo apt autoremove`

② `sudo apt upgrade hello` -> `sudo apt remove --config hello` -> `docker rmi hello` -> `sudo apt update`

③ `rpm -i hello` -> `yum remove hello` -> `apt search hello` -> `shutdown now`

④ `sudo apt remove hello` -> `apt show hello` -> `sudo apt install hello` -> `logout`

⑤ `dpkg -l | grep hello` -> `sudo apt autoremove hello` -> `sudo apt update --purge hello`