Question no.1

Type env on your system and tell me what environment variables are set.

⇨ The env command is used to print a list of environment variables and their values.

```
[19706@ip-172-26-2-101:~$ env
SHELL=/bin/bash
PWD=/home/19706
LOGNAME=19706
MOTD_SHOWN=pam
HOME=/home/19706
LANG=C.UTF-8
SSH_CONNECTION=99.127.78.47 49720 172.26.2.101 1022
TERM=xterm-256color
USER=19706
SHLVL=1
SSH_CLIENT=99.127.78.47 49720 1022
XDG_DATA_DIRS=/usr/local/share:/usr/share:/var/lib/snapd/desktop
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/us
r/local/games:/snap/bin
SSH_TTY=/dev/pts/4
_=/usr/bin/env
19706@ip-172-26-2-101:~$ 
```

Question no.2

Pick 3 environment variables from the output and let me know the use for each one of them.

⇨ SHELL: It specifies the path to the shell that being is used. In my terminal its path to the Bourne-Again shell. The value of the 'SHELL' variable determines which shell will be used when the user opens a terminal window or runs a shell script.

```
[19706@ip-172-26-2-101:~$ echo SHELL $SHELL
SHELL /bin/bash
```

⇨ PWD: It specifies the current working directories. This directory is where the user is currently located in the file system. It outputs the absolute path of the directory we are currently in.

```
19706@ip-172-26-2-101:~$ echo pwd $PWD
pwd /home/19706
```

⇨ PATH: It specifies a colon-separated list of directories that the shell should search for when trying to execute a command. This command allows us to run executables in a different directory without having to specify the full path each time.

```
[19706@ip-172-26-2-101:~$ echo path $PATH
path /usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/us
r/local/games:/snap/bin
```

Question no.3
For each of the 3 environment variables you selected in Q2, can you modify it? Should you? Some environment variables should not be modified.

⇨ Yes, it is possible to modify the environment variables in Linux. However, the way we modify them, and the effect of the modification depends on the type of shell we are using and the scope of the environment variables.

⇨ Whether we modify an environment variable depends on your specific use case and desired outcomes.

Some of the environments which I modified are:

  i)   Home: It stores the path to the home directory of the current user which is used by various commands and applications to determine where to store user-specific data, such as configuration files, user documents, and other files.

```
19706@ip-172-26-2-101:~$ echo $HOME
/home/19706
19706@ip-172-26-2-101:~$ export HOME="/new/home/directory"
19706@ip-172-26-2-101:/home/19706$
19706@ip-172-26-2-101:/home/19706$ echo $HOME
/new/home/directory
19706@ip-172-26-2-101:/home/19706$ export HOME="/home/19706"
19706@ip-172-26-2-101:~$ echo $HOME
/home/19706
19706@ip-172-26-2-101:~$
```

  ii)   LOGNAME: It holds the username of the currently logged-in user. This variable is commonly used in shell scripts and batch files to identify the user who is executing the script of command.

```
19706@ip-172-26-2-101:~$ echo $LOGNAME
19706
19706@ip-172-26-2-101:~$ export LOGNAME="new_logname"
19706@ip-172-26-2-101:~$ echo $LOGNAME
new_logname
19706@ip-172-26-2-101:~$ export LOGNAME="19706"
19706@ip-172-26-2-101:~$ echo $LOGNAME
19706
19706@ip-172-26-2-101:~$
```

⇨ Yes, that's correct. Some environments should not be modified because they are essential for the functioning of the system or applications. Modifying these environment variables can cause the system or applications to behave unexpectedly or stop functioning altogether.

For some examples:

1) SHELL: It specifies the default shell for the user. Modifying this variable may cause unexpected behavior in the shell.

```
19706@ip-172-26-2-101:~$ echo $SHELL
/bin/bash
```

2) SSH_CONNECTION: It contains information about the current ssh_connection. Modifying this variable may cause unexpected behavior in ssh.

```
19706@ip-172-26-2-101:~$ echo $SSH_CONNECTION
209.36.108.210 13995 172.26.2.101 1022
```

3) SSH_TTY: It specifies the tty name that is used by the ssh connection. Modifying this variable may cause unexpected behavior in ssh.

```
19706@ip-172-26-2-101:~$ echo $SSH_TTY
/dev/pts/0
19706@ip-172-26-2-101:~$
```