1) What is GIT? What is version control system mention few advantages? Commands to install Git and generate ssh key.

- Git is the most versatile distributed version control system.
- The way GIT tracks and handles file changes in very efficient and different than how other version control software tracks and the changes.
- A version control system (VCS) remembers the version of the software as it moves through the various stages of development.

Some of the benefits using version control system are.

→ Long-term history - It maintains the history of who, what and when of any modifications to the software code.

→ Branching and merging - Most version control have branching and merging capabilities. So the developer can divide the works into multiple streams.

→ Trace ability:- Version control gives developers a way to easily and reliably trace bugs and problems in a software.

→ Installing GIT

Once the installization of GIT is done:-
Check upon the following commands.

$ where is git
/usr/local/bin/git
$git --version
git.version 1.7.6

The first step is to specify your username and email address to your GIT repository using "git config"

git config --global user.name "GIT User"

git config --global user.email User@gmail.com

Creating a new SSH Key gen.

The command to generate a local SSH pair of keys

ssh-Keygen -t ed25519 -C "shireeshakb@gmail.com"

ssh-keygen → command-line tool used for creating new pair of SSH keys.

-t → The flag is used to indicate the algorithm used to create the digital signature of the key pair.

ed25519 → is the best algorithm you can use to create SSH Key pairs.

-C → flag is used to provide a custom comment at the end of the public key.

After executing this command it generates 2 files in your directory you select the public key with the .pub extension, and the private one without an extension.

Add your SSH private key.

ssh -add ~/.ssh/demo

demo is your SSH key.

Add SSH Key to GitHub Account

Step 1> Copy your SSH public Key to your clipboard.

Cat ~/.ssh/demo.pub

Step 2> log in to Git hub

↓

Go to settings

↓

SSH and GPC Keys

↓

New SSH key

↓

Give your new SSH key on github a title.

Paste the key into the key area.

↓

Add your SSH key

Commands to be executed once your SSH-key is generated.

ssh -T git@github.com

eval 'ssh-agent -s'

exec ssh-agent bash

ssh add (name of repo name)

→ To login through terminal (to login to the git hub)

→ Clone your repository to your local machine

From your repository page on GitHub, click on the button clone, copy the URL for your repository.

Change the current working directory to the location where you would like to clone your repository.

cd foldername

- The git clone command copies your repository from git hub to your local computer.

  git clone https://github.com/YOUR-USERNAME/YOUR-REPOSITORY

- To see the repository name and output numbers that you can see on your computer, representing the total file size, etc, ~~may~~

  git clone repo-path-here

→ Git add and Git commit

Change the directory to the folder where your repository is present

cd (folder name)directoryname

$ ls -a
all the files that exist in your GitHub repository will be displayed.

→ Edit a file in your repo
Open the editor make few changes to the README.md
save your changes
Check the changes you have done to that particular file.
$ git status.

→ Add a file

$ git add file-name-here-with-extension

TO add the README.md file that you just modified you'd use:
git add README.md

To add many files you can also use

$ add --all

→ Commit files

The uses of Commit are
- To help collaborators and your future self understand what was changed and why.
- allow you and your collaborators to find changes that were previosly made.

$ git commit -m "Editing the README to try out git add/commit"

Commit the message using

$ git commit.

- Once you save your commit message and exit the text editor, the file that you created will contain your commit message.

→ Push changes to Git

So for the changes that you would have made will be only in your local copy of the repository.
To add the changes to your git repo.
You can push the changes to git using

$ git push.

Once you have pushed your commits, visit your repository on GitHub and notice that your changes are reflected there.

2) What is function? What are the difference between local variables and Global variable with example? Different types of function s with example?

- The function is the set of commands that can be called multiple times it is used to avoid writing the same block of code again and again

Local variable → These variables whose scope is within the function where it is declared.

Global variable → These variables can be accessed globally in the entire program.

Local variable	Global variables
• The local keyword is used to declare the local variable	• No keyword is used
• It is declared inside the function	• It can be declared any where in the program
• It doesn't provide data sharing	• It provides data sharing.
• These are stored on the stack	• These variables are stored in the fixed memory location by the compiler.
• Passing parameters is necessary	Parameters passing is not necessary
• Garbage value is stored if not initialized	• Zero is stored by default if not initialized
• If modification or modified in one function that particular modification doesn't reflect in other function	• If modified in one function that modification will be visible in whole program.

Example for creating local variable.

```
function hello() {

    local x = "RVCE"
echo "I am studying in x= $x"

}
hello
echo "I am studying in x = $x"
```

Output

```
I am studying in RVCE
I am studying in x =
```

Example for creating a Global variable.

```
function LSS() {

  x = "Linux shell scripting"
echo "full form of LSS is x= $x"

}
LSS
echo "full form of LSS is x= $x"
```

```
full form of LSS is x = Linux shell scripting
full form of LSS is x = Linux shell scripting.
```

→ Creating a function

Syntax:

```
{ function-name() {
      list of commands
  }
```

Example for creating a function.

```
Hello () {
   echo " This is shireesha"
}
$./try.sh
O/p -> This is shireesha.
```

→ Pass parameters to a function.

You can define a function that will accept parameters while calling the function. These parameters would be represented by $1 $2 and so on.

```
try() {
  echo " My full name is $1 $2"
  }

./try1.sh
My full name is Shireesha KB.
```

$1 and $2 are the 2 parameters passed Shireesha KB

→ Returning Values from Functions.

If you execute an exit command from inside a function, its effect is not only to terminate execution of the function but also of the shell program that called the function.
If you instead want to just terminate execution of the function then there is a way to come out of a defined function.

You can return any value from the function using return command.

```
try() {
    echo "Shireesha is studying in nvce"
}
try() {
    echo "Which college? $1 $2"
    return 10
}
ret = $?
echo "Return value is $ret"
```

```
$./try3.sh
Which college from nvce
Return value is 10
```

$1 => from $2 => nvce (parameters passed)
10 -> return value

→ Nested function
One of the most important feature of functions.
A function that calls itself is known as recursive function.

```
first function() {
    echo "This is first function"
    number_two
}
number_two() {
    echo "This is second function"
}
```

OUTPUT

This is first function
This is second function.