```
git init
git status
git add .
git commit -m "Hello" → git branch -m main
git remote add origin https:// ...
git push origin main
[ git config --global username " " 
  git  --  --  .email " -- "
[ git remote -v
  git remote remove origin ]
```

```
python → python3 a.py // Run | Jupyter notebook
fortran → gfortran .f90 // Run
```

→ ssh 10 ... @user. // login to cluster

git clone https

→ // Fortran Code

```
INTEGER F, N, T
Print *, " Enter int"
Read *, N
F=1
Do 10 I = T: N
   F= F*I
10 continue.
Print*, Fact of N, " ", F
```

→
```
def Asth exp(x,n):
    result = 1.0;
    term = 1;
    for i in range (1, n+1):
        term = term * x/i
        result += term
    return result
# Plot  import matplotlib as plt.
x = np.linspace (-1,1,(100)
y = np.exp (x)

plt.figure()
plt.plot (x,y)
plt.xlable ('x')
plt.ylable ('exp(x)').
```

Python N.R.

```
def f(x)
    return x^2 - 2x+2
def g(x)
    return 2x-2
a= float (input ("Initial gess = "))
n= int(input("number iter- "))
K=1
while (K ≤ n)
    r= a - f(a)/g(a)
    Print ("root= "r);
    a=r.
```

Bise.

```
def f(x).
    ret
a=1
b=2
while (abs(a-b) ≥ 0.01
    c= (a+b)/2
    if f(a). f(b) < 0.
        b = c
    else a=c
    Print c.
```

```
git init
git status
git add .
git commit -m "Hello" → git branch -m main
git remote add origin https://—
git push origin main
[ git config --glob username "As"
  git                       .email "—" ]
[ git remote -v
  git remote remove origin ]
```

python → python3 a.py // Run — Jupyter notebook
fortran → gfortran -f90 // Run - [illegible]

→ ssh 10.—@user. // login to clusters | git clone RH

→ // Fortran Code

```
INTEGER F, N, T
Print *, "Enter int"
Read *, N
F=1
Do 10 I = T: N
  F= F*I
10 continue
Print*, Fact of N, F
```

→
```
def Ash_exp(x,n):
    result = 1.0;
    term = 1;
    for i in range (1, n+1):
        term = term * x/i
        result += term
    return result
# Plot (import matplotlib as plt.
x = np.linspace(-1,1,(100)
y= np.exp(x)

plt.figure()
plt.plot(x,y)
plt.xlable('x')
plt.ylable('exp(x)').
```

Python N.R.

```
def f(x)
    return x^2-2x+2
def g(x)
    return 2x-2
a= float(input("Initial guess = "))
n= int(input("number iteration "))
K=1
while(K ≤ n)
    r= a - f(a)/g(a)
    Print ("root=" r);
    a=r.
```

Bise.

```
def f(x):
    ret
a=1
b=2
while (abs(a-b) ≥ 0.01
    c= (a+b)/2
    if f(a).f(b) < 0.
        b=c
    else a=c
    Print c.
```