README File:

- **Project Team –**
  - Shromana Kumar
  - Sangpil Youm

- **Steps to execute program –**
  - First, we have to compile the program in Erlang.

    Command:
    c(assignment1).

  - Then we have to execute the *"assignment1:master(<No_Of_Input_Zeroes>)"* function with an input of the no of zeroes needed in the bitcoin.

    Command:
    assignment1:master(1).

```
(shromana@10.136.82.238)1> c(assignment1).
{ok,assignment1}
(shromana@10.136.82.238)2> assignment1:master(1).
1
shromana.kumar;ZYVWxrfy
0a927bcae5c850d6c56f37673556d197a66282c4ecd233b593d6a19742944085

Ending Statistics .....
CPU Time is 0.125 seconds
Real Time is 0.14 seconds
ok
```

- **Steps to setup Server Worker Connection –**
  - Start erlang shell with below command having your IP address for both Server and Worker node. Note that both server and worker have to have the same cookie and the IP address has to be correct for the connection to be setup successfully. The name of the terminal should be in the format of *"name@ip_address"* and name can be anything.

    Command:
    erl -name shromana@10.20.0.203 -setcookie guitar
    erl -name shro@10.20.0.203 -setcookie guitar

```
PS C:\Users\shrom\Downloads\DOSP\Erlang> erl -name shromana@10.20.0.203 -setcookie guitar
Eshell V13.0.4  (abort with ^G)
(shromana@10.20.0.203)1>
```

```
C:\Users\shrom\Downloads\DOSP\Erlang>erl -name shro@10.20.0.203 -setcookie guitar
Eshell V13.0.4  (abort with ^G)
(shro@10.20.0.203)1>
```

- For the worker node to request work from the Server, it needs to call the below function *"assignment1:start_worker(<Terminal_Name>)"* and pass its terminal name. If successful connection is established then erlang shell will return **true** else it will return **false** or **ignored.**

  Command:
  assignment1:start_worker('shro@10.20.0.203').

```
(shromana@10.20.0.203)2> assignment1:start_worker('shro@10.20.0.203').
true
```

- **Size of Work Unit –**
  - Our problem here is to mine bitcoins with the input number of zeroes. So, we have decided to create a work unit of 4 actors (2 actors per working nodes) for our problem which would provide ample scope for parallelism, without slowing our machines down which aren't built for industrial purposes.

- **Result of running the program with input size 4 –**

Please find below the result of running the program with an input size 4

```
(shromana@10.136.82.238)8> assignment1:master(4).
4
shromana.kumar;kGcRpRg=
000057917ecb11114512a7981430bd8435aab86cdfb0dc96411de3bfd183bcfc

Ending Statistics .....
CPU Time is 1.782 seconds
Real Time is 0.893 seconds
ok
```

- **Running Time and CPU Time –**
  - We are printing the running time and the CPU time on the terminal. We are performing this activity using Erlang library functions. As you can see that the ratio of CPU time to Running Time is more than 1, which proves that we have multiple actors working behind the scenes and hence we have parallelism in our program.

```
(shromana@10.136.82.238)14> assignment1:master(6).
6
shromana.kumar;/ATyHWz/
000000f64ee78664cf889b6dde3c2633bcb38a7205b021c431e99b6babe0b755

Ending Statistics .....
CPU Time is 82.125 seconds
Real Time is 41.931 seconds
ok
```

- **Coins with most 0s –**

  We found a bitcoin with a maximum input of 8 leading 0s

```
(shromana@10.136.82.238)1> assignment1:master(8).
8
shromana.kumar;Cqp22AzL
00000000074b3a42c3b04c86463297c1d589793066efb3d35978fff1c77cf794

Ending Statistics .....
CPU Time is 4094.859 seconds
Real Time is 2114.924 seconds
ok
(shromana@10.136.82.238)2>
```

- **Largest number of working machines you were able to run your code with –**
  - We were able to connect 2 working machines together and mine coins using both the machines. We can show the connection using the command “nodes().”

```
(shromana@10.136.82.238)2> net_kernel:connect_node('youms@10.136.214.191').
true
(shromana@10.136.82.238)3> nodes().
['youms@10.136.214.191']
```

```
Eshell V13.0.4  (abort with ^G)
(youms@10.136.214.191)1> nodes().
['shromana@10.136.82.238']
(youms@10.136.214.191)2>
```

  - When we use the command “i().” On the erlang shell, we can see the list of process running. In the below screenshot, you can only see the actor/miner process running

on the worker node. This list of processes would not have the master function call as that would be running on the Server node.

```
<0.68.0>              supervisor_bridge:user_sup/1          610      107    0
                      gen_server:loop/7                      11
<0.69.0>              user_drv:server/2                    2586    45910    0
user_drv              user_drv:server_loop/6                  9
<0.70.0>              group:server/3                        233      104    0
user                  group:server_loop/3                     4
<0.71.0>              group:server/3                       2586   228091    0
                      group:server_loop/3                     4
<0.72.0>              kernel_config:init/1                  233       55    0
                      gen_server:loop/7                      11
<0.73.0>              kernel_refc:init/1                    233       58    0
kernel_refc           gen_server:loop/7                      11
<0.74.0>              supervisor:kernel/1                   233       97    0
kernel_safe_sup       gen_server:loop/7                      11
<0.75.0>              supervisor:logger_sup/1               610      411    0
logger_sup            gen_server:loop/7                      11
<0.76.0>              logger_handler_watcher:init/1         233       56    0
logger_handler_watche gen_server:loop/7                      11
<0.77.0>              logger_olp:init/1                     376       90    0
logger_proxy          gen_server:loop/7                      11
<0.79.0>              logger_olp:init/1                     376      169    0
logger_std_h_default  gen_server:loop/7                      11
more (y/n)? (y)
<0.80.0>              erlang:apply/2                        233       13    0
                      logger_std_h:file_ctrl_loop/1           4
<0.83.0>              erlang:apply/2                       6772    22085    0
                      shell:shell_rep/4                      17
<0.86.0>              erlang:apply/2                        987   228820    0
                      erpc:execute_call/3                    57
<0.98.0>              inet_tcp_dist:do_accept/7            1598     2689    0
                      dist_util:con_loop/2                    8
<0.106.0>             assignment1:mining/3                  376     1019    0
                      assignment1:compare/6                   1
<0.107.0>             assignment1:mining/3                  376     3606    0
                      assignment1:compare/6                   1
Total                                                     40716   736088    0
                                                            407

ok
(youms@10.136.214.191)9> i().
```