Details of team members:

1. Shiridi Sai Prasad

UFID - 8073-3499

Mail: shiridisaiprasad@ufl.edu

2. Aditya Bharadwaj

UFID: 9890-1939

Mail: aditya.bharadwaj@ufl.edu

Project Information & Instructions to run the program:

1. The starting point of the execution is in the proj1.exs file in the main directory.

Please enter the directory and run “mix run proj1.exs starting\_number last\_number”

2. Please find attached the design document(Project\_Design file) in the zip package.

Details on the Worker creation and Task Delegation:

1. We first initiate an Agent (VampStore), which is accessible to all the processes and stores the vampire numbers.

2. We initiate the Supervisor and create 4 workers (GenServers) dividing the numbers equally. We tried different number of workers, and by calibration we decided to keep the number to 4. As we increase the number of workers, we did not notice significant improvement in the performance. Also, since we are working on 4core processors, we thought adding more number of workers will cause unnecessary delays and wait times for context switching. So, we kept the number of workers down to 4.

3. After the GenServer workers are created, from each worker, we create 8 Asynchronous stream tasks by (Task.async\_stream) to parallelly compute the permutations of 8equal groups of numbers. Again, the number of the Asynchronous tasks is decided by calibration.

4. Now, if a vampire number is found, the tasks will directly add the numbers to the Agent registry. The agent registry can be accessed by any process.

5. Once the processing is complete, we retrieve the vampire numbers list from the Agent and print it.

Performance of the project:

System Tested on: MacBook Pro (2.6 GHz 4core processor)

Real time: 17.716seconds

User time: 1minute 51.226 seconds

Sys time: 6.681seconds

(User time + Sys time): Real time = 6.66

Please find the CPU utilization chart and the project design flow chart in Prasad\_Bharadwaj.zip package