

**1.Group Members:**

Aakriti Gupta,1435-4171

Shilpi Kansal, 3358-1210

**2. No of workers created**

The number of actors created is dynamic based on the given range.Each worker actor is made to work on a certain range of numbers ( 20, in our case ).

**3.Size of the work unit of each worker actor that you determined results in best performance for your implementation and an explanation on how you determined it.**

The best performance achieved in our code was for work unit size of 20. Each worker actor processed 20 numbers to give the best performance. The number 20 was achieved by hit and trial and then checking CPU to real time ratio. We did try for numbers like 10,50,100,1000.

**4.The result of running your program for: mix run proj1.exs 100000 200000**

It prints all the vampire numbers and its fangs as shown in the screenshot below.

```

SpawnCalculator.ex — dosproject1-3
32  rend=start+count

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Compiling 1 file (.ex)
193945 395 491
125500 251 500
124483 281 443
125248 152 824
104260 260 401
129640 140 926
105264 284 516
105210 210 501
115672 152 761
125433 231 543
125460 204 615
125460 246 510
102510 201 510
175329 231 759
152685 261 585
180225 225 801
136525 215 635
129775 179 725
174370 371 470
118440 141 840
126846 261 486
126027 201 627
150300 300 501
105750 150 705
140350 350 401
117067 167 701
182650 281 650
108135 135 801
146952 156 942
173250 231 750
136948 146 938
197725 275 719
135828 231 588
110758 158 701
180297 201 897
145314 351 414
145137 317 461
152608 251 608
190260 210 906
192150 210 915
156289 269 581
182250 225 810
135837 351 387
163944 396 414
156240 240 651
131242 311 422
  
```

```
32  rend=start+count

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

136525 215 635
129775 179 725
174370 371 470
118440 141 840
126846 261 486
126827 201 627
150300 300 501
105750 150 705
140350 350 401
117067 167 701
182690 281 650
108135 135 801
146952 156 942
173250 231 750
136948 146 938
197725 275 719
135828 231 588
110758 158 701
180297 201 897
145314 351 414
146137 317 461
152608 251 608
190260 210 906
192150 210 915
156289 269 581
182250 225 810
135837 351 387
163944 396 414
156240 240 651
131242 311 422
120600 201 600
153436 356 431
193257 327 591
172822 221 782
162976 176 926
116725 161 725
156915 165 951
186624 216 864
123354 231 534
134725 317 425
132430 323 410
133245 315 423
```

**5. Report the running time for the above problem (4). The ratio of CPU time to REAL TIME tells you how many cores were effectively used in the computation. If you are close to 1 you have almost no parallelism (points will be subtracted).**

Ratio of CPU and REAL time is on 6.99 8 core machine.

Code File Edit Selection View Go Debug Terminal Window Help

mix.exs — GuptaKansal

EXPLORER

OPEN EDITORS

- mix.exs

GUPTAKANSAL

- \_build
- .elixir\_ls
- lib
- test
- .formatter.exs
- .gitignore
- mix.exs
- proj1.exs
- README.md
- ReadMe.pdf

defmodule Dosproiect1.MixProject do

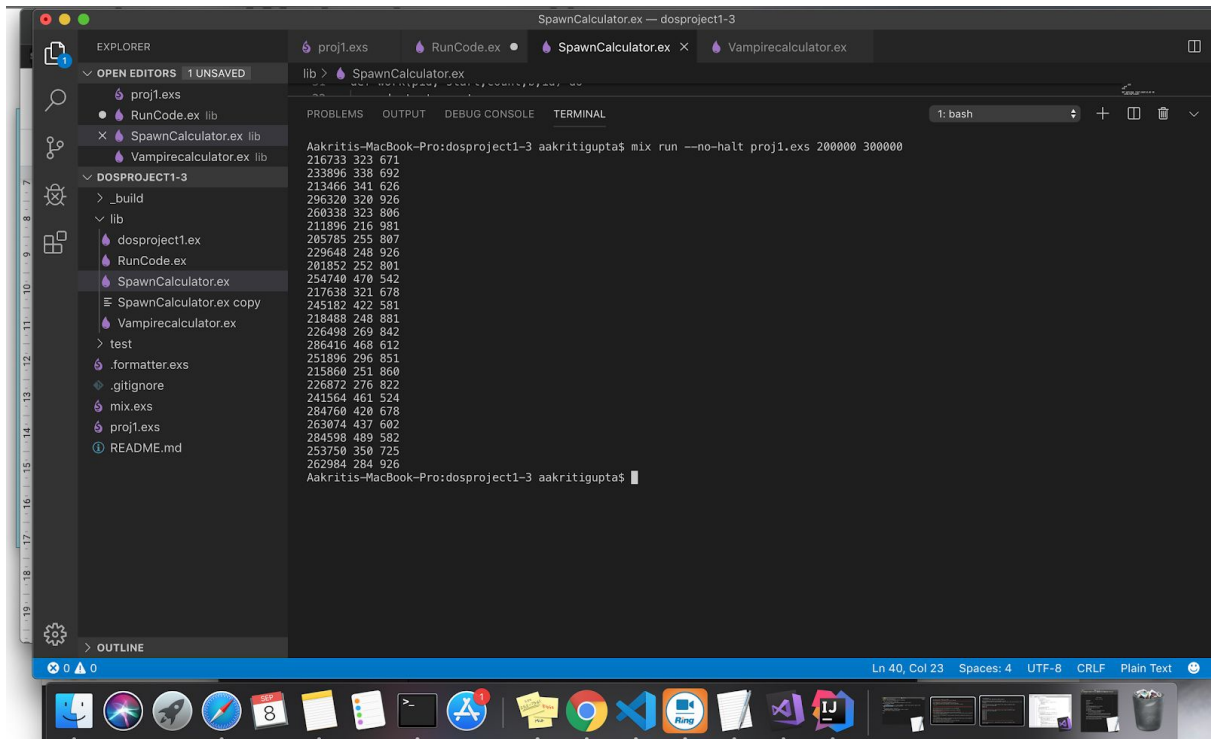
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

136525 215 635  
152685 261 585  
124483 281 443  
140350 350 401  
102510 201 510  
125248 152 824  
126027 201 627  
129640 140 926  
175329 231 759  
193945 395 491  
180225 225 801  
192150 210 915  
125460 204 615  
125460 246 510  
174370 371 470  
186624 216 864  
150300 300 501  
104260 260 401  
173250 231 750  
135828 231 588  
136948 146 938  
126846 261 486  
117067 167 701  
118440 141 840  
197725 275 719  
180297 201 897  
110758 158 701  
105750 150 705  
145314 351 414  
133245 315 423  
135837 351 387  
152608 251 608  
162976 176 926  
125433 231 543  
131242 311 422  
129775 179 725  
146952 156 942  
120600 201 600  
156289 269 581  
116725 161 725  
146137 317 461  
182250 225 810  
132430 323 410  
123354 231 534  
182650 281 650  
163944 396 414  
172822 221 782  
193257 327 591  
190260 210 906  
134725 317 425  
153436 356 431  
156240 240 651  
156915 165 951

real 0m0.744s  
user 0m5.037s  
sys 0m0.125s

**6.The largest problem you managed to solve (For example You can try finding out bigger vampire numbers than 200000).**

Tried for running it from 2000000 to 3000000



## **How to run:**

Unzip the folder aakriti\_shilpi.zip and save it to some directory

Navigate to the directory where you saved the code

Do: `cd aakriti_shilpi`

Run the following command: `mix run --no-halt proj1.exe <N1> <N2>`

Eg. `mix run --no-halt proj1.exe 100000 200000`