**1.How many seconds are in an hour? Use the interactive interpreter as a calculator and multiply the number of seconds in a minute (60) by the number of minutes in an hour (also 60).**

**sol. 60**

**answer -** 60\*60

**2. Assign the result from the previous task (seconds in an hour) to a variable called seconds\_per\_hour.**

**Answer -** seconds\_per\_hour = 3600

**3. How many seconds do you think there are in a day? Make use of the variables seconds per hour and minutes per hour.**

**Answer -** seconds\_per\_hour\*24

86400

**4. Calculate seconds per day again, but this time save the result in a variable called seconds\_per\_day**

**Answer -**

seconds\_per\_day = seconds\_per\_hour\*24

seconds\_per\_day

86400

**5. Divide seconds\_per\_day by seconds\_per\_hour. Use floating-point (/) division.**

**Answer –**

seconds\_per\_day / seconds\_per\_hour

24.0

**6. Divide seconds\_per\_day by seconds\_per\_hour, using integer (//) division. Did this number agree with the floating-point value from the previous question, aside from the final .0?**

**Answer -**

seconds\_per\_day // seconds\_per\_hour

**7. Write a generator, genPrimes, that returns the sequence of prime numbers on successive calls to its next() method: 2, 3, 5, 7, 11, ...**

**Answer –**

def genPrimes():

primes = [ 2, 3, 5, 7, 11 ]

def isPrimeNumber(n):

if n in primes:

return True

for elem in primes:

if n % elem == 0:

return False

primes.append(n)

return True

num = 1

while True:

num += 1

if isPrimeNumber(num):

next = num

yield next

num = next

primeNumber = genPrimes()

for i in range(189):

print(primeNumber.\_\_next\_\_())

2

3

5

7

11

13

17

19

23

29

31

37

41

43

47

53

59

61

67

71

73

79

83

89

97

101

103

107

109

113

127

131

137

139

149

151

157

163

167

173

179

181

191

193

197

199

211

223

227

229

233

239

241

251

257

263

269

271

277

281

283

293

307

311

313

317

331

337

347

349

353

359

367

373

379

383

389

397

401

409

419

421

431

433

439

443

449

457

461

463

467

479

487

491

499

503

509

521

523

541

547

557

563

569

571

577

587

593

599

601

607

613

617

619

631

641

643

647

653

659

661

673

677

683

691

701

709

719

727

733

739

743

751

757

761

769

773

787

797

809

811

821

823

827

829

839

853

857

859

863

877

881

883

887

907

911

919

929

937

941

947

953

967

971

977

983

991

997

1009

1013

1019

1021

1031

1033

1039

1049

1051

1061

1063

1069

1087

1091

1093

1097

1103

1109

1117

1123

1129