Finding a large prime number

Code:

```
import math
import random


#miller_rabin method shows if we have a possible prime 'n' for the given 'a'
def miller_rabin(n, a):
   exp = n - 1

   if(n%2 == 0):
       return False


   while not exp & 1:
       exp >>= 1

   if pow(a, exp, n ) == 1:
       return True

   while exp < n - 1:
       if pow(a, exp, n) == -1:
           return True
       exp <<= 2

   return False


#this method calls the miller_rabin method but for different values of a to ensure
that miller rabin test is indeed returning true for primes
def different_a_test(n):

   for i in range(20):
       a = random.randint(2, n-1)           #picking a random integer "a" between 1 and
n-1
       if (miller_rabin(n, a) == False):
           return False
   return True
```

```
#this method picks a random value for n for the equation x = 2310K + n where gcd(n, x)
= 1
def pickRand():

   ## Set gcd flag to false
   gcdFlag = False

   ## Loop until gcd(n, 2310) = 1
   while (not gcdFlag):

       ## Generate a random int
       n = random.randint(1, 10000)

       ## Check if gcd of n and 2310 = 1
       if (math.gcd(n, 2310) == 1):
           gcdFlag = True

   ## Return the random integer such that gcd(n, 2310) = 1
   return n


#this method calculates the value of x from the n calculated from above, and then
calls the different_a_test(x) method to check if it;s a prime.
def check_prime():

   n = pickRand()
   found = False
   listK = []
   lRange = (math.pow(2,100) - 9999)/2310
   hRange = (math.pow(2,101) - 1)/2310

   #while we don't have a prime, we take a different value of k and then add that
value to the list, we will get a new value of x which will be checked for
   #different value of a as well
   while found == False:
       k = random.randint(lRange, hRange)
       listK.append(k)
       x = 2310*n + k
       a = random.randint(2, n-1)
       if(miller_rabin(x, a) == False):
```

```
            continue                    #if the number is not a possible prime, then we
get a different number
        else:
            if(different_a_test(x)):
                found = True            #the loop stops executing after we get a prime
   print(x, " is possibly a prime number we got after trying ", len(listK), " tries
for K.")

   #print (listK)
   #return listK


check_prime()



def largePrime(numBits):
  found = False
  while found == False:
      possiblePrime = random.getrandbits(numBits)
      if(different_a_test(possiblePrime)):
          found = True
          return possiblePrime

print("A 1000 bit long orime number could be ", largePrime(1000))
```

Discussion

2. Write a brief report of the result (how many different Ks did you try? which prime number you found? etc.)

After running our code a couple of times, on average, we found that we had to try around 33 Ks to get a prime number. In one of such trials, the prime number we found was 683821071315637859074336699 after trying 37 Ks.

Output:

```
=============== RESTART: C:/Users/shree/OneDrive/Desktop/test.py ===============
68382107131563785907433669 9  is possibly a prime number.
[562291892031851935220140676, 632427800662469681745271004, 684036897207529523400778779, 1006649897396987025226857167, 1007953887071893065429444298,
373376688717212936221, 780271887036472734280496606, 742826052091259442292363617, 783189847297109670951076809, 677884243161283758448707385, 80189597
647335097024, 660668461067703544703647358, 848437630340461303038085787, 1054898326540993356161585036, 618704485129683485479879915, 1003347713141937
2103, 934357156154491009397473702, 874510876679301573067815401, 657263031160231495555531945, 810679914649644969340672555, 719048174521816558222406 0
399486433113117600636 87, 824386130240885435885335430, 726593084660536650993001480, 1093952075356750354658078459, 1064909422756787570100333262, 7067
644013848441506, 622557470907629877820065005, 749060676643284107338149759, 719292636528562373341523684, 958407695488711926763983918, 68382107131563
37  tries for K were required to get a possible prime number
37
```

Similarly, another prime number we found was 691271494154740877442208147, which took us about K = 33 tries.

Output:

```
691271494154740877442208147  is possibly a prime number.
[1012110969619105683475959227, 715625579151084471572304031, 893395587128410184108994772, 1028322642082498293636604302, 661163094340701484116502070,
584787052159253187554877135, 812546306156791358759825145, 932048816011932604777641174, 821879825843220702161104514, 657300157204430239871947062, 699
170805929458731903397205, 691070702206624069800732693, 600769640222334000904146842, 753376016072708878816711229, 721101239718510560792477125, 823956
446135574768423953173, 555495150722985594507310029, 785288744703196410541411153, 555089770344650821496106905, 1041850805877281280658225564, 88319318
539659428095469664 3, 830564898721372641112214839, 910451823157829170855169652, 563756698944544563218208664, 775031828486436209884087199, 86733115223
2447151223841325, 922400624251367158427616412, 1077055350065815163584861638, 696520025748428376258930215, 884095838906473191787705324, 8378336299904
72057849315780, 1012178634955734079656653954, 691271494154740877439480037]
33
33
```

Prime number: 904935529919164363069416871 for K = 30

Output:

```
 RESTART: C:/Users/Supra/Dropbox/My PC (DESKTOP-R012JMH)/Desktop/Fall 2021/Cryptology/cryptProj.py
904935529919164363069416871  is possibly a prime number we got after trying  30  tries for K.
[949474083988338090944323336, 1033670412519991586142013384, 880034340554236625165964637, 1007208230137933849324710671, 1087796589020057912894849004, 59123245
4663210436641757356, 615451189332882859108574112, 850029098280317831157444857, 718810760290634786788393836, 604111293252424764818223767, 884426065712973342 75
5621005, 961613639020299989488388020, 1014889852315261969340711882, 715407324694547644639619297, 648938783607797924868900043, 819196335242785863677326732, 86
5354511142196854647912489, 882369845075177067612374565, 588998201378826409350451553, 605454091590900249595375268, 1088298404671985924495248424, 9171873873485
56280597606759, 628397649344002647698882676, 815262871461242301098909013, 894903720585141479319067855, 1047224263645540175226098752, 5714762432767731966575 42
448, 630543777111474684982763681, 984173860537891647515326141, 904935529919164363067839141]
30  tries for K were required to get a possible prime number  904935529919164363069416871
30
```

Since we were dealing with numbers from a large range, there were also instances where we had to try for many Ks before we got a possible prime.

Prime: 783220050771918418626013421 for 417 tries for K.

Output:

```
>>>
 RESTART: C:/Users/Supra/Dropbox/My PC (DESKTOP-R012JMH)/Desktop/Fall 2021/Cryptology/cryptProj.py
783220050771918418626013421  is possibly a prime number we got after trying  417  tries for K.
[554608457009102048869776465, 903070285824610511314192200, 692758425097976162378901877, 894543913073292582499317986, 1059431813540771010918501953, 9261885544
94697844838128482, 853263746363601125779282800, 715732406104963267052771622, 714383514486482541004840771, 615312516259669814400908668, 1065026930210928484278
772544, 957745521813889166420127726, 1036023160449590088961437378, 1009361977935439456325279180, 567955549830414465042418581, 584172300245350979761815337, 65
4823095361115472475834339, 967829829826502622414875287, 623514080999493376982866694, 848042706045174736568645361, 672852321458651247493050453, 71474304924411
4858072259370, 999345473363393940481477133, 864702621517398264242616838, 997311078075292045796042839, 1022626791820062749278568668, 814184418233753424585319 4
48, 1085446402495691610802199050, 910958416801464650473318130, 626991527813552995675682626, 721436827227581859390828973, 906473383587821505790906091, 8966196
077431869181797418 94, 842064642249228374643833438, 1044561849146326981106476381, 1074233591130332145689159254, 897274073326599829720808887, 69873552102311057
2279328130, 1080210229931837685456982321, 863412938785377434066730634, 713212884952620547251976088, 715024377372451623783135184, 571049307185087143681859821,
692714222719864330855534765, 784337034891861939892043991, 1069252229499342448664593680, 574277710285396194653260535, 945577038560135874451228568, 71202825492
9025798308104264, 831564205127952921158093030, 1000894833999205452554191965, 775244635117203225573186733, 790944422450162023101723095, 809864936439590142024 5
36902, 1089179598348645393476488097, 753267135963444248746580564, 839386452705873968797911410, 730479975964001639445077478, 728241748580494105890187077, 1018
781545790578200191691545, 634641323533098055734335197, 599438228392222346263146043, 687682181825294559009848364, 778570617882657054697128902, 832913260407967
080733683798, 747991517802596883024995320, 997903533316091915261673561, 666201097200312352953591802, 852986984253298024196599985, 638591643696834654541253772
, 747351781449142785239239723, 981239640296492075689249890, 593671766166757530108854953, 990771421519292408034886091, 812610501340395735864437595, 8020229740
6786742026109566 9, 885219646561344704280911826, 797656161111535133670159315, 693601605979579925585050715, 1038587446189475588292816867, 79896344488605064343 4
785543, 752331863059037161754025233, 690516325688928414354762256, 740530280662294699229738290, 1091197888037249460356216337, 813280304544712927631288678, 971
470951752140151933133145, 594468766619254237091492237, 991833749708201872275299972, 960928912029674239133451207, 556401684231312706097843923, 791232285740066
913140025150, 977195330012816805989142088, 718665404177197951058177420, 811015483989172031883457940, 830770114866076310925160730, 862255676531919298544021744
, 708812955152534911494863094, 701632745072734027607451936, 620366743650636377651086214, 685164767409164672403867182, 771233146807340589331885741, 6586433506
761946369686729 63, 646654337031439239974342639, 607183336349285325563267557, 612705847563527827027501641, 614477592096984451808808353, 882468244446893213467 3
20699, 1002784794204838950510024226, 977714338805499541996997361, 920045497161314715930706126, 767673503237627392225607808, 972024925061474090799481712, 6281
647622377331910267587 12, 995020634199357225306967932, 899868044530075025635355653, 1010966713101504674938637947, 745012151080340248640435825, 702477992846891
008325498802, 686923671937506062075486172, 759410502068273285756310396, 951776187230901133131377745, 1004254804395264648052084054, 778820574448815783140659 02
1, 894923118284323722815932003, 1073194960485553972842695167, 645516113555699561607874336, 695045851164602323983466908, 1019981311018965721980309382, 1065929
019002293072682389408, 721868532940091985902441809, 884997046554252138419209368, 1079097460644126734765899305, 802245792275967937800690996, 57557397597099110
5385848165, 1012529549876977690795050386, 688288903674669946388364039, 612187584025731523043310360, 695618313660500776159449708, 702944447394646134925225981,
764129048269990558163140409, 908563405676742514582256180, 785680702180603456831861767, 1072668362369288534941144391, 854044121187512573224461288, 78778449259
495960669924049 8, 917214816272895993673100396, 559348482845271964118475491, 579411093799582542574036308, 665416749831995563643020120, 9230309532496258873293 9
5864, 906485316404762007817454500, 591313984696735100738780249, 845169004323880196933962012, 699976314623260529718743778, 627543815153630513910304488, 105028
942195051326822444327 0, 667337758923463591656540564, 663630864456355265170197697, 599744647290361960072935186, 916366049111914823838954093, 66896927290661263
4180495661, 910118017249499410479429552, 953661969357606067509252176, 869729054972480904800942338, 922915313435851676443299127, 943434862097879651844853146,
```

The **largest prime numbers** we found were **28 digits long**: 1082415216200667197411801863 and 1062686915609737331245372373. Other prime numbers we found were:

- 7185145382314668531963489 47
- 5973647738329466656570519 07
- 8534480700624577504037292 73
- 8461475490899196007796627 63
- 7889050994809772970165282 23
- 9800334355445683642899257 39
- 8748699307808324585334960 07
- 8201562690737338879831752 91

When different values for a were tried, the number of tries to get the K that would give the prime number increased significantly.

```
 RESTART: C:\Users\Supra\Dropbox\My PC (DESKTOP-R012JMH)\Desktop\Fall 2021\Cryptology\backup.py
1067082743252194303927281019  is possibly a prime number we got after trying  205832  tries for K.
>>>
 RESTART: C:\Users\Supra\Dropbox\My PC (DESKTOP-R012JMH)\Desktop\Fall 2021\Cryptology\backup.py
721422699455470776325797047  is possibly a prime number we got after trying  11596  tries for K.
>>>
 RESTART: C:\Users\Supra\Dropbox\My PC (DESKTOP-R012JMH)\Desktop\Fall 2021\Cryptology\backup.py
```

3. Using what we discussed in class, talk about the probability of finding a 100-bit long prime number of the form 2310K + n. How many different K's do you expect to try before finding a prime number?

If the number is about 100 bit long (say 2^101), the probability of finding a prime number would be 0.068. We found this probability as follows:
(2 x 3 x 5 x 7 x 11 x 1) / (1 x 2 x 4 x 6 x 10 x lnN) where N = 2^101.

In general, the probability of the Miller-Rabin test passing and the number being a prime is 75%, so there is a 25% chance that we could be wrong even though the number passes the primality test.

4. Is your answer from part 3 consistent with what you saw in practice? Explain why or why not briefly.

Yes, it is consistent with what we saw in practice. When we run the miller rabin's test for different values of a, we end up getting a prime if we try enough values for K. If we take 40 different iterations for a to check for any strong liars, then we end up with a prime, as seen above. The chances of these numbers not being a prime would be (0.25)^40, so this is really small .i.e the chances of getting a large prime using this test is fairly high.

Also, when we ran the code multiple times, we were mostly, if not always, getting prime numbers. The probability of the numbers not being a prime was very low. Therefore, our answer from part 3 is consistent with what was seen in practice.

5. Why do we want to pick a random K and n? Discuss the importance of picking a random K and n in terms of security assuming we are trying to find large prime numbers to use for RSA cryptosystem.

Because we want a secure cryptosystem, it is important to pick a random K and n. If we don't pick random numbers, it will make it easy on those trying to break the cryptosystem as there are only so many prime numbers that can be generated from non-random K and n. Those numbers are a cryptographic key, which decrypts the contents of an encrypted message. Because they are random, it's useless for deciphering other messages. The encryption system is only as strong as your cryptographic key is unpredictable and choosing a random number for K and n will ensure this.

33 tries

```
6912714941547408774422O8147  is possibly a prime number.
[1012110969619105683475959227, 715625579151084471572304031, 893395587128410184108994772, 1028322642082498293636604302, 661163094340701484116502070,
584787052159253187554877135, 812546306156791358759825145, 932048816011932604777641174, 821879825843220702161104514, 657300157204430239871947062, 699
170805929458731903397205, 691070702206624069800732693, 600769640222334000904146842, 753376016072708878816711229, 721101239718510560792477125, 823956
446135574768423953173, 555495150722985594507310029, 785288744703196410541411153, 555089770344650821496106905, 1041850805877281280658225564, 88319318
5396594280954696643, 830564898721372641112214839, 910451823157829170855169652, 563756698944544563218208664, 775031828486436209884087199, 86733115223
2447151223841325, 922400624251367158427616412, 1077055350065815163584861638, 696520025748428376258930215, 884095838906473191787705324, 8378336299904
72057849315780, 1012178634955734079656653954, 691271494154740877439480037]
33
33
```

7 tries

```
1081869127775614489940492741  is possibly a prime number.
[963126762759435388647491605, 702715189886168891422734610, 691273601581664616096448988, 877223413372242072675340680, 758366558716912049971772619, 73044105191
7129337089451111, 1081869127775614489918568531]
7
```

```
391808, 974964166545971246620287392, 1092415504276768266580938895, 1084342224376877164572409520, 946419643379156007161130435, 852558583231815248832042241, 77
555020376789289517561313O, 561738330569687449035363581, 777013354131400275108847876, 686115998387725207140889356, 916793358476894144666080826, 87749573184916
4258455122019, 1093785063368968941589136607, 562547544223983952524173930, 722057396002156804940146316, 627842280195307860731127741, 7980801792293526463449624
27, 642522815695185026242718355, 640752860730095695145648575, 639617820558395559958836534, 1040245022180243894492142272, 654790899442082934561378718, 9862448
82548339288246422974, 990285156045516482182907031, 1064898224185701395065310215, 915350749093403952828485214, 976420880733799825488226995, 992453655689291873
298146134, 851543511112528599867828114, 652664206897819155587380407, 936083350237608321904581738, 819939834921379348524218553, 885216700452311375849625215, 1
031266829226701009009716608, 562308854633440491010957852, 759626165556545715412330067, 884088122041764055791141428, 646055022958538554294626596, 102438084978
7094469871477086, 980205662962775371186197922, 806135311175111452437455856, 875476584964924506105347168, 815926293855580107352175715, 10130214560355369396080
51410, 801759413142168758472766473, 785632225012578418208617018, 731205373884745190648927762, 790065593365593168601359280, 711027675554127228946477250, 73397
3741301356402245536597, 1095405636781115531804835605, 1051957124264323678136705541, 771527035681625192474664914, 854635316461774684217470546, 737226387143128
728212301183, 946074105680009629621490931, 900958713275588938532425655, 965229847524462538741300069, 1080657659965911084209880323, 93579276626629929757215643
6, 946433452520723683463279737, 600441693992089959642573705, 633056680405214748904946466, 861935259318385675095243092, 1063172309753361408163119641, 63244631
5056989536084574626, 766316477739405782717522511, 1032494899022178222880880691, 916877624373550882822295210, 855952974810120937805284469, 1080090003883786385
312022747, 807459746274990055147540450, 660218259079989740844566269, 572684453663506027967291397, 797751729913929302839887641, 895314922358921678374182168, 7
27728107841968734034149947, 928287439539192805318475910, 735633837597431679143038512, 681367416221926723570636950, 564026860788338175653676077, 6990943841666
82156601240431, 874686451844580368101365567, 911838119639612002094280882, 834439867587491653980116219, 1007080140639529821886357199, 672646854246687774781432
657, 841779268074905411580797282, 710116161854161076074911032, 962639013140887331908908318, 928221458062720597820076409, 890448513820868028465901965, 7641436
75289050562032841590, 877950757777817648567541545, 1086079452585400516759131636, 704576577554156227323669988, 689482245626382312945951530, 950726440876850154
818759034, 647294822261629327660328676, 995396392979124715697369336, 741794870673750057850678169, 1055674304514898489954191159, 859579919110330919642540270,
1031611127651414353895502223, 564595635467176068735341881, 854685568969601940577363735, 830379223342609088845774406, 651279634839890514258677283, 84868939139
1374664580517080, 963446298917968884536874029, 1069831807383074416494233244, 785363359866687966887184184, 803628439705670347870087960, 1042764446606972754494
306756, 635284717969435455603500693, 607673560616740849083719440, 855486196882776220049175219, 817398140071351364234083815, 691797006432927008526767024, 1075
340766534660547700226857, 869013481026842418561289809, 998631298200748406668766669, 673891447569646939594290947, 563722314260438267013555841, 760726720437305
438869398080, 1073681618532156425773740285, 792634371543936877849527985, 745725009855761545435480714, 1038262945476498835741180783, 1078957757418014299972928
698, 890204772858325299177523167, 660726016673474076410328708, 826438989160910191865004302, 769084087237662690887826902, 1087056275790541806079103585, 649822
184006722170047680106, 750237867841884655066950404, 564890615073131623479794693, 774687395533773018917558262, 792827648884627062762912577, 100781982169084718
1915762567, 560560275182706289953171106, 906983834695714018081880302, 891271920281118851848010828, 828311983205810835359041907, 924314762360313593328914272,
712640710130566255517077560, 775949891511157612403279762, 700795516339993863974388518, 773248062846485227970998176, 895081956010919219732659093, 805706993766
366004442887093, 1073140967569709399767055360, 1086036367721199091531392778, 981210773362440669450807971, 711740714771446338343531393, 9653637358557073741059
04871, 738507073736490961371469200, 721378466259678734272971818, 903489718679642581184153747, 1073368422726151870125351228, 593884028840824884210095791, 7900
17087028060661454579903, 588644541478287812249887049, 1048064001186695532814576657, 699459637622823544451331812, 844213084586256748899686503, 777325876832899
614953442944, 901681248558449789348908261, 613263850239429750058836131, 868400263498659711903519759, 717146115255943483761278616, 104426006274117085965702158
2, 841492986868120777709935717, 940139077975929235654380727, 749605580463910061396948307, 664218104993390314802634412, 873120843640126255500674766, 958605492
110924287814488840, 763831495993654113676919294, 865081386817239930051122212, 794654844994319217652216288, 898320183365094325532176544, 106610853806227046022
9484386, 746858418752329516083422408, 966109753468205778787554299, 926644863088306117252756267, 969478324936342842644688669, 970828457658285634117950724, 107
5481170083187672228458108, 1085242816018056970863345361, 965781331979151830616589330, 575591644588098598842563216, 943598235015006389373449732, 8810658213212
15794450335435, 797246437959549314496063787, 1088570573079332221982111421, 835912160760948481051523438, 664726510389270223076093821, 992190497331748624292867
222, 783303287589864058331391684, 1044693394852607401147990744, 702047303522337686511225617, 609220953144298858440971046, 678323513467547459183749079, 685780
806499573865220968659]
356  tries for K were required to get a possible prime number
```

Prime no: 683821071315637859074336699, found in 37 tries.

```
=============== RESTART: C:/Users/shree/OneDrive/Desktop/test.py ===============
683821071315637859074336699  is possibly a prime number.
[562291892031851935220140676, 632427800662469681745271004, 684036897207529523400778779, 1006649897396987025226857167, 1007953887071893065429444298,
373376688717212936221, 780271887036472734280496606, 742826052091259442292363617, 783189847297109670951076809, 677884243161283758448707385, 80189597
647335097024, 660668461067703544703647358, 848437630340461303038085787, 1054898326540993356161585036, 618704485129683485479879915, 1003347713141937
2103, 934357156154491009397473702, 874510876679301573067815401, 657263031160231495555531945, 810679914649644969340672555, 7190481745218165582224060
399486433113117600063687, 824386130240885435885335430, 726593084660536650993001480, 1093952075356750354658078459, 1064909422756787570100333262, 7067
644013848441506, 622557470907629877820065005, 749060676643284107338149759, 719292636528562373341523684, 958407695488711926763983918, 68382107131563
37  tries for K were required to get a possible prime number
37
```