MAT447: Project 7
Mathematica Homework
Blake Willoughby
Arizona State University
October 10, 2017

6. Create a Mathematica function that accepts a passage of English plaintext (including punctuation and spaces) as input and returns an integer that represents the plaintext as in question 1. Test your function.

```
alphaToNumRSA[plainT_] := Module[{string, numSet, temp, spacePunctCount, i, j},
  temp = 0;
  string = plainT;
  numSet = ToCharacterCode[string];
  spacePunctCount = 0;
  For[j = 1, j ≤ StringLength[string] - spacePunctCount, j++,
       temp = numSet[[j]];
       If[temp > 64 && temp < 91, numSet[[j]] += 32,];
       If[((temp ≥ 32 && temp ≤ 64) || (temp ≥ 91 && temp ≤ 96) || (temp ≥ 123 && temp ≤ 126)),
            numSet = Drop[numSet, {j}];
            spacePunctCount++;
            j = j - 1;
        ,];
  ];
  numSet = numSet - 96;
  For[i = 1, i ≤ Length[numSet], i++,
   If[numSet[[i]] > 0 && numSet[[i]] < 10, numSet[[i]] = {0, numSet[[i]]},];
   If[numSet[[i]] ≥ 10 && numSet[[i]] < 20, numSet[[i]] = {1, Mod[numSet[[i]], 10]},];
   If[numSet[[i]] ≥ 20 && numSet[[i]] < 27, numSet[[i]] = {2, Mod[numSet[[i]], 10]},];
  ];

  numSet = Flatten[numSet];
  numSet = FromDigits[numSet];
  numSet
 ]
```

```
In[54]:= alphaToNumRSA["aBcD  Ef ."]

Out[54]= 10 203 040 506

In[57]:= alphaToNumRSA["a B c D  e .F ,g H i J k L m N o P q R s T uVwXyZ"]

Out[57]= 102 030 405 060 708 091 011 121 314 151 617 181 920 212 223 242 526

In[52]:= alphaToNumRSA["Hello! Here is a plaintext message converted to an appropriate integer."]

Out[52]= 805 121 215 080 518 050 919 011 612 010 914 200 524 201 305 191 901 070 503 151 422 051 820 050 420 150 ⋱
          114 011 616 181 516 180 901 200 509 142 005 070 518
```

7. Create a Mathematica function that accepts a passage of English plaintext (including punctuation and spaces), and an RSA public key as input, and returns the RSA ciphertext.

```
encryptRSA[pText_, modulus_, exponent_] :=
 Module[{string, n, base, e, m, cipherT, i, j, squares},
  string = pText;
  n = modulus;
  e = exponent;
  m = alphaToNumRSA[string];
  cipherT = 1;
  base = IntegerDigits[e, 2];
  squares = Array[0, Length[base]];
  squares[[Length[base]]] = m;
  For[i = Length[base] - 1, i ≥ 1, i--,
   m = Mod[m^2, n];
   squares[[i]] = m;
  ];
  For[j = 1, j ≤ Length[base], j++,
       If[base[[j]] == 1, cipherT = Mod[cipherT * squares[[j]], n],];
  ];
  cipherT
 ]
```

```
In[267]:= encryptRSA["Abstr", 1 201 027 * 145 723, 1309]
          encryptRSA["acts d", 1 201 027 * 145 723, 1309]
          encryptRSA["ue in t", 1 201 027 * 145 723, 1309]
          encryptRSA["hree w", 1 201 027 * 145 723, 1309]
          encryptRSA["eeks.", 1 201 027 * 145 723, 1309]

Out[267]= 170 860 653 406

Out[268]= 54 684 054 811

Out[269]= 66 355 387 544

Out[270]= 75 364 255 535

Out[271]= 99 781 883 775

In[213]:= encryptRSA["alpha", 1 201 027 * 145 723, 1309]

Out[213]= 138 090 455 815

In[229]:= encryptRSA["alpha",
           26 062 623 684 139 844 921 529 879 266 674 432 197 085 925 380 486 406 416 164 785 191 859 999 628 542 \
            069 361 450 283 931 914 514 618 683 512 198 164 805 919 882 053 057 222 974 116 478 065 095 809 832 377 \
            336 510 711 545 759, 1309]

Out[229]= 20 210 042 213 415 546 381 208 765 082 180 446 650 002 604 801 473 402 708 330 245 871 017 270 223 465 099 \
           172 627 983 968 554 930 189 706 359 691 885 291 295 020 359 501 554 580 019 202 897 384 010 370 952 547 \
           221 505 985 754
```

8. Create a Mathematica function that accepts RSA ciphertext and an RSA public key as input, then returns the plaintext. Use your function to decrypt the RSA ciphertext 102624265650629462932986, which was encrypted using public key (199319989752662759279209,5). Test the limits of your function's ability to decrypt RSA in a reasonable amount of time if only the ciphertext and public key are

known.

```
In[244]:= decryptRSA[cipherText_, modulus_, exponent_] :=
            Module[{cipherT, n, e, d, plainText, p, q, m, base, squares, primes, temp, i, j, k},
             cipherT = cipherText;
             n = modulus;
             e = exponent;
             primes = FactorInteger[n];
             p = primes[[1, 1]];
             q = primes[[2, 1]];
             d = ModularInverse[e, (p - 1) * (q - 1)];
             plainText = 1;
             base = IntegerDigits[d, 2];
             squares = Array[0, Length[base]];
             squares[[Length[base]]] = cipherT;
             For[i = Length[base] - 1, i ≥ 1, i--,
              cipherT = Mod[cipherT^2, n];
              squares[[i]] = cipherT;
             ];
             For[j = 1, j ≤ Length[base], j++,
                  If[base[[j]] == 1, plainText = Mod[plainText * squares[[j]], n],];
             ];
             m = IntegerDigits[plainText];
             If[Mod[Length[m], 2] ⩵ 0, m = Partition[m, 2], m = Partition[m, 2, 2, {-1, -1}, 0]];
             For[k = 1, k ≤ Length[m], k++,
              temp = FromDigits[m[[k]]];
              m[[k]] = FromCharacterCode[temp + 64];
             ];
             m = StringJoin[m];
             m
            ]

          decryptRSA[67 434 590 524, 1 201 027 * 145 723, 1309]

Out[245]= NOON

In[262]:= decryptRSA[170 860 653 406, 1 201 027 * 145 723, 1309]
          decryptRSA[54 684 054 811, 1 201 027 * 145 723, 1309]
          decryptRSA[66 355 387 544, 1 201 027 * 145 723, 1309]
          decryptRSA[75 364 255 535, 1 201 027 * 145 723, 1309]
          decryptRSA[99 781 883 775, 1 201 027 * 145 723, 1309]

Out[262]= ABSTR

Out[263]= ACTSD

Out[264]= UEINT

Out[265]= HREEW

Out[266]= EEKS
```

```
In[209]:= decryptRSA[138 090 455 815, 1 201 027 * 145 723, 1309]

Out[209]= ALPHA
```

Here is the decryption of the encryption given in problem 8.

```
In[211]:= decryptRSA[102 624 265 650 629 462 932 986, 199 319 989 752 662 759 279 209, 5]

Out[211]= RALLYSAREFUN
```

What follows a test of my programs abilities. A 59 digit RSA number was used and encryption was instantaneous and decryption took about 4 minutes.

```
In[234]:= encryptRSA["Rally", 71 641 520 761 751 435 455 133 616 475 667 090 434 063 332 228 247 871 795 429, 5]

Out[234]= 18 954 604 220 209 013 986 888 076 439 495 776 359 853 515 625

In[236]:= decryptRSA[18 954 604 220 209 013 986 888 076 439 495 776 359 853 515 625,
           71 641 520 761 751 435 455 133 616 475 667 090 434 063 332 228 247 871 795 429, 5]

Out[236]= RALLY
```

What follows a test of my programs abilities. A 79 digit RSA number was used and encryption was instantaneous and decryption was never completed. I let my computer try for 2 hours with no luck.

```
In[238]:= encryptRSA["Rally",
           7 293 469 445 285 646 172 092 483 905 177 589 838 606 665 884 410 340 391 954 917 800 303 813 280 275 279,
           5]

Out[238]= 18 954 604 220 209 013 986 888 076 439 495 776 359 853 515 625

In[239]:= decryptRSA[18 954 604 220 209 013 986 888 076 439 495 776 359 853 515 625,
           7 293 469 445 285 646 172 092 483 905 177 589 838 606 665 884 410 340 391 954 917 800 303 813 280 275 279,
           5]

Out[239]= $Aborted
```