

Міністерство освіти, науки, молоді та спорту України

Національний технічний університет України

«Київський політехнічний інститут»

Фізико-технічний інститут

**АСИМЕТРИЧНІ КРИПТОСИСТЕМИ**

**ТА ПРОТОКОЛИ**

**КОМП’ЮТЕРНИЙ ПРАКТИКУМ №2**

**Вивчення криптосистеми RSA та алгоритму електронного підпису; ознайомлення з методами генерації параметрів для асиметричних криптосистем**

**Виконали:**

Студенти групи ФІ-73

Морозюк Анастасія,

Гетьман Дмитро

**Перевірив:**

Деркач О. Г.

Київ 2020

**Мета**

Ознайомлення з тестами перевірки чисел на простоту і методами генерації ключів для асиметричної криптосистеми типу RSA; практичне ознайомлення з системою захисту інформації на основі криптосхеми RSA, організація з використанням цієї системи засекреченого зв'язку й електронного підпису, вивчення протоколу розсилання ключів.

**Завдання**

1. Написати функцію пошуку випадкового простого числа з заданого інтервалу або заданої довжини, використовуючи датчик випадкових чисел та тести перевірки на простоту. В якості датчика випадкових чисел треба використовувати один з генераторів практикуму №1, що показав гарні статистичні властивості. В якості тесту перевірки на простоту рекомендовано використовувати тест Міллера-Рабіна із попередніми пробними діленнями.

2. За допомогою цієї функції згенерувати дві пари простих чисел і довжини щонайменше 256 біт. При цьому пари чисел беруться так, щоб ; *p* і *q* – прості числа для побудови ключів абонента *А*, і – абонента *B*.

3. Написати функцію генерації ключових пар для RSA. Після генерування функція повинна повертати та/або зберігати секретний ключ та відкритий ключ . За допомогою цієї функції побудувати схеми RSA для абонентів *А* і *B* – тобто, створити та зберегти для подальшого використання відкриті ключі , та секретні і .

4. Написати програму шифрування, розшифрування і створення повідомлення з цифровим підписом для абонентів *А* і *B*. Кожна з операцій (шифрування, розшифрування, створення цифрового підпису, перевірка цифрового підпису) повинна бути реалізована окремою процедурою, на вхід до якої повинні подаватись лише ті ключові дані, які необхідні для її виконання.

За допомогою датчика випадкових чисел вибрати відкрите повідомлення *M* і знайти криптограму для абонентів *А* и *B*, перевірити правильність розшифрування. Скласти для *А* і *B* повідомлення з цифровим підписом і перевірити його.

5. За допомогою раніше написаних на попередніх етапах програм організувати роботу протоколу конфіденційного розсилання ключів з підтвердженням справжності по відкритому каналу за допомогою алгоритму RSA. Протоколи роботи кожного учасника (відправника та приймаючого) повинні бути реалізовані у вигляді окремих процедур, на вхід до яких повинні подаватись лише ті ключові дані, які необхідні для виконання. Перевірити роботу програм для випадково обраного ключа .

Кожна з наведених операцій повинна бути реалізована у вигляді окремої процедури, інтерфейс якої повинен приймати лише ті дані, які необхідні для її роботи; наприклад, функція Encrypt(), яка шифрує повідомлення для абонента, повинна приймати на вхід повідомлення та відкритий ключ адресата (і тільки його), повертаючи в якості результату шифротекст. Відповідно, програмний код повинен містити сім високорівневих процедур: GenerateKeyPair(), Encrypt(), Decrypt(), Sign(), Verify(), SendKey(), ReceiveKey().

**Хід роботи**

**1. Згенеровані числа, які не пройшли перевірку**

**Generated numbers for p:**

7C43C33A78726B56385EB4C0687A5C8F139731DEE48695F9AD60435000CA183B

238D060655C535089A72500111327A5AE312DBCDA877187134B78C1B64EBDD67

3D99949BBD7CA671E39A73C0527292917B29E0AD3F86662FE6790F01EF1FF9AC

2893297D4D7B0CFCF2920D5D619CAED1F94BE63CA40ABC613B3E6ACE48AA95D6

727C55AFC93F24DCF1F74797C0997E2B945447A42A6839158AA6539A8752FA16

505728E94254FDF390618E567D388F0FBE58D04CF2984479594D191D8C0E14EE

59BAEBC3841E7B598BC9CFAF46CED090C3EB256C7CD298B4133740A0BA9A80A8

096C83EEB9DC56831CA33D69C35AAD5BB368775E147A12E0A6EC2EAF63F0E511

34A1C572C252EA72FAACDFF0EE947C2E912F38780D48A4DF9F16762485882405

7A27A586B3CEB8F3572F450631F4DBABDA232C511DE0716B1E0DA03DB8CFD2DF

7075ACE6DB161DC471FD8A4A33734047110DE697ABFEBA740C5EEFAD17560087

74B1BA3399D1018F528FE860DADDA2D5F865CD5EC05CD09C8D8323C1421EEC7D

10A653FAF68DA1B3676CB7E88A6C5B583C9216156EE970A74123364B4405D6C5

42C996756D408875BDA224B1633443887B2F0A8232B37A6F99C08A2BD10849D2

6E8CA5857D2CE55FACD9A8930F019C87DD094082C2192849ADC29BE5136F1C40

2C7316E459049FDC0F24BE99EC5560B226E660469D6D957344ED21E8ABC7817C

7F1CEBB7AB182373B50D6E1615DE548E05F466DD086425B6915087C2567CA8DB

6768605DEEEC78A0E3B1D4D060D5D52E413960EA9DB30065ACEADE849118FAEB

1AD552663D89CB022116C326739C5F32D3FA6C0E43E6DAF80B772C37AAD50A2C

3140323FCB61F9995122A159FACEE15CDBAA5A3741E6EC54CD8C241373FA59A0

27B919CDA38D2D8AABAC045310F648AD0996D4A650EF07715642FB31CE18F686

58ACBFC914E5E429C9FAFD9C8980471049C777736BE5F1E0E83F4E3A08F6725D

30F84BCD076196E2AF7D831046564FC263D9131F93F47E9977F0F4105F0E7758

2B3F9C85094BA7F6E0BF7D32D8F7D4D0B47D8438448E1009D97BF90190AD766B

79B8E00CDDBDE691EAF213C05199ED080111B959829AC29C6D16337572DDC720

5AA05E8A763EC1B8050C555C5E5B5DB1ED699DEB50E4FDD3323FF68982A09F78

2BFB4E16B7CA58BAFB3561D961EA7D9A827CB31C15FFDAD2548050A29513DA75

1D2BA25B68A063B09F0F4F243DC3EBD69201184E5426C53E7D875A03699BA612

43ACAA08E884AC901526210201ACD60F559D1A25F208397E5F790862CDA51AF5

383412E5A921BF1916863145986E036F0692782534BF74A933473016F3DBEA51

6B6D1970712EAF98AAF3FBC090552190B086A97DDA346DCA0BEB81FE247C7880

55B8744E2E5CDF0934296DCFF85816B1A167E9158205E9B8BAC474EC452D81BB

45C5E8A83EB37AA9554C2CE9A181A33632E8ABEC97946370C8FB51C7FEC02411

7AA25F789420947583553C30D474909024A6DEFE853BEEB41B339CAC97A02953

158A4C62C33310DF70E2955530A0EB0E4B328EFD4A8117B18C9F5F097562B250

0CCCAF16DFF9B6B97BBD1BF30661B35BD7A0ED5F1B105E519B7ABB5F3CCE3853

334015FC19EDAD470E782D55EE1BE2B8A85CA99F146080E8E0FABA26BFE091D8

4E140FD6B4825AD4F0959505307373598749E4CF95ABD50B0CA258AB2E853713

1ED5E8A6BC55B6CB6D5238980B43ECD7F58523E04A2248E6D453680E9987B5EE

3F91FA10312C3A3477868887657E44EDAEA7E6FC0EF587466D71D1675BB5F254

4C88CC32A1C3C990E8A5BFAA1B6DA47510C99396CD0E9FB14A91E812B8BA7130

2C4321B5327E1214E01C734BAADC0C1FE7B3305FBC18E7CE55AE98B776C404D3

7D4888B04F211605DE6678D1196835CDD6B966BBE853275A07DFF056BE4DCD2E

2A002832EAFD35F3828FC02174D0186D1E09773A10DA583F883A42DFDB7A1E8F

08D0209B4D1933079605998725E6F5B24E567783F4082B03A2CDA42DE113145F

4CE628A0429F4229D68A448FE3DFD374A9833490C82B24DC2F92F263472FA99D

46F622EFEA6951F7D9070C1EB0FCF4AD8D74618FEC94986772D2D9514C54581

6C068A09B0C905A3B2C1AEFDFF6C5CBC4DFA987A0008BC3EF6E101F996144529

201E64025EDBA67710F50ED58CFBCA0606B71567EDAAC838E5C619F13E4E6FBD

3DBE58699DB6B6BA41A6663056CEA142EB3BFF07E4300CB2E5C8BDCC6E70C87B

3347260ECB7D03DF6E45B502D1CE1C80B1EB218B95FCECB8B1E42574939698D9

3C4AB41BA25AB6F2D5099126A8FBCEC5442EBC7DA6AA3F7491F24B98D229602F

7097F4FAD4B0D097691C0EC654D51CF64FD18F525A1AF25A4C2A23F030C00120

3EE1B2BBC8D221202C0834CA2FFC44DC1EE79EEB4E18E45EEF254A14AEFF9BFD

2055FEDC6B7C7642F78DE19199D58F69F57C49D1FB65BC46BFD99C20A679B541

49ECB18948E42D7F2DD39057DE3D9E7DA81A1C45264622EB0042FDC8813BD1E

41B81B361EB598201189FB33757D8D5A18F7E4D68E73A617CC4AC8AC4EBF6B6A

15B40ACA1001643E2A3EA95B5B588B18BCBB3E66A437DF8CDF6339819E7F2788

28337C7B833AE7D5533D460423813DC5EB56A569CBAF0FCFD6AA5EAF21DAD4D8

29B7C4212FFF11AC465B03255D111203F6187C473CB516B7F47285716224693C

4A7E0737B96069D99F3F16FCAC4DEE1AC0BF930EDA27289A37D005D85859F7A5

7F1A1E473CF7FEDFFD9C611F24ABD4A0FCEB0799F3C93DF24ABA8EF20E184479

5126DAA7979E8A7938618C800D7E3BAD180FA880D778A70F3BC1B149FC76C6F0

1827C17745D693AFD128DE5EAECE4A3DFF068F0C7C3A6095392091185D6ED415

39D3D3BEFD122812F14F637568C1BED9171705B1A00429CF22731208444D9B2F

5BFF8462094663BF52F063397FA237514468B925B4A43E3E3A12079B0F2D29FC

4C216CAAB57B7C2F2B3646BE0A2F5678F01B579460827901C44460750717DB3F

0E1698DB884A2E639776EFF56C06BBC5930D45265F309C90AA1064B83956BC1D

59047E53E610FE1265D5E04F00231BA8D15CF164A9C7390DDC7F754FDF266C39

2205463F7F4D4EAA97BEEB19D5CE94E2F6F34009A6FFDB942231D5EF71C64484

0F543238EDD42C011D6FDEB07511FE24B3AED4492CE9263CD7D8516811EA8291

18BC7F35E05F358F51431EB787DFDA21AFD31C5807510B6B9327E9DD0EF60525

0A261860836542F06038C54271BABACDA11766741CF186FA4EEE5F4A67888593

0D3605F1974824EBD51111F46600D39BF0884D6A6CAE9F8A2AC0FE618B38D825

7157FD9CB801D2C2178A6B282498C618E07D578B2533B48B5BE05751FAAF5E1D

5809BA6256DF0A227AA29AEB49DA2403DB61DA01E2A57C3420F8545DF13AE0EA

5ABA0D54DB7BDC7FE73AE1E36EBF63B9FAEFF9011D0DE55D615315F30F8A2AAA

098040D147C76DBC535D98B57431604D5407432932C191D8E51BFE0B164CB5D9

54E69F16513B487B0043F4B30302E1609D625A0197FD9CBB3611B6887568C40A

50DC74704AFE146D09F6311A8883ACE61F21954CCE3D292AE97A8DF81ADEAC14

1694CC9F30111DA0D7459A2B5ECDC6A956641C4868867383A360E37FDE0C17A1

7F63C7FDD4DA18E2D3ED2A5410FFDDFB86D933B00601A04DF02E7F41A34AA2F8

42EE15180799753B6E971857FDB622DCEE483728F3F492D705B3A5F228B690A4

6B8E2A13F9E81C3C8821D144579AC2E7806C5E07FEE96846B8FC0AAC1EB41AA2

7C4059A7A2AB5C5AB83E6CD7DC92918BEDBD7E0F6AA70D356A3FCC373677714

324A2A44584581F67004800AE234D430B31089F03D5B13CB848CA11081490181

5F676679C837CD05650E1F40085B5E3600270E49609BB0D9FC73C21F689727ED

75685BD555C9443ADC8B04D8099677103E45A3487A16E1FB0BE020298CE983F0

5925A26C3F4B2151C529191CF0F9DF6E60EA2F6C9353CB7B6A1FCE1FECD87182

14479B14256629398657E2024173536E829FBBD0355E152B0025E1A16735F579

3079C3EF9F7D4A78DA3147CA664A2B4721AC5E897DD53AC3BB91EE012F2D2198

466212C97BF8E7BC6DCEF7F5A283096AA4BCC18883C7E374F48B4F8CA8F3C313

19F07E04128232948F152BE6217DF0A60999C8DA23FED814FA518F6E8A8DF488

3F177C503139CA0BEE9F15BA31A00C5A2C24298037EEE489CE57EBA1AC21F5F3

1D2A9398A31C1033F27F20849BD6DFA7045FA2E999C353111B67E62F8236DBCE

6744162C4A892604C106BAB07279618D8CA8FB327592D3930EFBBF63AF779A9B

7E40382EB1878A4A3CA4B7DC3326A42CA8D9439D26CFF57EBF90D7987BC0A2B8

7F9D90E5D3DAC4DED3277ADC4E772A1025973B0E8A1375CE9C9516F72A090F84

14CE871FF939162EECB1212C48DA6629DA72039648A1F22FB59ED9DF94B1BBB3

5AC0B142CC976DC57CE2B49016C54EF656A08BEAF59EE33ECB3E7D32665CDEAB

5EBE889926249C8131EFC26E1F496060261CE6706D889F25A35366A63CDB2D0D

69B4DEA6FDA1DF568E44A461FACA9607C87CAB0560AD45D1503A1666474ED074

1678C6BF9D242A38D493A72C3308081475221C620DE089674AB1CB70DD98F5E6

199BF3F7AE61288AEE71CD3EB55CC83B229682F9166F1B00B10A8FDE9E833150

1630C717A82C4D293F54A46D2DFC880F8A33F017DEBB3DA1D134F42415FD679E

63F816757CB2481EFB4FF31D89521F3205B3D3E53C8E7A890DAFA0C67F8110C

3311DEFDEE2978BC6B9E433D37103E727D595C35329EC4EEEC937D4F336B43D9

64B8FA588677372CE89A4A6AB5612B1DC0C3470785B260774CEDAB06003794BF

771C90FCF2C8D7E036E8DDDF1D4AA10B470F003974DD17F5667ACCB62FF9BD60

106828B734816690AB17E772D616D2F73F1AAA22622493CA3A4595BAF8425F08

0BC9676E4389C2906D7CCF864E4FCC1F149FB9AEBDAD6880AFAAF2AF5AC144F9

20DA7BFD80756AC6681007445C4C94E64449320EE554A81BA44DE5918089E8C8

740E3CB1D7DAA32A8CC023486A2E2F19184F96BCD567C3B36DC53FDEE76457C2

49735ED77EA1512308B4A9E7E9EFB5C97436D60F136EC6BCA6219450314D2009

57AD8B13256E67B481DF2B8C61D735ACE78D27864C6FF6109CE5F160599312F9

486A945ADBE3321E904BC0D9B297A0EB07BE053AC9348B9CF7FD04277964AAD0

302454295B9DFD3DDAF8BDD3B3E3CD0B257C26EB2B25C59BB85101D71026B77B

31197C0DE7654C3177E4C7A370EF8C9BD926D7D6A09F50CAFCAD0B602D18C534

615472B5ECC7C6B304F0596606250F755BA51E4E939C218D2CFDFFCA2DC87AC7

69A657930C20E907BD72B81B548547C0AEE7FFB6AD076D9C06BA5AE2A33CB1E6

0F0F91287908BBBFF1F316060A0EFB0F9928A1B3A794DBE128261F834F3BC76B

72A91EE0F2C8D37E7B4CFA396E40ACED930248AA7125F6C895B56F2F4319C93B

19C3F4AA73ED8DE20B583B4EF75BFD99DAB3E804E2DD69C46ECC66C475CDD010

73FA759DACA81B1BE6052FF0AC33E732E2C59FA9283614004358E9BDDF137955

34B2A1FF03FD1956CC84E2795243639DA9BBC16AC178FD290E0FB5763C37CA76

13303031AC417D829CA95E8EDE6A75D0316197CA4FD67D84CB005512BC7B9E82

62EAD5D8DB0AED9D9936C710E9CD6A560D3132D9CBCEED34DCEC23325A2A5A4E

3B95205FD426B0B0704FA8793C0399DA6D1157C92133A0DA8A14E8DD750533E3

37EDD5DC491785E3EB0240D0C9E9877FC92C3A41FF72BAC9039B6952377B1CA6

2004C91F8D39CDEC06B6ADEF9CA7471FEF3E7E3D268E7E34E1DD459AE4C98C54

34A52F233AFC9A3F48453CE699BB7AF210C7E20B1F23CE04ED0A3DA633DB065B

27FF437DCA2FE8441FAC503C34C5E315B99A7D55145A4A79A0DB9D7526D5FA4F

3C464CA3D814D5B0A61C012ECB212FDE0B55660FD24D5A8A2F92F346EABA47F0

655BE9468DC08AA836FC90FFA494B095D86C9AF95CDAE066E6C13EB69AEBD279

6B194F8332767E71D41AA8CF120E8C712B6C2B8316E0B311F32768646B7EC60A

1906C6BD55A3F6E39A9D5B12788E22F98AF4462875EA2C2BCE8A1E875DDEA5D4

28F142648315DD07237C8F8383E0D287D2E7B991F56778DE199D47132D4E583C

31879B5E4845DC5E5B14B25DBCBD321A69A2C06735333E23597D3A9AD40EEE26

41DE31560C8FCC9800B827AB382EFF25FAF5348DE88CB403BFF7190FD0C80142

41154CA171DEE588F273DB4D872B5B5E99C6BEC671D4462EEF8E863C419918CB

61B35F0B1E05946C0D35286735C46065DDEC02CAF185766399F987612DF747DA

2FCA34E2C6E4AB28F9E7187BD342BF68FE7FE76EC89CBF81AEA45DC9E051DF6

0EED24FA49EDC30CACCB263E01B2065F9B27A176BD5959B4D917C893E02DC234

556B11103FD7FC151C04845F96FDC36D01EAB54C52916F3B53D7E648FF855691

56B441D8B4026698D637BFB25EBBA8B4C6B0A89084DA2B469DFAEC78CCF793C1

6ECDA084414C559CEB0B2475695727B47F34DBA0D46A765D38B22F3207BC07F

76C48AD19271F62A3296A511DDF74A02C890B6F9C65C88A39E462F0EE11566E0

616DE70599D93D50B739EE9AA525E4261660FDCAD2DE169852D45727EBF96476

77AF0801DFD622C09A971E2B65153D125A0430E3B5F64C1DA5B90B1A6709CB43

43B66E79055AF64BDF19FEF4D05EB8D06AA720BFD2E9EC604118439A6E1525BB

1C5359224CC7104BE1821C6E8FCEE87812A31D828964F369DD6928924E2441D6

1B90D1CBAAD4005F91EBF9A6DE209912CF00983AC7A08B8FF4F121D02B94F940

1D1A63CE3F71DC0830A6302121A52F47C2159A772D07775DE13C87822E46F895

0C0705204E591252F82AE7755819CDB4D71FCC17E21ADAE7FB59A8F04217AAB8

2078F78BA157F6CB4E8B7D2A5E0FA22DBFD23C267E762545E2D73DDEB5C9B190

1AA3252A18510759798C60AFEB006E6F90AD584B368DCAC69A4A1DBF668AE8BC

7949A1CAF2635574348374B5575BABA78146F5D7B6C781E859370CFE2DA1CF84

228298D9137C5AD22F6E67A0C6F61E7CD0E2F0EA78B5E797BD0122A3914000B8

6F18D4FA3B88C28854799EA369ED3ED760E194976F7BEB957C54824C57C44A63

77AEB1712227EA19C502AAA0FDD5CA4EEC157E87279500D642C9395481E1E930

4E562908B71B64C5EBA09891EACB1449CA686FE899E67C81F21CD483ACF5F185

3C0547286D5F988A5F4A717ECD2B5C4DF93E7431A558223CA333B4BEA77D63E5

34CED0FF99FA9EA3EEE6E49471077FAFE81E657DDB13B0EC291028A8CA483E4C

3C3CAA6DE114A06F2600FC89906B4043BB97899A908E3D22F4331BA76F0C62C5

1F4E26B819BA1B294B2A4DB6D94C80BC6B5527845AA7268CB205608D81726F4A

0879402BD857694732C1D7DD6A6A23A2F86D7C4D66FAE3CF0C052591EB33AF5E

46491D8A91C4CBA4F3E5A79AB7E6E7D2A92D776F380F2FD22AEFE5410AAF8B00

0C225B5D6A5EED78B32FE2AF3DC5AD7BCCFC449B5BF873B3F0DE890E19331594

679F4110E49427A4D0D519EFF9ACFE15D399D15281F1643B13E7FE7F2365CBD8

65AA55F70281B5973E3F06459C26529FA8460338A64EB342858434D78C46437F

3B6CE76EF64D33D27A1B82D77E5B84A291B0AE46E8418E4DF263900F82A752F6

4B89EE2009F0C731BB5867EBE43938B3ABC797664BA544BEE2B8CACBA31577D5

74960920918B13770A44C0F36C78B48250A4E8785A6CC8D8BEB5D860999BBCDA

3F56170747AFA8A2BEC6E76E949416E3D1CA4F17A257F1A952CE8C413EF5E0B2

641A9E4172E61FD0D011C41F82C1FD4AFE3B617E3734E6ED16A71581B2270338

47FA3FCDD54865525B99AAC9079BAC6A3FF26F965281AF9147CAC4A96A6A7170

370FA15505C59628D16142FCEB5682C6B6C161B6E0B6977183DA2FC02E793EC0

430DB433A2BE641C73A670B82492888964747D92D3B47A4284D9BB761B50F4C4

36CE80DC2D857E7AB5E290F226CCEF6ED90C38D6E8A29CBF35AAF287E99C4461

4CBC2F9961F61FFF6B2944B1F293BC06B5C42D3A6255FCD7E08941A3C1462D5E

4169BB6DBEF699EDFA18C98BE111AC0FA90971047E8AD1671303BDD0145DB569

76F5E675C855064C80B87DB1C207B413C9898B0103876427F3D6E781FB31C350

76062893DFC901EC3B57218BB03AAC3527F6F1BC87494C70B4F937C8C413066B

0CF92F96FB4C678537EE200B97BAFD3D971AC555488414AC509826A6E9F8E7B7

2EAD05EB6F78E26DE750E226BC4E9EB1ED41EFDE7820303D7DB3C13824366289

296F695A330CF2520AC74C62E6D1F0062D438908EB528414CEB1A7E8F33425A8

645D9156DA6EF1B6EC0813BDF273D66D5C35A8BD3C9584E99DC562A1C8F3B296

126EAF79FAACD4ED0EBDB96DFC975A7D077FB2BB0B711900483AA02ECCA157F4

7777DC4ECD7283F94B662508FC08193AC95DB79D277292295CFFF48DBA116959

67F10739859B56FA0E7E07E24492C57799DB1BB36C155D28C5983FD3FEFF5FDD

56EA180106AE0E6925872F21CC8B287DFECF79E947EFB6CB2D1D44EF07D250DD

3666C47C158B4BC441A0D58D2E1AC4A264D9F6B3F8FE52F0BDB41C3BB40E1BEE

57377595A564A98890B5E105D73B4BD1D96050B4294F3C8B0A90E85A6471DF5D

1EB71E2936AC5F5070A45EF519A9451F93ED21471BFCA46CB6CE347FB1BEA6EC

0D405D8F5491DE1A2C0A0D75E4166449090158299466015937A7DDD35C97944D

2A845B762A21478E209920E0AE2944BB7250BD3305C543BD5ACC7B65726163F3

43DD7FAA7C7436E8F53A81B6E4492D2D0578026918C7C81358F77D14FAF8C8D1

2201DD4F4C8C27DA72E3069FD805A0CCE447F50AFFAEF596221A014DA7296AA5

7B4B5D6D62010695EDD8060843ED4489F7A8C19CFAF7D93CE2DF91FD7F20B97

4743D3710695709C9D1E7436DD60C00C6C3427A244F82ED185AB7E7D1ABAE333

11E0A653DD35586BCC9DC6396DDAE9E4B54D0151C460EFF2F7B51685149D60BE

6B2256940247638699E225E8F2BFA7D7427CEC8CF8DD5615C05AB302F48C6869

7B407BBB07E3F6910F1B42E58B0406C6F7658E66F6B4D14F7FD88894481F6327

0D8A658C9164D8F98C453694B32ACC0192F99ABCC2F114A6389E14F53E05DF3D

795B3A708F59C6E7D1F50FCCC4BF8BC6315325D0B6F6772DD5AC7D4DBECF6612

7BF86F18EB8E913F303BC3B74FA2300494978075A32DDDEC8F024421012796D9

0B200DE2D013CB982C2AB399DC052092E892F76A1B8AE17CE2272478B59D2EA6

21F06D33CCE37B1E0F20A8F86730113777BE5317FAC56BCD056609B00E0AFB69

4E3B87349ED2651492FC4CDC0EA1914325DA9CE7CF0E2AEA1489CCE46F97635

550027000AC688388822098B8AB91930E911474180ED5580C3179B2EA501F88B

1818F4CB5DB6A9C35D44E48DD47A8917CDEA238C24092AD93853FBC660C5AA9A

5F81B10543AB6B51F7FC6D4819B237E5750916940B3B86021B7999652E645335

68FAA63DCD95F8129946B19046FAA966F55C71D4328393C122A0F12256C70B64

516E4443F4439A516D362F48D386750F741B7962C4201926F1ED7EF256D54F3B

10C8D42BA6F538C23DD3B517CACD0DD58D139B3F815447A177D3FDF5F19E64BE

3ECE3AC9B4435B711A370772C6DCEB4EEDCA1B93AD1EA10580474605E26A965E

45293885FFE7D2C3E69742C3191B85373A4AAE1ACD3563F1C5158B23D4FD8B9B

33CC3B5936F51EE9CD8C8CE7C58E1EBF22408F14BA81D751DB79958E9409F0C8

5F61738FBC6F68345BB9E2FCE388BCB59A64FFEA4C35978278B33AE33BA7DA48

13236CE2ECC3D98A125992C62659E7ECBF7C2ED8475830DCDD491D4E492DB76D

6A9354A2193D0AC78D71F549ADD242547B67E42CFA4B1B650B4E28D986E0D11A

4CFEEF9700DDE69DAB56366C9EC39E5DACD7BF687A05E024F500A94D77152E0D

31ECD840DA581DE25228F76B1B1AEC8FC50175D73EFD5191F771213E2C00B007

19106DC871BA863EE63C14E6BF25823941A4FEA0591525008C328E91E41C4E1A

16A2CD6500D5893BEC0E1A8C9744612931B9FC93DA5FBDD679D6CF47D1060530

5A344EC57B0226F71B19965566BEBB8C646471D85C76EB4B8F269F7433662A72

358FF0448491D1B7B7E19731CA2475D899BB76F80C72AFF75DFA7FA3E54F49CB

7A32896DF934E1210106893BB5BC7DDAE133378BDB5A49BB66FA441174C22BE5

731CAF587DE0043FABEDDCAF3F43D9EAC7F5907767F911BCF9BC14CBD2790DEB

41EA610A321927CEC0C814EE03B4A9BCD4AAFCDCA6584BD827E01A70BF0A48F

314733E33222B1A9AB85C4B1E92A97ED0A3E3FA6C6727EB6DEC96CE943B84582

0F4D308B14CE456ED0F22B4EB2F3830AFD96A574C695A183CCF2E6891D79E10E

6A4E71FC56EE5126FB7F17F327893DD0057C2456AA37E4680D4C271FDDE45615

3B0A7A0863C3A6D03E06BF53C0737BB65937D9580F5599791408F49198B92E9

4124B78722DD01D1A1927A859C022D4E5201038235A84D2F7C9B305D0D91CFBB

70017EEBD2DD51A8395AEEBA7C95DB1FF24495A95EE66FC7C0870AD5DAB9BC13

7D2A4570A8A520408D93287DD50EC47B8400CE582A01AC689692905BB5152D08

383DB72842744F10D526BC6F73EA91BE53F43C1086C725FD27FBBB08652EF961

285AEA88EEC3F56930062A16996463EEDB63A2570E519FC0CEC04279D1FA01DC

1B2C997292DA9EAD2F8C75C4C690096F9D048E98AC0AE9FCC2FA047F18CB43F3

7588978AFF2916A11B672928507DC5389DF76B86313A0ABD8A9C9A302640EC87

6900D031E9F17873BE15AFAB44A3E6B69B1EC2973F708C341591D5F4CD30F3A5

600CF8AFF3B4DFEC25497AB18CAC2A0A204DEC9386C876BB7E63F1BB25296569

44D73F32855EF5AD95584B3BDD0E9B99D9666C7FBDE10284F7688F35B4BDB392

49A24F8B995E61E69A6BEDDDE8C8140E70496288FD3C8B2E0FC980FA3031B78B

3468C4A71D051E47CB4FB9907E0FB565CF9E3465620767C37FEE3051644E73BE

78A2F109391C21E34BA7F4BE4D5DD066B8178D66BEB62FB35F705C535B98701A

60EDD488747D09AB6CBE1600C9031EB0828CB6B17D75D9C9600BC8B7FBE20060

0F1F3A0037D6C7DE662CCBFBDC145C9E60933DB61DBCC5D49E80566EC52EBB9D

1077BB48CE58685E38B6326E20C47295FB45D95A633DA139EA30E0987DDDB4CC

3C919BE9A9E66B43D36166B20B32FBA2FEF3432010EF8BBA492253BFF317BF13

3DFD51E2032333F7ABC7CCCC77D41B80003ECB5486CB9E80D05539CEA985582A

7C6D54614B09C31AA522E8347ADC3EE6F6AB506ABE7F0D86680D54EDCCADC030

3567D8914A5E056B9EF9BFFD3C4D259D58FA1A93ABB11127CF8606BB433CC4BB

7AAEC592A8975C386B35802FDE47D71CF898E8285BBB0C75A9FD08634B4FD695

495D494150D599942567054134AFA2E5DB4F009D75FBD318578DC4DA615904E

7F90EE34DD98757DAA9D4846E26EB06BC90256FC25B82D59674899B1EAA50E53

6BC53A275428EF6170031212AD84B48110950E7E3881AD541FEB4C4968939F00

0C19ED6F33EB234F0116ADF4300B2A6850A7D8E2B023130760F5E76BC589CD53

2F4E7C16430F24C2EF9D9B1B20DAADC9EA16597581CDD1D48509611E0368004B

6C9D6F1F73A7250DE06EBA7F80BE56DF3DFA3238EF29A966049F357A279757C1

6BD3AFB032E2BF2A87BC49F4690CB31486DCA28EBAE4D5983C0C788509154851

0A1A2054B04F2C218D7579ABFB7F12D252A8DEE5F58F21C758B0EDCB1388B6C8

370D86C399DF9147AEEC8551E50CD796A0CE152259F88934056570F9C708E809

266594108885D7F514B844E6BF93BE91371670A7A6CB3763A675C987EBA9C9D

27D67C5370365152CADF27B5F9E69B6B13F4AE1721E7FC525290344E27F7C56A

59C1D4CA1011ACDF689BF36717E22A415216920A2DF1ED16BBCECB446230FE7A

5E52B269BBA193A7E3C51AD045FEAB8622C79EA087886B159D46A32CFBDC123C

3E6BB059981542C6BCF446D2683CFBC632E7CFC1B87771D9C6921664BA15EDEF

111589AD512EA8EEC4A5A243E7CB732E68AF9937736822EB0B8A21FC7E10238A

48AE11DB5B8C97CAB0CB5B0A38D3139C09DBAFF91CD8E3CDB1EA549739C3188B

284738791FF05449ABCE511BF229DC505A917B7FA783CA4F1827E616119B043B

0AAF27B015F69EDC331ECC51472DA33766208E5CF1BD0AC6DCCB8FA4872C418C

39D22B22E6190B7586F9D42231B325DE77694EC9E9FE79BD7CD60A5E425E151

27EDD472B8D6C5AA67F2427193F1984582D1A5E2D40C83D8E571749999BE8DE2

5EFC74A9C1CB03450D892A16F259519EBFEA1C9448BAC56EB0DE55C3D9F46787

3A8E16734AB80FCFC3B3013629BC39940D221908375E4BE3946E16420C48B8CD

381C4529A8D44D250E5BC96334722AED0331B7C0FAAEEAA6C6E40D59A20CC17A

55F30EC71CEBA2D802722013775D6D13EA9D71AA4577761FBB2C22A844799F15

4570E71652DD304A97C7977FDD37EE59509BAD88E2234E2B6E571C275F74F359

0E6C6662422AF8016F22840C132B7B0FA9D7285E8D77214E2BC84BEFBBD4E6E8

2214E6AA332699261FDA4CD3580CC3131C6BE09045B2C32770E320696A4D1C3A

19B8163CC5DF5AC36FBE3276FDD4F7AE43F87419A3F2E4FA33B59A36A4422573

6D3F545E6A72281CFC631CB832EC92A31F94C65F023C42BF02A85B965FF1F7D4

2E179C2D8AC17EBED42635D2664C57459CAC698058E80E188DCE3CD9FF088EF3

1237A879116C38E825FBDA98615AA037260047F48C5C69331CA258239B55E324

098DFDEC6CF59105D0F8B2DE357F810C18EDCF418DF69CDE6E2BFFDB1AC93A4

2DE3644DFB35768B5AF64E45D0A6636B24F1495337F9835CBD0968D8BB84AE73

0DDC7EAD9D5552BDB939E1026310C1A7C2A5294B464B926A4115E0FACF8F137

194CF2B09B73B133B87E6CC8BD8500451CC45F4F48562CC7E0D7E01348870A14

44C46BF019EEEE4F2C27BDF817A611F6417E1B887253FAE04A81CB0A1DD765F6

0986B43C49836B62395DE7E76CEC052E2F3C6F499CC51B02872E75B8B3A3B382

61A7F2C9D0E682A798361C3D18915E4E0672F6E4927A540CB548D1F427FDC009

697C7E743013556950958D2A8614E46ABDF5DA49DAB61E646AAD223876283A1

5EAA25E56E6490F0DC4AC6CBD0A984141073F975946C7D39DA5E7A65B29E7657

397862F7210D6ADC98AE77B146974DCB74DC4D1F5E4E76F3683A04590BE29473

30C3A250B4524829B880041CA33426422CB6442040910D4A18DF183DB085026D

4DC36D35F526BADF72DBB98B6ABC5B124DB7E4CC1BDE83D76D59AC188DF326CF

5B5CD475672C9CAB57A75AAECC05478C665E373E8F372E665F90DDC4C4AB9304

1EE001291956D40E5AB9B675F0506477633F141B1A394DBB32AEE1C1B4384080

603F4BD915A3A245A0AC94FC0A98AF2D9E082379A070125759A9386C35110F93

5F94D017C2A323B05C21C5FD92E90C909BB342A82EA5E1CFE903DA18172BACA9

4232B7B377303CB9821D9D8C1F9EB266B1145C5C24ABF1189999E4DFD0B51F3A

29095F03644D078D9B9FF3373764B4D2DE57081A472A0B602E936A58BF0FDECF

777541983C8B6DF989888BD816B96308A402F7E7CD37152B8917FD4AFFF25A71

5254C9EB6253095D52D31E7D312648346EE5B3C66077E01DE726604641383B79

4D729ED49DCAF86C771C87A71FCD99F5E6C3D972E37D2305B572AD7B9A0C48F5

37F8D5EECC3944BF010CA93BE8BFAD390557CE5C6DBE026EBF207C8F46A5DA3D

347E2FB092117AC2420BD9B9F557218CC7D7126C5ED6A4AD62674806B5FD8C33

2E8CEA4FF4B729755CBC07319225E8A950DB1594C8A1F46A71405E9B03113A3A

17E9E8D0CF9B3C6C68F9B5F9291E558DFB371B75105A16AA8EF72A2580D5BA44

5BB7315B46D7809548BD15787570ECD79856EF0F08CBA50B716E3F4048501D41

5FC333F05D10A5857F13E54866E8BF44795B3F6C099C55007D3CE576F2200199

619E32213EE26144417F3AA59F55A04159855125DA83067326891AD2AB862B41

330BBDC8703E7EE728A153D5D7D885952CCCF36B1D87DD5DD4061B69360FA5A7

0FE2A049AFEE615307F12E4BC1C2F80BE82109013A5B2308A047F08DE7A49820

3E60014ED7AA10E49BA1388FEEE6B6A44F42191554435E70CBF81B4FEEE96259

6E27770F7F76DAB1417BF6268BD43524218834549F612841224374951CBAA8A7

7B470F4279F2338176282D7E5E9465E8553B9826F1F9FD475318AA45D04D8442

12C54DB3DEB4CE32B242486B8E73E0545DF1D7185DB44EDB3FAC2B61F6A4C60A

7F8FE42CF079451CFCE215F541F56A7E04E2AC1AFDFBC5FAC131D3187EB64E14

2A8EF1A179A4B24F6A62119D8B2329801DBBD3675C3ACC47C1C427015309FD9A

6D6A9B80EA272B717A141F249DEC7D99E66409074FA68D571CD6F265DA7FC78D

36995B56DD09E972BF2429FB6A4D2F2C5C2B17E8B1BA16FF91A6752AF77A2140

5F1B36A07DDD96EFC941B8F827D549A957FA0AB9E419A892A030C1056D9F36AB

34965FF1F3F36E4BDDC750210BEABAD41D662C2976B3FC769985A61E4857F1A1

659E2426A6B99DA0E7636F8615BC13A7E1A12B61D387A9B206B6A9C638E1CC34

306BFBD1285204A610BBBB946C22F4AF7C60E69F61C72EADB00FEBD4E76220F2

1EEB1E65D1E6F5705BB4375CB910682AF18C39DF8754ABE1FDD13F4786DE3B7C

2709DF2953CBECE531D67082065CE988222279CB5BD8C529BE75FA34B1A2F7A8

63BEA492EDE250AF993528B5F2227DD9B1570287F7A812B34E760C8CE2416196

5AA76242CD559034F36CB2A08BB23496515E6DBAA7326110611555036FC666C2

578E3CAC460464FFFD06E59E7157CBAF2BB23DA2C9025549A161EDE6F63C8186

46E4921AB2B5E1DF54D9E5F60B28568DAC65466E721A02992E30C2D45C62D376

2CEA183901C1EF560DA4AA92F5CBF081C4B7E81E28A2E91AD3FDC2DAF92EF9AF

0D0DF138DF0942D6EE9BB301D0A0216BF9D4D1BA4EF9D353EAC2D5D9F2EB9658

0AABAB770EAF607329219C64EACDCFE09F0129067D921AAABCB4E2D9D0BDBE0E

0E2FA83CB90A5BAC291FA8ECFEE46C29EC89FD106C393005B452CA52B2311B14

4591110FB49A00753F3CA3CB2EFCDA14DB378057F1B47DAE571F099EC1455A7

604F4E5D80FB25EE02345279C39592FCD51D7B7BA5767D73950700ED570E0C18

173689706865EE457F5507EF891B099BC6B4FEF31BEA20CECAD51192A5E6C79

572D733A733B1197A25EDADAC998363A50CB0A0238350CA4C03C9184899B978

3C7629C5EA652335DB50435FBF2B692644DED50D082024B6D20B7109A1DD361C

45F1087014CDCDD341C5114BDBDE92827308684BFB6B78C86EA28EEE6B2C2EE5

65746C82F553B0712F8F492C18F0C1676745AB5AC4B0CB5076E7EE27DE060247

37737EEABB1F83F9347D9259F46177CD16597539111B79CAB32361BE0F18B46E

3AE37CE922E6A26F0150728228013F2136AFC4C9BAD75DDD292BE02BC8C367BE

6309A08E6CC5F11AEA698ED4E43F8034E60E504934BE89C5BB6D38C227351BCA

2B38DB7C57AB52C136834C335860DED557B1DD991B6A77FCE28A3102B53B2E3F

5A6244B17019EEF951DD66D440C3594BC8C1793EC4C3919B87E5CCE7BE32515

6881B0D5881099A9FAD4B4F91268F9914C5A20EF641D567D7D3D03E1CC29FE36

0F5B785634C3830F9C1EF429139D73BB759245CACCE3234A06A704361B322722

380BBB2BE6D47D353F30AD5116476C87F40C09D93E1373120662F29450F48090

7425B71AAAEB5015E65EC4EE379E5050EA0264AF9011B652B4906F4F84B48399

503882AF4159F1DFDEA8A3C00344DA49385C950A082F4706C59F8E193C1DA2CC

4D62079EC39F0DE056F2CA20AD00F2F58712544BA3FA43D8BB96A24179719035

773D96DF7DA94F74FBE4DA2388A9A9CD015B8FB1B7F3ECC6DA41BBEEC6B10BFC

26E0CFB16FB8B10860EAA647AAA59129A44C3C7749245486AE4E44F477992B48

4BB774D3721A980C3115D7AA5E6C326C34EF8615EEE4FF2B1ADD313083D26006

73A10209803294E4B265577FCE955792E29BFC8E0E615E52E2A341EC585D6AC4

4F5AA3628121C52BA2C512E00FA0558C235A551DDFF39D0079061496DC0647C9

599F1163053F02D70644C318C8E6BAF09EC6CE1F543317D55ED2CD44865051F

3C9B95057552B143BEDBD4117A47734E334A6662B1BB7AD07F1A8CF7AF64605B

6F9FB5190EC2EE266904A4384514A95DD7309C11109619E711F07B4D1D0277F5

1FE4B7AE8CAF3C8DBBD1EE72AC606F8CDD2ECC3EB2DD202A1DC1933E462C57B2

2CA27E54EA2F88072A487D9EABFA8B68E9C8146029A0C798099C7673AC01B4A1

79F05623724B4D038C4CC44E8617208476F02D468E40E4ED69BA9575120D37DE

49CB74028A9CCC4DFABA97DDD00E26C42DFEAADF949D6C8FCB9A97775C5F5F05

4CC090D3AD19308560C0DD3302D734255F03F89595CF87239C773C12BB1977C9

1B289755C9844F52905FBCFA71ECC177B3CC24B9676B41274C65B403AA3DEDBE

50029383C626DC9D12AFB7C28F42C838906F5BD5F97364D7BB62E397D7568A8D

4886C876648822DBDB1E70A99D80BDE338329A7E88BB338BCB9A876F73C833F6

34F68A41EA00AF53680CCD5CB0F337F0CF07EA6410BBFFAFF363D769ED9DE0B4

4213EC7A81F9704271E9A9F0EA6CD4BA0437D39A1833DF6FD096031E96F090FA

7CBDCA9E9799C593220A9FCDF5C268E6F2F5F56477847280BF1AB1E7D0975E58

0C538F7942138596C8BA9856D636C67C4CB855E67410E046EA7672B27CA9D916

1C287E9B2C71C29C1C8B0C2EE72D823A3FA7F2D5FC74DE5E792AB3E42E755862

7D7F3B5DAFF40EFB8BF1F3E9A1E4FFC6642B0592E65BCA25A704F21307ED3A0C

468A9AD8666C8CF2FA56FEA53482C5B70242B2B86F0ECCB60E408A34E54E1604

1DC95F01400692FA2411A7F1FA8534DF649CFF6572AC9D1AA84DE7308A377666

59A1365ABE650CE8F58272D9BEA4BF6793831E3747A0F37AA7CEC285BE6031D7

6EFCDDBF2471A2A17383CF99E50EB0F9DE8C27500410CCE34F600DA99F04B85A

56A3642359CA765044B4F06F4C9B7D38204EF89C7CDF1D8FA87442A0A92A5549

3C682A3E625107B7C7B18C66267E6F639E41492D24AD5CD2F6BD46E39E4DE461

71A4FFDB8A60322D7CECD8E496786CD4FDFD93ED23E97777D0E9A6265DA3E2E3

55BC4826A69D742024EC1F738AEDC281B1F78D29B4650B2DCAF5160E6BBE6542

34219BDCF7E99EC33313B2FBC32C89267B9F04EA586882A314382328E712C88A

4C22E2E14117A43B5FB374C228588C3E6A7262959ED49DE83607A8F6E219C09C

779C589BA15895FD2533F40AF4DAC3B524E484205A3F55AEDAAF61921AD0CF56

32669EA2CBA0A96889FDA40D3CD2A9465C0A4E52F62D8AF2C5D56C74A5EF68A3

788F64298D243C3BDA69C6A07DD1F83191B8E4A547A74998510AFB08F61B5A43

4050575DA693D6D93DFD9527596CF6E8CD26AF055D465652B7FE05982DB13A78

4EC26436CCD741891B6C9C6C3D87061BAFDA0A70AFB64CCC58BA70968E38297C

369E249B84327AEF890CF533E0BFC73B8168E4C93727A6CDD7755800C8A47B87

3CF86ADE5B3976CD4B4CA848A1FB649C93DB58CFEDB68337C6E838B71B02D36F

34EE98AFB8CAD1C4DBBA1C980C9B5EA439C84DC293E4E19DF7C44BB35EAD8D89

10E442B00FD3D41C457D5F38C020F28DC49805713B8D1B4D27271B1E7F5C3A2B

6BDE6B00360803CDAA90480200FB4DA8B897660EA5A01D2F7462E52930554B

1F9F11E473D2D4B03BA5DEBFB2F9AF641D80DBC93FB98372174816E53455EA4F

34311104395AFDCADA9025A014680F686904854A5F3AF6A74E05F2BEE836CD97

7FB684C74E5E22348B9053F2F065600473F08176ADDD4553E159EB2138F3E141

13AFB13359302F8E6D1D407AB4EF89BE58E744BD0F9207469BDAC4C957F3585B

1EE8EE7EC60C39550028899F5DE56468ECB3D3355981E8E73B30B87D8D1369C5

6C18CC6B12B3CD0E400DE284E23EECB86BE4F2097050D9F065C2AD47950BAF7F

1731DAB3DBF3A3BB23A50279766CBB97D51D8170B8B495A65C8CF6B7313D5F00

35909B3FF2C0CBF2CA386FC70FA41C3338C8EBC5FDB68C417AFF2416B4DB5FFB

29938AFACFAD9CC6E421FDFF114ABB46DDAB70E275704BE27F0E35CC5F84CF1E

3B33A958538250D6963CCCC672CA9AB02528366B9B0783F727F9D5CC81A68689

7410BC38B605B6A8F2906AA128F95D63F4EE52ADDC4DF14F911EB93B03D45571

1866618CDBAC5C1E041292CADCA11C1F432E3FF71639B594C7BFCA6DA8935380

6212143A68F1E8C75AC4FF39277666F668D77F7D134586495357159BC199349F

3C706A7AE4812B5E700A4FB59F9176D77D2C14180C0AE507272EF7DAE9FB1DAA

6F752B20EA46A8F589DCFAA1E30D0FCFBCCF7D8DD0D00DD50DD06DD73EA68470

18516D26ACD2D1A0AC29E16403188EC38B4361F2FF4CDE377E259629092E4B63

0BA3DDA7BFA4F00FE0A7B19A05A54DFEBD720AB4853A0977313AD9DB9FC6F48E

6AC81FE4240AEA86A3470D05218B6D71B087A4D23D6FAB2D91BF1A0071A0C637

78D969B36AFD83DD3FB3A9415EFD42C622263E0A85ED2F1A00210D7D7038501F

2F36A6DF8809D58D034943CC8106EA158E606311BF3675718EBC766C908FF206

78FA7FDF57851D01D5CEC661FF8C435DE7C4BD3F33008854CF93AE763F638986

2EA1A59619AB7CF47E7F59D8F94356F07EA5605F88E87D37F7B705BE70F0DE35

760544343184FB7BFE29D1D4D6C91592AC80A30E8D08C7F83B2287443F9BA13A

3DB6E48C6CAB1D6C7B3F6DD0B5F98285B7AD4DD79A4D1BC68E55E5849A788AE9

1E84C18AF12843C4A91BA5E5084146C5AAFAB0DE676AAC317440E48F78AFD19F

230FF8EB8170EE6DBD6286970A374428369C5609E60947E207F3881356F011B9

4B56EA2ABACA8FC82405C096D16C1E273652562AE3F082AAFEDFBAE49A5B846A

37713C092004927A9EFF87D79A058FF81C65E98E09947095699E6CC46D7A1AEA

7DB1682BE7B15A5B6754A9FE539D6BA8105087AEFFBDE4E410A5C745B6DF1830

370FF63671D922EF9199BA75B2425CEA3E2E52B5D6DF0CE6367E2E7891D9A337

737224017EB6833C7EFB6E4B562C4A71B2C2528B9F16EC5414030A7C66907850

180DA208E97B93EAF51E17D122851F7E23F8F6C9F3D9F369519A89446D6FCB86

25B51ADC0BF4D05C91B66B248C07617E1B017F0A311A168A1F11A069A63B0420

50C0219F09F21D43442BF9842EC571F46881E6D35992921381DD321DF1C337DF

0A313853A2E8BAF6D85478F2A8D427E492F76FA8860BD4AE02422ED92BA9B7E6

555BBE9A7138A1963E88ED8D94CF9589C3D36607B85F91515A0E702ABBFE0EF

7B38F98F1459AA5622161E52A896F9AAA7C6E85528CF28430801A94E25C03B7F

388593E48A328AAF50B7D8625FA682A67001643782C82F83DCD17A0A75320

75ABA36505216B18C0CCC3E129B78E4FC8A7367F283B37227866F2122D2BE076

633617020C307593D0D5DEEE7E04F68AEE6C0369114B043B1998F03B67AF71D1

4C3C64D6EC42FDFBED9031A2BEE6C82CA885C9B99C0BA8A9780752D6BAFFBFF9

536CB03F55F00C14529F8C6265942028294264F85379326608776D772306ECF9

113A2B20953984D0F492C91FB33E8C5502DFDA35F194581D85E8193105E600E

7C8739BFCDE09C2F0012305CBB3B7D2F0BF9DA42E9B047B2ACC4D02FFDAC2494

567FA9820B1FE9FB5331CF6800B3F518EF91324AB03AB9CA99D2B878B57F3116

0FFEB31CB0A471C0E25697039EEA4E5F6726504BC2CFAD24D299C45E37F10591

0A305A1FE82991561282582DDA0594423656027BD61EC3DF718964A5567A09BA

3A4277C7A6C17771648400FB2FFA9F8A2B18C5C29A3EFFBD887A2A3021430286

5FB7EEE95778A7A569CEF7419C8FDEB3993AE380E5E2FCD022BCECD7152FCCD6

5351F7196F6D25944D4F7323E55C0D9DE98DB60607683B6F720EDB2E59FE5E1B

5A24C8E2B47A421C826BEA0D74F74D2A9728984A944C62BA82EF8FF1FD7D348B

7E6AC328FA27E33AF96F2F3BA8347C189E92E87114FB4FB5974D28033A3CBB51

110BF3BCC3B3F262EDEADFD991F5510FEB0ED7DB012DD201DF7425F2185E4BBA

554AFED57C7C74833C97B225BED4165BB5B101ED054320AA6E7A79C54F2E7A2D

5E6E0E2F3BFAB489AFAEEE6C4E3ACA2B546BC70A1FFB297118090819737707CB

**Generated numbers for q:**

572F49D2A6A1D07C4E15C4D3D327A2E2A5D3125A5A7339CDEE9A0859405644F4

47F14054E4C58A0BE97C4915E1F84D7E409E465AD8B2219C71C42E549373BB7

7B85F853B7FB73AE4C273B1F94CCF1FAE3E7BC307FD3D06C87BB48D7A27B0539

24F1CDDADCC783491AAB9E47C9FDDA5FAB3BF5B9D9105141B31CC4005387DA12

1C8030932B20BEC37A424DD099DBF7A78081951C5D531B73870C348ED7645E1

45B7A5F2075C843D3B8362597FF90CDBA2080A013C67119347A155B35D2F0DD6

19B35045AE8BC58071C1288F01DAFBE3D2CD3CC2671C5F61BD9FDA9A11494563

259B691EB218088306656BC6FDFDD37285A6598D66F8F912105ECBF67D71BEEB

7000B744237B08A5A455F2BF171AA413B934FC8A10ED80A70D6D109BF6215501

66E36E341E6D3454FFD8D9F453B9184B4DE4F6DC08D7B20EA94EDF21A9E1C143

2C4B1B30F306D19992EC7B27A318CCC674607BBA84FEAAECD68F48C1E974D55A

68736A2925C62E9FC3F7DAFDF1D1A410248051DC5C0921E0EBFDEB837206A804

336C4FCC985387DD818E970C7603D034E0BCB0B012A367CA96D07CC4537335A2

4DEB94F3DF90EE6FFCC3BDBBAC9BBFD9EFF320DF641DBAD26AB76B06F226BC16

21737BDC13C2D7C2E9DF35379997E6ACB9A22D1D59A4C5570F7E6EE00A0C0A77

3050B8061A2D72E185FE42320A27CF50F670A31EAA1701C2CD96C71285C544D5

6AE539F5EEADF17CEA04C88CE86197862111F38437E95F09B553AB19C30D342B

257E862DFA72C564A1E0111B04986E5CD43E0C202EE6C1AC84AD474A8B0D352F

49896B4EA1077C9FB04C1EE043EEC057FE0E022A6759A55B436CA4523476D092

34D51B4A8F4EF45964956CF1A5AAD830EC8C2B9383CD360E47A842E9C68552B0

6651378CEC49455B01D8F7A27BA8E4D9461889AAED145E2D517A8C7CB1EF919E

251D1D4ADEAF817DE4E4EABA3F902AC36366ED3EAAC559BAE63065318B071F79

5978DE5658FEF7B4AF1052FFC07B3F20E5C2260CB82DC5A9D82F560CFEE621A5

0C903179798D99210D7CE3DC49591C4F1000E97B223A28D64D5B7C26CBC1EC75

4D8489833A3053097527E8A69D05DE3EF986F374A7EE1D3AA50152935AEBAAEA

7AD79654477478817755D5BB0E019CE5245AB2093115C4B8C465DD07E5978700

294919D8288BB27A37C8609988F26771C30B3B50A3303AB88D0BF3B39FCE2247

1B715A48C884C9EC661EA5E728866008B4561F47B6E88F9721F7F1B625C05A15

686EE97869D3E5CB41080D8B723ABB6ACB51FAB5867C456C83DFD846D39F43AB

3B1A1023922C3F8E9D8B6D0E445435D2D82339E3C8FB313E755482A07012BD44

4BE23EF5CCEBD5A92E121399C71D20B95BFE10AF0C693B962F0BCF1762C9AD33

2717EDAE5CE9E98A9DF926B9B6A10B3AA97FB670C7E62721D8C0EFCEEAD3B0B9

622E20D5D267A771375220B317F03A0283561790ACE53EFD8058E343A05ACEB9

6EB16BFD325C93773534FD1ED811089663B18FA3C2907DB572D488FAD19E0FC3

446F6148934457A765BB87D374C43961B40CFE32D634DB24B8000FD5A5904697

293099639A6E9852DDD300966D6C24516C16301116BAE8A7427FABCC57F8FF36

1A9BA6016A634492A500F05B8444B0630AF267E1B85DE8A2FCA7C8EA0EE1E314

1D904F89D111C4A02BE38560FE1B515112C9F921115A06FC6D9B8029E01F6142

158DF4054E6240A9DA0282F6E00197EAD3E03B8B7A381C1A34B9FD73CD40FFCF

523862A98F241B6E470EE2EDDE3FB0A21DA05FFD7CDD503D02E2DF00C13906B

28D55B58E2398CC6FB5A3C9AFD04598DE19B08D4F1164A5A62EDDFAB247D1A67

0D31E127D3C7E5F21F067A43E5A126CFEA1B85389198FD060212D92161B0AC8C

5D40D60A2FF3B4209D82A37AD7D5A8E93CB2897F24199FF1820A2EC0606BA408

28D3D2118439A1B9B514C1091E349FBF992DC38D617F6800417E26FA9BAB7B40

178DFFF6E943CCBE61C4C34423F6F5225303F22255E1BC770A8CC571D524CDE6

6705CF0FA544F820F2E2EF504595792805C02656BB02F0E634D8D7584D66E1E8

0E7F303BBA076C536B5D8FED3B3EE7FF3C1AAA7545BFC79885522708AAE763BC

6C1C11612DDDC6B60E8D6B960FE92DE1A50A553B56876279A3AC25F8D3127329

6A68FBF8C1389943B2DCF0DE312BB4CA6BA405F58A380731FCC1E5D5628E6503

4A986979DCB7FF9B5584D3570752C4E835D4665CE5F1BD9C085AD6F49F0E71A5

296A9E1E4E4114FDBB83D2212DDE6EED0D053EA10600DCFBDD649BD6DE92F1BC

39AD7D66B5DE45B1BE8E2583ADB58170191657B7ABB28C767CF2F0BBD6CA6261

0802A99060FCAB23C0A8FF144F4B4BDFB4D5537EF3AAFE216994DD169D9246E2

293E283E29EE58DCFE939F0D31472AD59D6E70F65F61676C75C668FE631DAB02

7E852FF054BDB56E34B42787EE177CEBD85439159D65BAB23C589CF8872512FE

3B3D21617CDBF484DCB60FD095AFADEC6AE3AFC9D5ACC3D7BB14EAAAAB656A2F

0B177D1484BE25F97A63C184A642A15A4CB341F17C7FFDD97014304E1774293A

3BE0BD5E524DB30D54C91A2B544C61A2386F89644265B44DB2BFAA6CD649E330

5A2D93B5CEBF60E95BF716BF64A268920F4F62C2BC4A9D98CBB6BA38A512A2AA

6FCB51B4141C0F91E7B91E640FF404AD2E0CF4C04603CEDF47B4117ACBD6A4DE

4545C2CEDC795D78A7AA41A84B0C3CFD5CDB6B2C9244DF5D5DB7CEAB08A9CACC

0FECD5961F7CA044D0D02482E8DE8BD6943A2CA01F1A58231C3FB9481CF005A6

3AAFDA9C4070AFADEEAC1E12FB27E355C6A18170D7158B2AE856E493771D88D6

3DA7CE4D6989ED554EC95C7ED01F0914AF98133E6BF876EC104B45C7E90D56C2

62E8E25268C3E8E8AD21AF16D6C74E043FE9F7757E4CB77F2EFBAE27DDE06C17

3AA0602227969137C510DD1B758A2B9783C5426C754A49B5D30293EB44E5AC20

73CEB5F8E0C738F079B0DC5286626B1BECC0EDE0EDD923566816084F6CF18253

4224189B5C141405A80E5338B1941A685ACE9AF447C7FEF9E8168A29DF025A

7F4F081A2BF0C1057AD5FF1D0EBE8EAB7A6C1AA27C6EC4C27F95164C82C0F53C

137233929B5516A4F9713D334E4CF6E39BB337A386B8040535D036C28017D9AA

7E9659A5AF7C2BB01D533225F279C26743F712D9BBB1FDBBB69095919244C973

7E95D995FE1E7C4266A9E32D0DFDF7A9189B3C14527C38A5CC4CA93637127A7A

51531B6D276A65AE159350F19DB9B180CA3B70BBD2700FD9CA63603F95A3484F

4A384F48D63C246FCC47DE0E106F00A50B1B668BD319A79A6AAE99166153A383

6B216B918A803F13C2ECD91E684432FC30B224EF71D130A731BAC82E91CABC78

7898C22B8EBB6F80DBDACB394A6F1994614B1BFC6A87A35878BAD3E75C9E10C

7629CA5A3893255466BEB5E92FABAFD8E0ADBED86411C91B1385E64936632B04

604ECBD5E9150B3C99DCCBEFD57F106B41F787A8F2202C50F7C8BCFFA5D2FD0C

459592141FD72C5762091DA7E3C5B59C5C2CA6978DAA036AA0B541487565395B

10DC0842AF178DE0A92588D21C1866B9C0DA7625987EC28DC569672CFA803897

6EA3B49113E4A9EEE255F21202CF0AE23EF4A7FFB567109B1A94C2B9FC86507C

389DCB382AB8507E0D0D80FAA2CF25E3A67B2A6C244A1ABB1D2F23A107B35904

40276786BB1CD86D6EBD35C04E609348AE81B508DD364299D411FA28759DC531

56C433D9275CF7EC66568DA74D230581EB662882A9630A018DB6F8EBA74D9DD6

10C6D8A86187B222391185F28F67BD6FF53E41A43D345D068F38E7E52AA50F78

75146C8D6715ADA165B718C9210BD8CB00A161B22CE239CD4FB13FE708C5EDFD

7B5AC9063AE008AE41C6F4F58C567DD19E2993094F7D4748D703F701265C0B58

39E33DB387CEA14683D4875CFB1D79825163F0F067BB2B3F3AE832E73B7C3135

65FBF98302BF005F971E0E33B2A2F89D7838DA7C630859BCD19E2DA99B23A8B9

55FFB8105785BF07E3BFF017EBE1CECDFDE1C1BDE157D9285CD87A953D70647

215EB139E0F5DF707D1116F6897CBE82B706DD8421847088D9ACA25C8CEEBD85

38EEF56075B31D67BEC0E16238D4901A997AA900FC57DA9D2ABEB2333CAF85C1

6BB36C76EF1CCA3F1F7ED58E6E0674813F5680A67B3E927507F515FA61E2E390

23235EAC6F8B7E81FB0962F176346D57D976FBA01160B86235373A9ED64F6009

23C98D5FABE7CF1733633959393F80B82733DAE5D229B1B0E734EBF1598AEB3A

599AD70362A2AA06E64221B6D23EFD02D16795135DB9C2D41521BD0A6A055530

316D3DE0512BA088595D25524CF8E232F78C0BB180344F1321B4F56ADB916D2E

3F282E63C366AD40D122276BF233B5662F7C13117BBD27D128D24D91455F8EEF

75F45BE45D89B685D5826AE88D828F27B0D45CDE55C0A758980ED599C653CFEC

3D979BDF5C7226C8B99A6D7915D4F0FE4CAE6C9FA7324E6B8EB38CFB9E066D52

62870D1708204C936327F639792A1F0E0D569B513CD51CC53B22FFEB5281F698

199949CBDEDDDB957BE3F7CC41D24E865D663E2A8BFBEBEE9044D4126D88BE9E

5C953EE8F451388F6CE32AD38A46968AAE98EE9A9FF1F168BC936074CB2F7881

585A204E1DC521CC04A62DA5509892F11F3E534056E6ED448CCF2AED4F924502

490857F8C41F66ED5813379A4C714C5A7569FC9EFFE83E00388334FD4F4CF54A

47E0EA7BF42D18A481BBAB7BB999F1E8A4E8CECB1876663FA0D560B11D0D168E

406088502B2EE072162208FD5A26D93F15ECA158212123D43ADED31501364BE1

363E699304DB280506046AAED3CF063957E10B04A870126FCAC491BE4AC56480

168574E7FC8F6ADA3DFCCAEED4803646C9165E6CEE87105DEC23AB4A3B70DC07

553AF8037FD8A5A41A7349D017E1718DABFBB9BF61C2F231CFED4E1EFB49D691

7D83C5CDADD3A80999E2DF165DA2ADD753C247B665C6D36B2D671DC961D96D6A

783F59C7680D7FDF6A42B26AE4EAE317B92010D565AD2C754D1D17A65F20E718

421923E026AA7C44A21319860AB98999B4CFA504F37899FAEBCEC3E3933CA6AA

5A946A63BF0052E92AFDC6C1370105DD34046E997742115478EFBEF73D673F42

7E34FB66903FF9F132ADD257FB38CCF9D86C5AEF254C06E78E6DC33454F4CFD9

752B16BFE7D05A2816B52D57780C0EC82F277D964FCF2B0AECB3EDB7ACFB2492

740473F42B47CCB2A3F8549CB0EFEC8049A86B0F129725B531E98F9F52BB0DDE

0BF89F562E4FFC543CBDF37243579604291C51431131E062EDE44836965881

698BF78640699B4B76DF45F38D228BF447582376B82537951A70232AD5A8B714

66CF38F69DAE5263A594271BDDE9418DD45153D2A5FD2F327F167B4AD821DD59

4EB3C344D535C1A93CEAC11BABDA287405D3EB66210E6CBD2C7F9B35F0FD3824

262504CFA7D831856F0AE52EA012E6D1EAC1FAF255E8DBE919F4363F65B59D19

5C5F0C6F14D1FCE5C60152B57074C8DF1D40AB6A05795C7F71914226A30CC720

22BAB0841D30904C940649507F0987C916A202CCF2BBEB15F53CF5DB16692945

3AA14F8C6B19C04753DF89E3C6983A74F4E82491ECBE14C58A5AF0DB1E4F4495

2DFC893BA94F249D067788C8E17FA81BBCD8CBB0B2259DFF8289D1323D3A80B1

12EF1E73491FA5589921E8EF03F3105AF92A592E7D3A66CA680DE6AEB8BD7539

**Generated numbers for p1:**

7F99CC84DECBF030A88EB7BAD2D092D87DC0D3C4D9A1D554B576C3CF4611B2C6

22767119DC3278FF0E6B375FD691FB1BF2C833E2A0D8CD81ABDB67D76A322457

398CA0EFD191EEF356D49ABE19859C7F42E17A6004793B79DAB714E0577EE0D6

462A452CBA7B110E9D319F6E4F8D11A4727D724E89B91BCCE4740E58129DFC0

136CD935DAC499458B810E63C66877266333FDD20C28ADACD8277C461FD01B2C

3F7001690F6B07DB7CD4690834A22B8DED1552D21B771251D4A0C1246BA5B945

2AAA3A0D863DFBEF1A0945E73E3A91D5DCEFB0DC49685428CC2D93D85DBB46E5

671929B3531C87794501F106B369B95C5A29EE2909A0ACFFE734743401DEAC20

797FFDE83515C19CF62807845A51AD2DA54EEB8A34A9A7FC66FB95231FB3B3BB

780820AA06E5DADEB0A557A8E6700BC02097E304DF47148E0F7DCA3709B0B32

5B0FA7D9388273AE3ACAA6873FE4C8CB57DD579A5C29473505D3E788FBED8EE8

3D5D8B566E8903452F1F9B78AFC0F35C463FD63EEC86DBA52F2FC962DFC00512

5BD00D8C963AF585D3DC1930D0506009B669FA1809D5ECA1EE36B0571930F92E

3861DB4695BC712E23DD0F21D30D30A79A2A05E1DAA8DA4520401854D3E2B207

1E9C8F775376DC4C18EA62E605BDAF010348AE7F1404CAC4B9C2116C65D41EEB

564CFA28FE3982E36EE86DAB0753AF60F94221B582C87BD762944829D7D4ECF8

640BDE304F740DF120CB75EF0B561A53FA9FCE68B4ACAAF0BCF0F3E670FC97C0

45B6BBF625F8D3ED60F551415C97A102F931925510856A8112E089E1F8BFD691

323DECDA11B5F4248D6D7E508CB10D5D1CCA5D86B2C1B8636539DB8C5BD9FE7E

2907A4021F76CE38E718FAA8FAEE2B0C6B82CAED71EA0F36BFBC0DDB9CA80185

09E8BC5C6D05F3955CBBEFC3CB4B17A5625EAD7A82B305B2A70B5E84FA12F539

7CC13399488DDE8075CD6D313A8C0B7A98A86C392D0C6510BD31FC4515A2A860

305989C7057A150581BD4D366EA4D1B7C7BB49AB8BB8668D341D3927511A6339

5A7F3388B01FA3B459257A86F3390096536D50F1244057349C716D14CAD053D8

465B8D0F19EE7E1ECA65C622861A8DA60ACF7E80749FE5AE60C2F4DAE64CA21D

6D2AD088AB7553E4D152B83855161176D54AC298A87E38D648C739CC20FFB491

5C380597D69FD300B3D08EB3EE3C1F83FB648053495FF70892499DD4E21117A7

128C73E5EB98CAE34AF71936AF7463DE210B4F389AEB49052D0F48B58BD189CB

5182067E0DA7F1BB5814C29BCD0A1B0DBAFDC3D1D60DFE163EBAD1C2FEBE41E0

7E3BB646EDCC394B78DDB4A553F322DD863B5E94C107115011A41DE88AFE4B6C

727659376909A9905A45542ABF8214592A1EECBA9C0D28D3D67A18BE06698F6F

113917E010BD9FA7C26BE29B976687DC726B5F7B0D60BDB27E8EC82F5EDFCC93

5855D2D4EDB8D1E4811120B1F2A775B40A6A1272FB06149BF879011342FFDDB0

70A61353ACE7DD7E9BF2135AF8EDD408BE99130F6C2D58A4BCF0F33DED9E40C3

5FFD2710756D5E9911A908C0B5684534967CBF78C5BCE8A31225C548D6F5F7BB

4DC314AABEFFD9319E0729362AA77C97DB98A93F746F4490A9BC957577ACD1B0

5A77A38F89ED572E77E13BA1F256F21A1D80749428A263FEADA7155638381187

**Generated numbers for q1:**

3BFA0D1F2469195F60F52AF87A8F498A19514EB07E42C2E0CB5F1BF27A845B59

0DBB214C2790AA72DEC9361D46178BABFEC9F45DD10363E3C7AECF1B01E44E6B

0CF317A93F107BA333CDDF7385F67D6D38EC6571AB12E55514AA8FFCFDAC049E

5E4B171703A8C999EF7B3A0C8D06FCE92F023629E5B3849A4D8722556981744B

75EDB2DFE346E9ECA1950DF35F0AA3312FF9FA49EC7ADF4FD6C4C7844343C99B

2362A753174F1742D9B3309A223123185F273108FDE0C5468DCFE8EF51DECD53

3EF078257B465CADD35D570353BD174A4F158BDE232BEE6AC2199F2753BED16E

0CB6BBC21D072BA5E0D2580D61A64CE513F2823CCEDE7630DACD3E2493D4037E

25FD5B43EE86F2EE6A78EB114E8B77AEF7C64B0EF84BE0B945D6949E4C8E7FE9

5BB961D9EECD7D2525C136ADCD5BDF7746A3BE6F8CA006BF5B1BDA34D411B033

2EDFD6FD98C55270109E43BED47DA25A10B690516161F95230B04A74063E3A0

3AE1488B6D1679AB647E3F560CE830CA5CF086C729D7BBD7BA070FF1122179E0

08D4A2E206FCFE8B2823353B9940BF05C84A84A681EAE5399AA3D5B089319CE9

748424D1C45B5ED33333AF651B1800F38A1B0D6C2A98F5AE016E0D471E40512E

58D32E3B26FBEB9A7AFB72E6A8F0C8D5BED50896EE88D34E3470602C51A365D0

09C38BC022D373338CC211F668764221AD544E16C740D787E125E337897D3383

318E98E0B1CF391BA28F615D9D44D975508EE15F5EEDEB5EB78E0D3A6E1162AF

510E888A82E4D38DC683C671377DB77FF538A1B0DF48C1F0C18D6B9E4B47D12C

412B872F46B33AA3771408D680D045ABB7246A849822C7AAA6B839A7685E4490

218CCB0AA76AE3D5E90BABB7E34284F39014C312AA2E8F1EAC83121DD06E5157

253F767D5C00155701FDC8EF028DEF3C48C35EAFDAFA3E2462D13034D93A59D2

0E6913A1B7D819E74818A8E7A64DD21C2D1689395171CF2384F0862EC8B5051

122B19BE81358C48D840A45BEDF9300A3411468F40995C3668854AF653F57B1A

20D76B348DC85AFE8BE76B132B0A33ACB1F840BB06A64EACBFDB6BA0769C9773

254804DC08F05B5096955238C61F2CD7D1A3465C7A6C442AC4C35BE60125351E

64DAE862D7C1BDF33D454E71B9D31651C02A75CBE4A7C50B718C542AE4890FA7

117E99BFAF92B43F63547D8AD9834BA55F9F82F4320DE7267611740630DB1AD4

446015787689E68400C9E7131D5B2CED6DD58268CC243A22AFC09EA931DE6DE

2C43C328BC673B950EC0FEE235FE2B42A5B9E726CF2F1F823BC3B20563A8A128

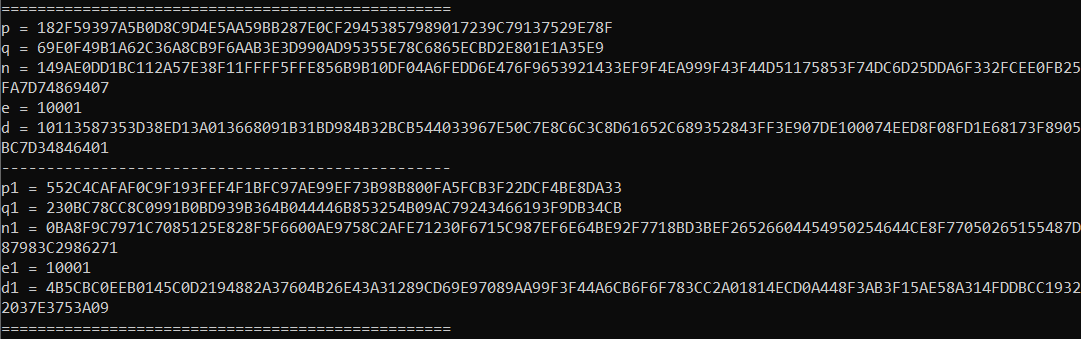
140B9C3E7D6F6BD2AD9B88DA0C1174EBB1AD5E66CC2E7AA967B9C6EB4C4D473F

7BBA4DD66EC02C351B53364C43F652AC0B1C72881F8BBD2070009D819D928A62

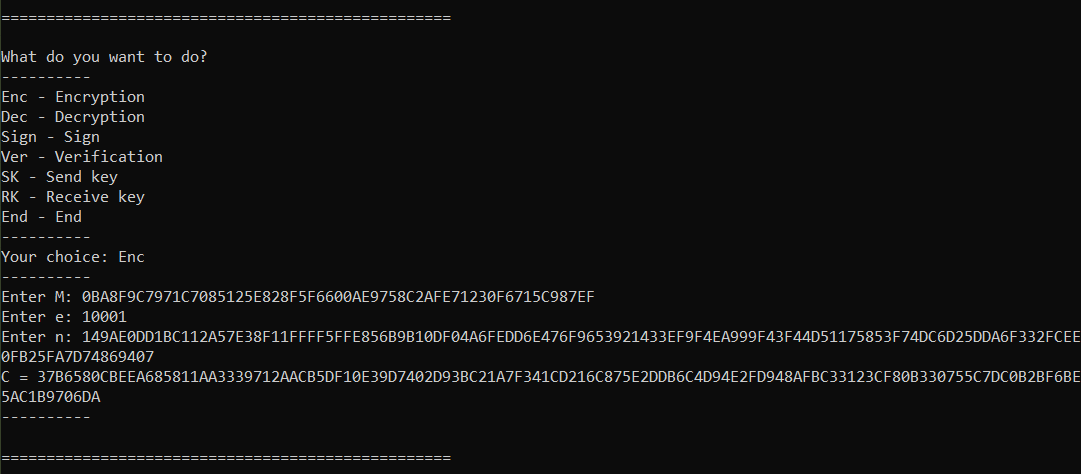
2B69D56EA68D4BF68B906ECCBF786F49FCEA83B267379EE908669BC751B8338B

530C468C1E1ACA0B3D87D60F3C6209997F6258CACDB2C90B2DD6B73B497BFB47

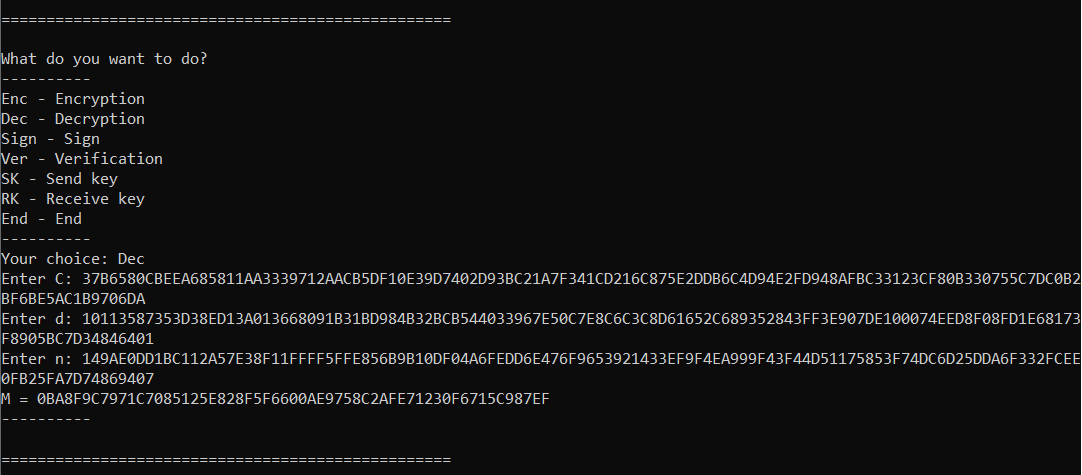
**2. RSA**



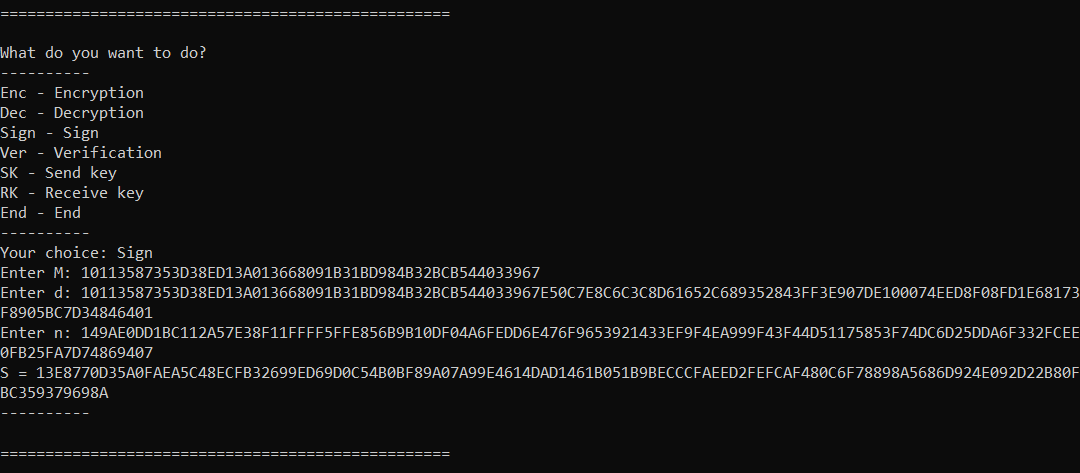
Мал. 1: Параметри абонентів А та В.



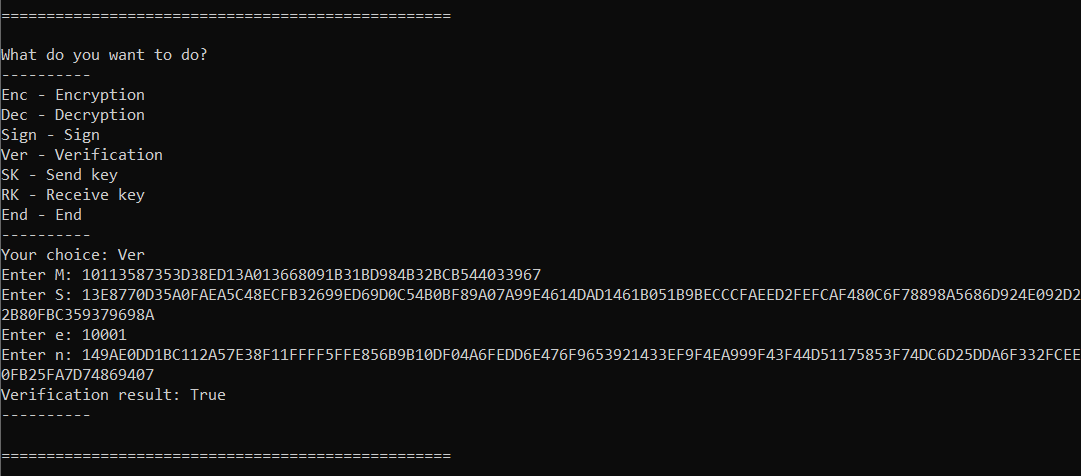
Мал. 2: Абонент B зашифрував повідомлення.



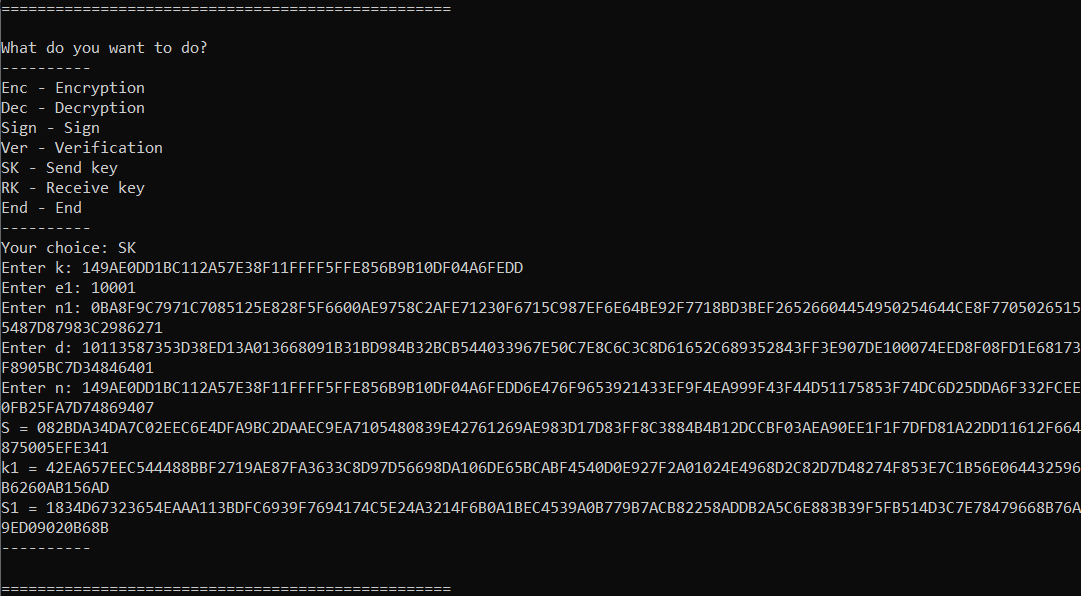
Мал. 3: Абонент A розшифрував повідомлення.



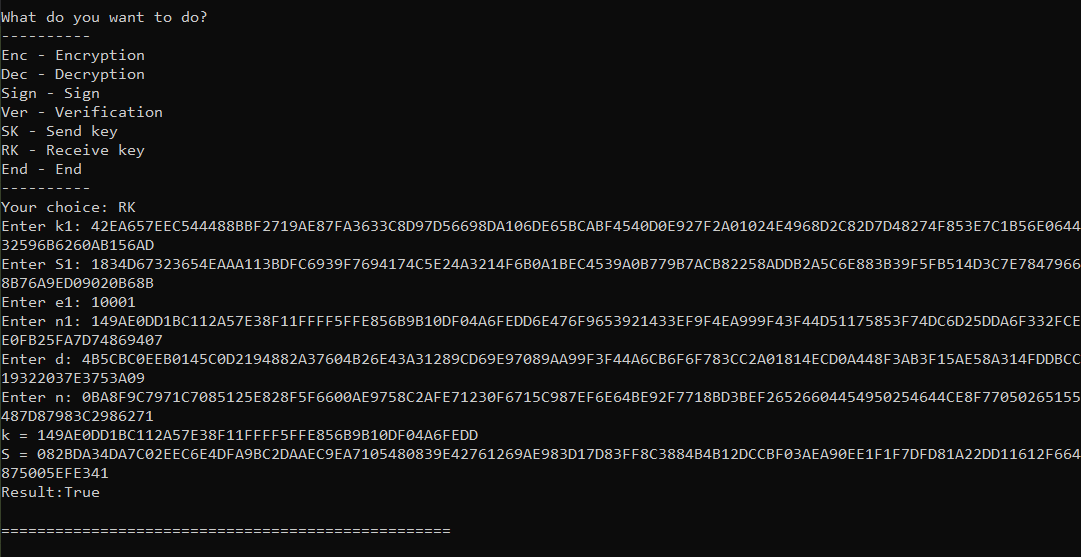
Мал. 4: Абонент A підписав повідомлення.



Мал. 5: Абонент В перевірив підпис.



Мал. 6: Абонент А надіслав ключ.



Мал. 6: Абонент В отримав ключ.

**3. Опис труднощів**

В ході виконання роботи в нас виникла проблема з функцією зашифрування, повідомлення неправильно зашифровувались, ми пов'язуємо цю проблему з особливістю вбудованого метода Parse, який ми використовували для зчитування великого числа з hex-рядка. Ми вирішили цю проблему шляхом написання власної функції зчитування hex-рядка, яка працює наступним чином: спочатку переводить hex-рядок у двійковий, а потім - двійковий рядок у велике число в десятковому вигляді.

**4. Висновки**

Ознайомились з тестами перевірки чисел на простоту і методами генерації ключів для асиметричної криптосистеми типу RSA; практично ознайомились з системою захисту інформації на основі криптосхеми RSA, організація з використанням цієї системи засекреченого зв'язку й електронного підпису, вивчили протокол розсилання ключів.

**Код програми:**

using System;

using System.Linq;

using System.Numerics;

namespace Asym\_Crypto\_Lab\_2

{

class Program

{

static string HexToBin(string hexString)

{

string binString = "";

for (int i = 0; i < hexString.Length; i++)

{

if (hexString[i] == '0') { binString += "0000"; }

if (hexString[i] == '1') { binString += "0001"; }

if (hexString[i] == '2') { binString += "0010"; }

if (hexString[i] == '3') { binString += "0011"; }

if (hexString[i] == '4') { binString += "0100"; }

if (hexString[i] == '5') { binString += "0101"; }

if (hexString[i] == '6') { binString += "0110"; }

if (hexString[i] == '7') { binString += "0111"; }

if (hexString[i] == '8') { binString += "1000"; }

if (hexString[i] == '9') { binString += "1001"; }

if (hexString[i] == 'A') { binString += "1010"; }

if (hexString[i] == 'B') { binString += "1011"; }

if (hexString[i] == 'C') { binString += "1100"; }

if (hexString[i] == 'D') { binString += "1101"; }

if (hexString[i] == 'E') { binString += "1110"; }

if (hexString[i] == 'F') { binString += "1111"; }

}

return binString;

}

static BigInteger ParseHex(string hexStr)

{

string binaryStr = HexToBin(hexStr);

var res = BigInteger.Zero;

foreach (char c in binaryStr)

{

res <<= 1;

res += c == '1' ? 1 : 0;

}

return res;

}

static string DecToHex(BigInteger num)

{

return num.ToString("X");

}

/\* Miller-Rabin Test \*/

static BigInteger TWO = new BigInteger(2);

static BigInteger THREE = new BigInteger(3);

static BigInteger GenerateBigInteger(BigInteger max)

{

Random rnd = new Random();

byte[] maxBytes = max.ToByteArray(true, false);

byte[] seedBytes = new byte[maxBytes.Length];

rnd.NextBytes(seedBytes);

seedBytes[seedBytes.Length - 1] &= (byte)0x7F;

var seed = new BigInteger(seedBytes);

while (seed > max || seed < TWO)

{

rnd.NextBytes(seedBytes);

seedBytes[seedBytes.Length - 1] &= (byte)0x7F;

seed = new BigInteger(seedBytes);

}

return seed;

}

static bool MillerRabinTest(BigInteger num, int k = 30) //робили по псевдокоду з вікіпедії

{

if (num == TWO || num == THREE)

{

return true;

}

if (num < TWO || num % TWO == BigInteger.Zero)

{

return false;

}

BigInteger d = num - BigInteger.One;

int s = 0;

while (d % TWO == BigInteger.Zero)

{

d /= TWO;

s++;

}

for (int i = 0; i < k; i++)

{

var a = GenerateBigInteger(num - TWO);

var x = BigInteger.ModPow(a, d, num);

if (x == BigInteger.One || x == num - BigInteger.One)

{

continue;

}

for (int j = 0; j < s; j++)

{

x = BigInteger.ModPow(x, TWO, num);

if (x == BigInteger.One)

{

return false;

}

if (x == num - BigInteger.One)

{

break;

}

}

if (x != num - BigInteger.One)

{

return false;

}

}

return true;

}

/\* Prime numbers generator \*/

static byte[] GenerateRandomByteSeed(int size)

{

Random rnd = new Random();

byte[] seed = new byte[size];

byte[] zeros = new byte[size];

Array.Fill(zeros, (byte)0);

rnd.NextBytes(seed);

if (seed.SequenceEqual(zeros))

{

seed[seed.Length - 1] = (byte)1;

}

return seed;

}

static BigInteger GeneratePrime(int byteLength)

{

var bytes = GenerateRandomByteSeed(byteLength);

var num = BigInteger.Abs(new BigInteger(bytes));

Console.WriteLine("Generated numbers: ");

while (!MillerRabinTest(num))

{

bytes = GenerateRandomByteSeed(byteLength);

num = BigInteger.Abs(new BigInteger(bytes));

Console.WriteLine(DecToHex(num));

}

return num;

}

static Tuple<BigInteger, BigInteger> GeneratePrimePair(int byteLength)

{

BigInteger p = GeneratePrime(byteLength);

BigInteger q = GeneratePrime(byteLength);

while (p.ToByteArray().Length > byteLength || q.ToByteArray().Length > byteLength)

{

p = GeneratePrime(byteLength);

q = GeneratePrime(byteLength);

}

while (BigInteger.Compare(p, q) == 0)

{

q = GeneratePrime(byteLength);

}

return Tuple.Create(p, q);

}

/\* RSA \*/

static BigInteger Inverse(BigInteger num, BigInteger mod)

{

BigInteger q, r, t, u1 = BigInteger.One, u2 = BigInteger.Zero, v1 = BigInteger.Zero, v2 = BigInteger.One,

a = num, b = mod;

while (b != BigInteger.Zero)

{

q = a / b;

r = a % b;

a = b; b = r;

t = u2;

u2 = u1 - q \* u2;

u1 = t;

t = v2;

v2 = v1 - q \* v2;

v1 = t;

}

if (u1 < BigInteger.Zero)

{

u1 += mod;

}

return u1;

}

static Tuple<Tuple<BigInteger, BigInteger>, BigInteger> GenerateKeyPair(BigInteger p, BigInteger q)

{

BigInteger e, n, d;

e = new BigInteger(Math.Pow(2, 16) + 1);

n = p \* q;

BigInteger phi\_n = (p - BigInteger.One) \* (q - BigInteger.One);

d = Inverse(e, phi\_n);

return Tuple.Create(Tuple.Create(e, n), d);

}

static BigInteger Encrypt(BigInteger M, BigInteger e, BigInteger n)

=> BigInteger.ModPow(M, e, n);

static BigInteger Decrypt(BigInteger C, BigInteger d, BigInteger n)

=> BigInteger.ModPow(C, d, n);

static BigInteger Sign(BigInteger M, BigInteger d, BigInteger n)

=> BigInteger.ModPow(M, d, n);

static bool Verify(BigInteger M, BigInteger S, BigInteger e, BigInteger n)

=> BigInteger.Compare(M, BigInteger.ModPow(S, e, n)) == 0;

static Tuple<BigInteger, BigInteger> SendKey(BigInteger e1, BigInteger n1, BigInteger n, BigInteger d, BigInteger k)

{

BigInteger S = BigInteger.ModPow(k, d, n);

BigInteger k1 = BigInteger.ModPow(k, e1, n1);

BigInteger S1 = BigInteger.ModPow(S, e1, n1);

Console.WriteLine("S = " + DecToHex(S));

return Tuple.Create(k1, S1);

}

static bool RecieveKey(BigInteger d1, BigInteger n1, BigInteger k1, BigInteger S1, BigInteger e, BigInteger n)

{

BigInteger k = BigInteger.ModPow(k1, d1, n1);

BigInteger S = BigInteger.ModPow(S1, d1, n1);

Console.WriteLine("k = " + DecToHex(k));

Console.WriteLine("S = " + DecToHex(S));

return Verify(k, S, e, n);

}

static void Main(string[] args)

{

var A\_pq = GeneratePrimePair(32);

var B\_pq = GeneratePrimePair(32);

var A\_keys = GenerateKeyPair(A\_pq.Item1, A\_pq.Item2);

var B\_keys = GenerateKeyPair(B\_pq.Item1, B\_pq.Item2);

while (A\_pq.Item1 \* A\_pq.Item2 > B\_pq.Item1 \* B\_pq.Item2)

{

A\_pq = GeneratePrimePair(32);

}

var A\_publicKey = A\_keys.Item1;

var A\_e = A\_publicKey.Item1;

var A\_n = A\_publicKey.Item2;

var A\_secretKey = A\_keys.Item2;

var B\_publicKey = B\_keys.Item1;

var B\_e = B\_publicKey.Item1;

var B\_n = B\_publicKey.Item2;

var B\_secretKey = B\_keys.Item2;

Console.WriteLine("==================================================");

Console.WriteLine("p = " + DecToHex(A\_pq.Item1));

Console.WriteLine("q = " + DecToHex(A\_pq.Item2));

Console.WriteLine("n = " + DecToHex(A\_n));

Console.WriteLine("e = " + DecToHex(A\_e));

Console.WriteLine("d = " + DecToHex(A\_secretKey));

Console.WriteLine("--------------------------------------------------");

Console.WriteLine("p1 = " + DecToHex(B\_pq.Item1));

Console.WriteLine("q1 = " + DecToHex(B\_pq.Item2));

Console.WriteLine("n1 = " + DecToHex(B\_n));

Console.WriteLine("e1 = " + DecToHex(B\_e));

Console.WriteLine("d1 = " + DecToHex(B\_secretKey));

Console.WriteLine("==================================================");

bool isWorking = true;

while (isWorking)

{

Console.WriteLine();

Console.WriteLine("==================================================");

Console.WriteLine();

Console.WriteLine("What do you want to do?");

Console.WriteLine("----------");

Console.WriteLine("Enc - Encryption");

Console.WriteLine("Dec - Decryption");

Console.WriteLine("Sign - Sign");

Console.WriteLine("Ver - Verification");

Console.WriteLine("SK - Send key");

Console.WriteLine("RK - Receive key");

Console.WriteLine("End - End");

Console.WriteLine("----------");

Console.Write("Your choice: ");

string action = Console.ReadLine();

if (action == "End")

{

isWorking = false;

break;

}

if (action == "Enc")

{

Console.WriteLine("----------");

Console.Write("Enter M: ");

string M\_hex = Console.ReadLine();

var M = ParseHex(M\_hex);

Console.Write("Enter e: ");

string e\_hex = Console.ReadLine();

var e = ParseHex(e\_hex);

Console.Write("Enter n: ");

string n\_hex = Console.ReadLine();

var n = ParseHex(n\_hex);

var C = Encrypt(M, e, n);

Console.WriteLine("C = " + DecToHex(C));

Console.WriteLine("----------");

}

if (action == "Dec")

{

Console.Write("Enter C: ");

string C\_hex = Console.ReadLine();

var C = ParseHex(C\_hex);

Console.Write("Enter d: ");

string d\_hex = Console.ReadLine();

var d = ParseHex(d\_hex);

Console.Write("Enter n: ");

string n\_hex = Console.ReadLine();

var n = ParseHex(n\_hex);

var M = Decrypt(C, d, n);

Console.WriteLine("M = " + DecToHex(M));

Console.WriteLine("----------");

}

if (action == "Sign")

{

Console.Write("Enter M: ");

string M\_hex = Console.ReadLine();

var M = ParseHex(M\_hex);

Console.Write("Enter d: ");

string d\_hex = Console.ReadLine();

var d = ParseHex(d\_hex);

Console.Write("Enter n: ");

string n\_hex = Console.ReadLine();

var n = ParseHex(n\_hex);

var S = Sign(M, d, n);

Console.WriteLine("S = " + DecToHex(S));

Console.WriteLine("----------");

}

if (action == "Ver")

{

Console.Write("Enter M: ");

string M\_hex = Console.ReadLine();

var M = ParseHex(M\_hex);

Console.Write("Enter S: ");

string S\_hex = Console.ReadLine();

var S = ParseHex(S\_hex);

Console.Write("Enter e: ");

string e\_hex = Console.ReadLine();

var e = ParseHex(e\_hex);

Console.Write("Enter n: ");

string n\_hex = Console.ReadLine();

var n = ParseHex(n\_hex);

var result = Verify(M, S, e, n);

Console.WriteLine("Verification result: " + result);

Console.WriteLine("----------");

}

if (action == "SK")

{

Console.Write("Enter k: ");

string k\_hex = Console.ReadLine();

var k = ParseHex(k\_hex);

Console.Write("Enter e1: ");

string e1\_hex = Console.ReadLine();

var e1 = ParseHex(e1\_hex);

Console.Write("Enter n1: ");

string n1\_hex = Console.ReadLine();

var n1 = ParseHex(n1\_hex);

Console.Write("Enter d: ");

string d\_hex = Console.ReadLine();

var d = ParseHex(d\_hex);

Console.Write("Enter n: ");

string n\_hex = Console.ReadLine();

var n = ParseHex(n\_hex);

var result = SendKey(e1, n1, n, d, k);

Console.WriteLine("k1 = " + DecToHex(result.Item1));

Console.WriteLine("S1 = " + DecToHex(result.Item2));

Console.WriteLine("----------");

}

if (action == "RK")

{

Console.Write("Enter k1: ");

string k1\_hex = Console.ReadLine();

var k1 = ParseHex(k1\_hex);

Console.Write("Enter S1: ");

string S1\_hex = Console.ReadLine();

var S1 = ParseHex(S1\_hex);

Console.Write("Enter e1: ");

string e1\_hex = Console.ReadLine();

var e1 = ParseHex(e1\_hex);

Console.Write("Enter n1: ");

string n1\_hex = Console.ReadLine();

var n1 = ParseHex(n1\_hex);

Console.Write("Enter d: ");

string d\_hex = Console.ReadLine();

var d = ParseHex(d\_hex);

Console.Write("Enter n: ");

string n\_hex = Console.ReadLine();

var n = ParseHex(n\_hex);

var result = RecieveKey(d, n, k1, S1, e1, n1);

Console.WriteLine("Result:" + result);

}

}

Console.ReadKey();

}

}

}