



Міністерство освіти і науки України  
Національний технічний університет України  
«Київський політехнічний інститут імені Ігоря Сікорського»  
Фізико-технічний інститут

## **ЛАБОРАТОРНА РОБОТА №1**

З дисципліни «Криптографія»

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

Виконали:  
студенти 3 курсу ФТІ  
групи ФБ-73  
Дем'яненко Д.  
Проноза А.

Перевірив:  
Чорний О.

## 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.

## Порядок виконання роботи

1. Написали 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 використовували вбудований генератор псевдовипадкових. В якості тесту перевірки на простоту використовували тест Міллера-Рабіна із попередніми пробними діленнями.
2. За допомогою цієї функції згенерували дві пари простих чисел  $p, q$  і  $p_1, q_1$  довжини щонайменше 256 біт.
3. Написали функцію генерації ключових пар для RSA. За допомогою цієї функції побудували схеми RSA для абонентів А і В –створили та зберегли для подальшого використання відкриті ключі  $(e, n), (e^{-1}, n_1)$  та секретні  $d$  і  $d_1$ .
4. Написали програму шифрування, розшифрування і створення повідомлення з цифровим підписом для абонентів А і В. Кожна з операцій (шифрування, розшифрування, створення цифрового підпису, перевірка цифрового підпису) реалізована окремою процедурою, на вхід до якої подаються лише ті ключові дані, які необхідні для її виконання. За допомогою датчика випадкових чисел вибрали відкрите повідомлення М і знайшли криптограму для абонентів А і В, перевірили правильність розшифрування. Склали для А і В повідомлення з цифровим підписом і перевірили його.
5. За допомогою раніше написаних на попередніх етапах програм організували роботу протоколу конфіденційного розсилання ключів з підтвердженням справжності по відкритому каналу за допомогою алгоритму RSA. Протоколи роботи кожного учасника (відправника та приймаючого) реалізовані у вигляді окремих процедур, на вхід до яких подаються лише ті ключові дані, які необхідні для виконання. Перевірили роботу програм для випадково обраного ключа  $0 < k < n$

## Значення числа $p$

$P=101355380124076268587266947719799178486423436012413065988713643571576687296511$

## Кандидати, що не пройшли тест перевірки простоти для $p$

90185745186582484012431069302109717433682393124464198202083061348896417513471

92884909479496225649857867470764381278496867326516667961136953907377780817919

112836599375266534185892557828991795381068405926330909091078454464163149250559

94868186569517773636644070014505744409878314567214111017979529488965228822527

98437900719763542069560137401950672556905670723819965475365553580797518151679

## Значення числа $q$

$q = 93355627360367236277216721331318062881555725369696294768340295769881619464191$

## Кандидати, що не пройшли тест перевірки простоти для $q$

106911071905580176037004271999867278131762667098309827817753723320388028465151

108594862910361663630268403215990897923047106559663518919974777254117992562687

111393038889014292279576725206507949832105591746624403138533326465326198554623

110979173914216287519821527334927459659130946886902589167284615354858197745663  
92905260605448487199585566399383506901989749688554534401158869849569157447679  
114969650079081539632406087071974064444491730907738640355839127751184963600383  
108121298798113357126241749166764785451500282197288765281606813476847607611391  
106837556244161179877624632951475942180141672852012750101598578772546791931903  
111582962650101772690104071201968855761342649723437057692584398598935567925247  
102912403748993450679149377618023183852250267195128696144538308573705743106047  
114406789811647660159315662166518322718781142323197347684428505446675571015679  
110925562227156649157265312499413429256854557093528927517163091293198100725759  
92466735060894699120569627545745785334046984991954231694555510283999243141119  
95703115270103582607241903459268426399963967678148436480857822282738986647551  
90996407034825390916689267054517749801571223669820660142844155794377511272447  
91030435897075413049725056763013513684007717905714239807300464881695677480959  
91121482467715711351009333185945013759402081533098705696711110927367593263103  
88314263498894788968295029626388109418617063938139834666549541428558963933183  
89447301461093462477187526401066286634222618437518783992420064058176998211583  
112814745245125966238020781654202192171687151049627043458113285368829756571647  
113470622570224584452547060025278956134824465302180874088715148535339352588287  
88829979848056644825117901321277355588635028148725132609513203119439829008383  
102883681831868954704807426599649160914929415030615394929582564898012846358527  
110321242573255122162576154339603444468502778709140606057794826261423313649663  
113997483791306214432394168636117020034874355637903968203957635083144626438143  
108121835558049082055626752715136228135313057237903430996851384307705287016447  
88745100199151944475265230124934456314618365115757316651957475780330562519039  
88605027908919201421553231847691457662972438640588498697436330578786809544703  
107294671770476960566862103913973391517547858060571215231532472477281542995967  
97522892425120679955451869729201636408459566306796693431688876839192714280959  
106252043901895234496750520682254178201245219813237818927870125945730556231679  
88180815911884386275191114593567725902523642532890442595779338886894049034239  
105711269607789731419227807265855889433415910938440687559283536371083135418367  
94256027502959196667176021206382159261822816942422204548045089024573653909503  
88051383750798810883336999093512519618937189060050729645980179282292300054527  
105341162661152504728687104999973106208588979759689695484965751419590324256767  
104673622022725504943091538479671145190646247254799433581506927640771798499327  
93009991352103540282791261880409414759012945856108716009855641646703365324799  
110295843367317648503946262423018041726667340828162531379388494744734369054719  
88046708259551840939656134950601222245526611523260088670076359722219641765887  
101417327390052253210520594475432638395983776096720748608087354972151260119039  
91573762691572073681440557788170689379675929099291326930937137228079498788863  
112395439601804249156770107338730649551712651644487834762809108640066074312703

100097161245677089743808929065929592440330718123576605081479706246443928911871  
113163677263563439228008032568605853120214880536386158155559283045296274145279  
103140113989083724077190157853060974504786826950537457231363135399433252896767  
99013203983859177097041226537770801189972709061377413196589812300875911135231  
102522631335103503110231470260238485477336621656723633261697643273147438333951  
104080631709427652611740626056776813424207857537336322824686857152952384094207  
98842436921581411559030295461406735309458617158251523668213048570066849759231  
90279228220252860435130004936947347729635299738569696935244956456903537328127  
111712587702666780953853755348010445816661243918890708462230381708383048695807  
108396631469384351911143216499379923226172874334995951769237180928271805579263  
111078114024350951832433918413365355678200979830798311235612786396130330017791  
90677037892271319389641037554876595529709416312614197249638842114100122615807  
89020029956091759636796311961654348628082307043268883702496516629140237975551  
114654764936823798832223592578107740044456174717358746671300395346793798303743  
92680038084064300744944482733567059876020629317626795141091187386890542120959  
95218630560834135957549143407584191371171702151569066612662556634332910321663  
106804222969079770282911903749395774069184412493285075838433686458959688368127  
111960471441666847845996181433055674992403081481731976615598872071740413968383  
101655298384511451490836922129675845794609242967582974871009764270021049057279  
91551917662033190173735764899906238975077639279409113013344658453285551734783  
90425973531643898418601973566340588985123353950264591889401160768164693278719  
87839632316119160488506422319603265158103968489308219784237004816342560276479  
107033854850603620575322144942871139639605451888377932716071971298084200120319  
97514845090020437021544989581900324718078278427246511374200726846738343657471  
10429669334998122766766850936471636261227743322490048893325750885530444444671  
87496675994135645292419387802723883787273346072010380375013303196882653151231  
97006074518622986081042961601632555679768883648354915550983429299421759668223  
108922277136508460530434954850205097078141371020562961422396206074396565045247  
88523135568052283911367624271389866633703212945104944515885605823696852746239  
89350894180934972891949247670671224082596959898764900624447297449124798922751  
97825918089748456501058761042680220428865147471677815266527140393557036105727  
114563526389330149367179225480212358909541419294551512160599179064175975464959  
104369327755979947102215201656140139512895902899702540565996574793093204672511  
111632454644293240650344464354159713619944603444489822485774456627479739105279  
89723784492945320315709545588364486189792575461873636068954451300700947218431  
89538730009963416836347365406352978192344790539459345815536128250227797786623  
101653804132613346559128659080676453423300558749762302627144498029606513672191  
112035298580065949550515151780750256631989911673908572589131722285806683947007  
91871314251329012060858830120164958381473449480301735156495220838403563061247  
99647748583032586322243866920673117730983392743477706638875932859062678454271

106185282499324178431541598863802590201406963812358428955695257772978306285567  
100687697071569111729722754487571449046227032273828439131149073775524852006911  
93392855660844702895168712295343969977428236277882225080415096767249593139199  
90544219438765179219802677706939615545008598288665960922937359101142392373247  
110795468191041035970469959026146372334769737751394826628586314028056053809151  
113621060197173609839795906790669735264917734470403298644203258963825140957183  
92526443601197757496891261659923517104419816335863673130356850425034117742591  
87812561407053826996612837016845053363822177257206610177831720231604861272063  
88418787069206208982221058617295107092814543174093084729569915892994293104639  
110719841013686177247473102392077569716703395824907678671218135665314807939071  
104027653159626612494322589466238740492529654250183618223518543501307966128127  
96231689148624225517098442195547464197703354328725019743990180008124775137279  
98963289107075072390785939216338899334311909031618189997827061944554753097727  
106613003149206631212378546195670057657345164075636331984237238540654071513087  
105930581823349947390764932684062499960022436001423431159883669287471203483647  
103926695488259169836744593127626944348111314893881293909559474855110707773439  
97646063755389225595466267411677808943064036855985100861613834025625163137023  
114708942763400139601494806254868654367290692304291599683712224952769243512831  
96204288575813096899335175940286474303814368158875565700413597777162089267199  
106639509339198485608769501672962218779940354944113723098253322195201905655807  
111306708933279064269378720631068113369893811088448951925109886577668610064383  
98045480060933108958293943977158806991224185266705920340211545606580672135167  
103229430171486264258228875546708563902670964313460080977901616603826478907391  
111600716039337817115145851418808584862154972808811522342092845950679352606719  
104047223929293708523532410359831172216202420153603494951835135226648473894911  
107911558587493748204306223855614331390914125368234124759845377899764413104127  
107516334929437782865496086754246766619630729203206162673979556231161193693183  
87356470995107444509201750309548083937457638886728038368546112636401307615231  
101882281779791458792592292482292947703019073720579351393987971216936280260607  
94651928866459357784324249059829711768452898082641595533120063868046682357759  
97341839917455944531973834774745342165621889610141775287052222415296882802687  
98162122650294078994278637932100151655856287074928594049211241164648562556927  
108892091639615944062059428712500295708567257433099218980429496223750319767551  
110047272600515459702967387035264110823208854639521266166064074274590007754751  
94010657134045660519089970365082293881648700306543077584539521977033496199167  
96741759943228253734908385053101126054092523631106079431801947240953695698943  
110942063345623639897817458688931689369673871228621723330970397412032352616447  
111656370787843451126762497620308396560747408042295644539825647215867307491327  
97325997367325595989791517994634026784693680289018847110088937644165425004543  
115555285503500685713003384960849714326731041180612333919424084076821728985087

103753247331763217760615038569564736141005943295960116214886248375458488909823  
92000539454293931407321933971915273267513499296056124040194191278308262412287  
109362574453470107709245294781587772758832102505737103070288043684425375940607  
109598530290437172989959947114747775796685637503031398109189041465158233227263  
88539649699778536299356620215667891078498702049541183359045403723050724098047  
109891463011446622540850317968539658461700792200306761938536522460939826495487  
115640945840312960954879340552254854073917394825566182143441489589745787863039  
100119252751639493647906263042681632130647382215642252576091075900587085135871  
105730638357344360025347374291916023855954762999832586794577277626097357291519  
87905650194665761596483479024428571152610677579820860744253953621432077385727  
9330526084228996637135336192973449901001776033573709595374262025855333181439  
105787428241203874751469539387747109844748285835741838774838986929959277166591  
107074834886059996714684187517732040466493917333970023650335048386246504087551  
111536621204243648783042640576092014161121562874748401351137923126450370641919  
103550144297473556076031380209693041322248624549642092444462855400508067151871  
106178576744741139281620085255569185315185099157332661358660593527347252035583  
104903368418867652128152346576614192241465483432882641894421636246683454013439  
91041656165299578259189129251027626349130009540679258125608724070824597782527  
109366915157375922986392707660357356376473163752562353084435818659400713764863  
113840770425713143338151233688586050529797274309487813994458876598853222531071  
94842044325898877275110768965503277337493981848370298068201021187759624159231  
107634001068743446148045870195036610836722384714495030899899548652086379937791  
101529914878744810090095201465506809775620034165310848970562940145558376939519  
97469907231779221886562642655085823771441428761901896863773074550520994070527  
102247400659577879234084355623127197904389465049324899897737590337240932810751  
94005520084119858726452460380594480976825205416138540155786950480847886614527  
92104289951884589357973745413363862553246902212886820214530319801322280321023  
112263050551035406850613940566599326718575097554072974948139215494191880601599  
106535305301927132858810176189103003108780387509233462532520939385327257124863  
107959527726798892123445687819833509942944471089674990374084512863711924322303  
97568407566581118477428818814944503767943116949240717327784323146550389768191  
100136765356695082870331032309439415317933889166217251043288190958344580628479  
103822114035111028939307358217298176037401186984796670587961002627130206453759  
108803116220331917201467183016837527474361036692335494717638811450194308104191  
96428383376313105798949406130015756351702678773895543891807789052247440097279  
90071531551463656732188375491865764684021269022218047668754317865626653163519  
90881643004050177393680668674371869336810294179251178501167510303932038512639  
98058845716636597323108125074199235063470475683120176808024718251914073997311  
106429973385122421101174869649832290616779659809121521987441181543159826481151  
105800175037083762674559299206097405544329933646291072949859940025793484161023

105802291659671950688099833550948635161010302734387798542981938749026756198399  
87876636668171689446021582419843248928649681870992543161980891835352695701503  
102669511316098143294596389322657808446612975700949613991989266529667212378111  
91672464042363066468888027741422464605623369867814625492777984019409930289151  
102378029559480608816622613689574940252603124313716622801987274034125199114239  
109369157261525574944671995349558811845940557861391731752041695911214105755647  
110718539296841042940190931331873504471236028708841185161609661884204161433599  
105750605363873604253315682066360366935284820346019830801422750032839504297983  
113406414981188227316320536356230885758550719835476367655163524758322020351999  
9519172779488958023648742668773064537621813258885318910435979713765770788863  
113836569721452152882870774422031608417433523465700189884224864740685046087679  
93723424816678458228695293928902521301387482503595170317965479402056829632511  
94164751663559550184862174075264532773973347752971520201461747049374777606143  
92961641112460533282007870368062300230117662361515457574379479016683058757631  
99575370628615793344672340440824039461042437410557812873640154695227479687167  
87875044658022004407145868030579926022476076048545837088257105385316736303103  
88608254968343741710133676551449345386388552256819820593120261924968151908351  
110520824487370318973101959424226607412937350539423734817144423770535942422527  
11102669192644973880816956865655483649826772070631275116201508956851824230399  
102652292892020028582152780156950774339050629319212637521988327991063433183231  
88896046222828481575864184436206116272358909577601270007093048350862305394687  
96100855982593813070136758823925212633832855188405473800400017570866879528959  
90725453435934743270398026926859177000186823349007084607556232022118369329151  
111246744689309813559056108595524214009156037812874432124248098614619175649279  
92249518605040919629617191450098131783656376844929904621933815829520898326527  
113021718943808950576094135502598844643924220101913347177958461648995574874111  
101320655541270431259819436163559638915819283282036925685858005863096022728703  
103698047055257088794598250221153471072691598087957917275689185733780223033343  
105963087122235976195130031965150286607861543966425666187065229189983468781567  
97125362240970198369131418835014435557125280497966133694737469235361501675519  
99953252352184660342243647314012795196774377952099474382276530070482032001023  
97299572804492788933699886139481416397174962244623258704581710616783233220607  
108531881970468604697335471329902156748846883620150451206662295619302254968831  
109181493729414282153193766034200023084774923483480913879582132228467093143551  
101617621760244434592196055174773478648457774628810593580057138916810232430591  
111570484605435771360711143257748340368182207088888866709347210230330464141311  
113142524105053803749738978735696473624442340070355702010505872942780207071231  
89288479246739738728937272208801798551476648943253624601691350306260294565887  
94558732942506972297845750351678217161079914081223880912958305193232277962751  
99956717696118826590926101054595651061603378700222378217314032337457575362559

## Значення числа $p_1$

$p_1=88265247073978377837946878484926123522413904627216961245034438951970524889087$

## Кандидати, що не пройшли тест перевірки простоти для $p_1$

106975932357842933025799675430219416316366892216277889260632628970178597617663  
112186483902000412803123077429294427286814870781855414507999484280821682012159  
108831516604814634460337734408090420601038518561739395666120628341315731455999  
112797829655155147453547599956228381999199957017068581231867791262065207803903  
110739294048111856382878113936243405544085028575672472955112746137050308149247  
95652647689134804557885185610964486437666375486099346207351210426157353664511  
114461014420762465153797705642156713959504401521651420783730821538992439164927  
87894437767609985221691853964320704228169806695781398511445489806586547273727  
92035487467173808001031175211776822706032287170939849228000714849165481345023  
91565801690972803507038675219074933280550714159636468325973768856031547359231  
94247994958238975367175038163424280123335807692678676559961165156065057177599  
8825931490944035856775849899076791740513356709808384537396098077124722687999  
93257006113411413782128254479340380999818553020512558481186137249172435238911  
94608133256304572401253447993585263737795538976042460249615001850134357082111  
91066817427304638991403536592425438975137389067997791556135904584169689186303  
87313218362766899902444907719448151473673625555923978589353439238165039677439  
98882698529579461320230679917012668934388172361108049443400097031249454956543  
96464442708659942942184715833856750438195584840602674501626265879463187185663  
100360571514076796651967142845570289797231794736333939127917547477680059842559  
105242045718183263442261677714056724657838577752832510285538853431350415851519  
113558153051518118299661821768273283228581319914596984738984733511399776452607  
105293079012853057822955573478433515946177884954817734917151086709700131028991  
95238778789008914632512959178089348045987354201293595822487834481980984000511  
112546373338166031940901264193320775942897994079147229096257211864592687300607  
95017624541796393310268863228166341842218374555864157254556358344162800566271  
109024067433565252430356529287395455550815770027020582985150977044378122452991  
89276418060925880793608589163051688853526846589233648934607392218663733952511  
105276250794870396341593084698183671168478647165800161910995782887972446142463  
94409994409972337312379520470841071950269668728233458150765395120039640694783  
90143187118206739050822459518439335681194063994105640787846352275078507397119  
114524913468721317507544841311039884033741167410555751268826430014093053132799  
101803898842774339658207414388293763496766387507605888665487760294817659092991  
103835181756023562196536078808839713179442671088210565789605958777779951501311  
92025065584351972646241454147266555726830370856133846371816816919616933593087  
9538736414990012604399115172220824933555455085927125063977416003915574411263  
114281522549597818475131550448255365636867012213922052799334872717401239060479



110885394725147611410731716031696934826891642672356240889428325737314002141183  
93368300850379214328990289550971094533978305590787917670128067672154233110527  
114000406773704331225246087205290208453988693703360035774896282198713181405183  
90527759192109849436479181466086424919957769215358327397529651356314789478399  
90803561959376238510163141944119728859479763710365995770742933831469773094911  
90246238985751202695702304252373557203373182463721903002012015336771396042751  
105590285150393565647639268785585918292253821422810108299250465891071410831359  
92003242674566362074581536878605391227036165457995551225364240145536478347263  
103189718932729155387312649167263283509038832726178775239747234324200831320063  
100466194370032476118236416253184201340831340211598954378383351169665216806911  
91862384061067660105525486643681254628739957853083666800660305665810589810687  
114913265321782920400068476452154608575162263747619320565797711565711269167103  
93659019898125220642213214241610424102183051033564443697982137091627147067391  
93862188615206453503235226122397099109261741328727031124593677224808586149887  
92783466937026708971568542071073732327290877242942198799770527682937468485631  
114135839825101735492122340531873995607022112450478483683980064255759324545023  
113798146868937456657227529886289155627613220619216913700161176474121492496383  
104455776033459108396382780042580961584550729100681484245312946266135458742271  
100244943318647150796319707432260534449584655972005647359324020467827778519039  
98950281924533900722596675832592537870517322328111635871148382642200498929663  
98020825766162267251020307084763980811074945494482018420429237570221620854783  
111990619071894013463516781103686085370756208885164383971742473986327319674879  
100752810982587879054453278690487830726940966529462561592287449429577322987519  
109479701142842806371633852785604580846880341407131049517926306846386335776767  
101754017095790518980814109574491759794875801228396219834668563204478934712319  
90854230877199396038241089459085920993112688131065935948726576829269941420031  
113765137544841629872340713990747130459448299443914801940884404153608111652863  
95568675541325550821855351088779186646624331897796910343752406850479778168831  
111383942411859735770823014134773123829893828589639059617473617219380809039871  
113471461001241509946121219902245078060334692286380682553488978624757643608063  
92552551128460123515248695933964637083379918955208012354214252151737868615679  
104801923954061649701814956894852662052442162541321802261335853787672970526719  
94415190771804570236223080670956790833702186732921823506649484984616235302911  
111745618017305103224272448005095996101105552291901302290362166114456795348991  
98793132724301610778720383225770183277661487401897613184925959129781967519743  
110179138870911800954757660269781938343829690140790563543929583774078024548351  
94923586327200915984824645004839890947477416837414195576647631792347679817727  
95745083299321875586803471587415556630001246362774915116720953994192793632767  
95021616088571682178420517351166895989284331088634555141700094254603791499263  
112284539922493540683181276188523924277419935525973247353789909163168665436159

93326226053732230677725877404103383643143110628026292890670104590269357752319  
102824530607286617492993523445998014119451564475020835625979797687642903019519  
94971517070697479770894984876097802113895609296631976884386916267772361048063  
111096999398399538957050058975513495424504237570131491623359745432893826007039  
114840539809579590318436684586264593055519678725148961725955957519993163218943  
92224302953091022327375181400532571374510235678866624737976367291546399145983  
113820690406753937389508343314224229388241462046110730113892085112856624758783  
104787408231127154982982249299243070765936332488299642546646661119918068465663  
92196812392272017169158216576845065119338331802591379626827072938253085573119  
98011076997048878341638855320279979362038799873445999277860871700270167359487  
109102008783312152859349617488007344680114055118678672348547525867627295539199  
89591495596733214726485278047783168187231046578191355277972303199328010239999  
102082227337722349043387272302135980405723762429126217617018398447688428290047  
114840239510467510635860972044667485230479674995814457259749015339931694792703  
94004665394698709167655901877944849613561572601693790411449528113249820606463  
104482408161249648281957542004103664966609888549700138484001355804664418795519  
102738137266777020815159803338915699128877944624463065519439320100391304036351  
111932723662536736411093715033999944163312371286214859565184248413438076780543  
101965479219287070561909620787087806915835628679835646955530608415949843333119  
109122510989863985609229227743510639008799779482153709415322422580060610363391  
98294142808093423560460271571963046093521086660318104249362357511656805761023  
96850546996885020602411422822627206152776523844623929672114411098396321382399  
100543483318683328556568849905828478817567204188682183910552676771614236344319  
107012331314623847793304820356255548437554279110491705523751383070758559285247  
111481925707228133513831617213765230501208948355853604704852555275340972294143  
114379948533848681383610133820449341686841960362908760239456458695874281734143  
86924950367964616495463877568567118561274931994572798417738222865969240342527  
100642287873907439936593138214174573052513096051158102174774117029365533376511  
95636173963066088402392102752058002245649041750043978178058838989742402437119  
109731175799960361875341210966229995199152989824726836811436290188412832448511  
107795946782140328395793041440930490235282445642088100654043427861322042179583  
95562660340270774420493340568524513717578248857074256120810502158040122261503  
103862197704895496074590787320324158166771720253166749094803844944185060753407  
114525293756690119246060859375175412455896536119236605739977091648818495094783  
114017601519024085144220989908984717161429991803459467286168952786809058754559  
89213449117697240409100815285257933469561396941444067065457994369883021770751  
89921959597956064956977697371063555312492707458048018287288102831248524705791  
108424833098394795107111238125755100569112531894763718241616637638833885675519  
105073541862461667677645728506262128875454655262338181874679591666260765573119  
103944338684134661290754131818899363956233657687914879352030441376559333900287

111198126702173830344024092650696485755576681546996739107982535501208115740671  
91442537942791500039307821967135197986256078859762578685163454072218519076863  
103292343392703057167168963755919138844304909818897863846150062965015390978047  
88279012794415793014165874028136590629991617898659864575994642275293219258367  
105018256636123459608064081305395253740978019398733289980008929145831504740351  
98733009314045945223835210277891436923933578161950545707623799210498884370431  
97827936581080156600455570587287352686221203725888504275566221000732487712767  
108978716401295764023852219300297489943946736978542015949441386287959493312511  
105600488318423572980443758339542451532425657100960087601934615614502775816191  
93709336830130680989932454796777994862653571446200012878444807803315316326399  
97183526240125579609323256001716450772330286757334406376129492577955363160063  
96048068021116034733527933096801133546165330358489687348325971871651409690623  
87342410741762369500890543158673242717119208033948578819523707413269716140031  
90498410708750127998190075314483107238039804374248726208931003281941400125439  
9285060377858595325327712638250293632490873327729776842761032326239888605183  
97592820274871607412814846843266341300343867494703479917579564599972997365759  
96666185963737513671554741189420053012159059988493419964641533131787578900479  
107906377154324501293618432234247388365074590242017313239451833970307612278783  
104905256699884593286219425440673613107934976133207697113217020838432496156671  
100754662641858971256686714172968889894597699736826350892218162585965806223359  
113862764975948791036282733625913361937460867100689338604333225411067913437183  
101744714253127696937706975997847671704375637793182242136692649519313680596991  
107388793798504491567908770717754018324243128666416399632804947782696412643327  
110695735519280353092624925743240181413484466124258795691962168410054001164287  
98599925397785902288838827339943112105493193138539156225576807974692137730047  
88595256963607021717663769858200838314560341988150125493318502246671471083519  
115656864176289286416338183491958854714259796995811821789078571109037553745919  
90216005735169265407139649763598079953158535753924379511544035829054986780671  
102875757550279887336264551616508415801092318345710689633638269694995326828543  
113930167123286703991204803618061640837650128708663529750618889729053275717631  
105865412081422997013268231599648241748880899227031403716893816194544519086079  
109162111729406585666957821030394530874727970145796305004829402715525545459711  
106975558967453385624731562326187640576922981191796660422826909510987874828287  
112215187973849022170849413907409013359395208006900386992982420660931088351231  
93718750706517983536870585202786656017827578544091131994577060466531508420607  
89996703216324404558416886255157406729414705945450684552910203693465623068671  
110364402338325779778162001013170204128581632682848565516170771968878155137023  
109940991127574348587814052674586377100369902030407065049127321855735774576639  
88647118907330663892043412378215140122812081182494975834595786872983860019199  
105580874917231064533671680480404236605839450980154169432643366543190772416511

111419480154081093484516355327476401363760290682381569651068743873388234145791  
91074182835435752171678661992149661206074610304132776427951959060266629464063  
97153971176844035796016788922511831888645307158320518125723870854068404486143  
106091022414347834140059004399289334934160451271044105813549828069485749207039  
93502113545835865432274558529162070126324255327025564533852411819019080826879  
92191545841757639450123848163808717283222702595935484526580160005823382159359  
114879042655149931281916799198575511221894435066718694248905827788691844628479  
99926641501820282506748151320588703395018979931287921483728676640365775683583  
91410976635815492555816795490293688444179509573057887890777645729457549344767  
107578902215616463525734600858147366293942094342894385598516854946834424004607

### Значення числа $q_1$

$q_1 = 112874119324645891945540799327642322036507848386732622064580935769023464865791$

### Кандидати, що не пройшли тест перевірки простоти для $q_1$

99013083422626350128409288854797195990115594808801599091149060299166792548351  
109909797659212072863418017975476123631969254746471325638127043974316423118847  
95645404722455929284523171369662184345946051256468816103085279713651582828543  
101205331700586809902391215240145661412750004115367658013914870732038156058623  
87018680396809924946881566208332330232253486629908386651469460255577368166399  
88408569109815896386495271603225727081633364761394626820778235366712585224191  
99562890099799194635861423037963559261592392255188240022989776814079496683519  
11333277741333304690664980686082984292131062293117602515496176172011962236927  
108344153695783615626070724902760376905323622089785569581392765576180727283711  
89863275214380327122621599574303060848036579611118650809246716747089312743423  
90310754287231411995073821543501661848585107985578619876420626357711922003967  
106174876089349601310420825558378375344076389248230110598477191368175381381119  
87568997490561218175213528634351353667873858122497959829971775626491035910143  
92286085487101111418016323608186661188687595143839495885066485207359511592959  
114391531802439456074763351772830516454270246012213837727291586876194359869439  
111263567177302897985156132254087081950651402186091015806200249594091185111039  
93667363916034526928200842123285308720095418731366397117996294665658305609727  
101274718029853249936869542569346776681609766928164761463113119068078053785599  
96726491272465475610558053237689004398113128389638078878879973548613125013503  
102847577378063136263938740699535231655496522627548857943094427930656226934783  
96199916523543774623100456560898897225505001174224666580067430975424367165439  
94552474087891897850672412922852039651811112817811012738891168344094382489599  
103212397116463890716670212509173527744663456369435048001367639273354091823103  
88409770038014312864630017179994066787380505577813037333686844879550522851327  
87232914431889667872170076450992404247400149232124404745660597386524972023807

113526801363779749639041364546315292238369367776655466807189337448639625166847  
105598766503338349098674052071100868877991867983259924971488044666474140270591  
87061865474173862700357158417684771805492059802680162729491797482811728855039  
108815898387810236368459759708238468626850019363486341756772688237890914746367  
108430687087065387099007227874593939683294971955341759985520438722088635727871  
113052154556679760924319230262957847055913687630748934828276471318323841204223  
112261882096135746189737021210063060065715783280188105448651811875753811771391  
97564098899104098297403745298674453514946116531616284885965810012448641515519  
112229817185527462692164729947382565104095753860312896395304836889029778604031  
92867661231213754526168741398609318661034617203186503757304137352002985263103  
114145746799092830949223317244844753138717612377727237766733479279199172689919  
111665745760241785288889040321891122158623767911495392800900546195513371787263  
108627966382636448666019954597060205481599703612406376497675879042113247117311  
89805008442634289505958868754373172870736479829627467136157322671220273971199  
86979308015457589275874676191506240754012169635826185686110489065996905611263  
96433221415951592413946427771725948103012451694879263144779380946539171020799  
88232466537161686473015078148789560258038722739476596569750480688456937766911  
92680911191278710067848490085775434734621767519840011514852572123251928465407  
97149241621149492193768266306068342901613840738070522366180191169526030663679  
107694007062965055116738930147753560770441777439678360289057626977302137012223  
111761030677198614765743097462845988705678280246605023808694617562963584745471  
108575478996714435779583714452102892690081014087655825017838370528740239212543  
95665338975901515640739539906825824472551642189973113148875041914169493291007  
87199667480201522488356978949916963310477080544723319204163961335005833068543  
86931540114772516754164174123294434600056302269714470770334714623038804983807  
112385303071257147462618159832027860129001486345075638187461797218113439137791  
104790056376832827106673912106198523913358233238754961653769265829403483439103  
104792255875986964555136991646980671371779475868148451337744907197765635801087  
88132437547011460161003063246351922362930633086853423089941398114149832065023  
109243466168456686411146087071243779572300187699417201580403828219705160630271  
106703881125625576324385667435688288396465082006655489003527822283347966558207  
111278132087084728422517932621018688051170922215025737785614960644880953507839  
114832751137057761179233019733592441265925008825395610823551395283742368989183  
88841439420982287778478060641120955891498283150589849766519695493414191628287  
101916853092800489864668381681435656790090529057939917112893511201115550515199  
113445122177227140016409717057024045027519081545799317648332870944062630264831  
98705601502326411502011316763274414915954920630746005847941085069542888046591  
112665436686765539096623249350823543520464278199981853936620562705099010342911  
102518659676207602919614247290295664950633937093960328763579688684392376434687  
94061936146589110121744331956661735562481607450190034959763020135360962232319

105447184946627298974173081552836371285130410633903951885166412105767723204607  
89409399409883106600229512563155705482974091890929520662710978125039929393151  
110310968303601210079163704812954740382296026530641190013342997209317164187647  
115400388027937864380809903653731752758501324958982448067772663842664068677631  
105245143379319789762700074076802257612318813057410046970517869132505461817343  
102721876790059281041994160624334590328839094128248280070338473202923572559871  
92515829982411691937598952181071258548777172053344226267491789140006413533183  
93766797616923121660324804944390109234897409925112198374010833461720937136127  
102729069177336463414134058863803050387489302808983224302728384888089599803391  
115465332299871631285299816517213041182119534862589415623812936822626002665471  
98178093361660566124043329680431158428049804400437634926789334982285143834623  
105715863308014292859453002633146949967458199458462382819402159529647411298303  
91762961211470123622156758781410624390880629270857596975056630403168617889791  
95067618181316386471924315185651746022113863109749501681892766670652180529151  
115036759568835885485256124280145170363807264217834967545219838018323438632959  
94825428408648251611790059795874543194898254181467140576174595295338404249599  
93330277248441761413668478835282372914775853609311950500721313990529955921919  
109201271015543044609397583719389140250913347719364671426775614201356504530943  
107415761646699574322992819416249274926333792287036671501964130868794314719231  
11466257391525091628874180528525544777945878903406026823065931610042178469887  
102453533224521317448807321498372724189545782710855353049133990850102269313023  
107769402487358938354754738119806161621559025323672212843569353528695712645119  
110539274727324109099217533726924262520747132852399498848169099868981577121791  
87704719223844747820468434529260535616614085225896551287668528772926631051263  
113856251796416265323433125747541308979284982222699924068801759741180377038847  
114601527182816274040857176595635545222938667482588019175467322830633262120959  
88788037754614305054038488675404566559234826521934653847716018482111387271167  
99673851625379316781917762195925077058918573093494329590815094898024679211007  
87520226885721817328907338567880113955630653149682618509357455116905094316031  
107442777303585035587964423245039541496802582835786620284367114888532931379199  
96918253558187215370623844450716739643208340907315591027172361138082309734399  
114714765129040838014665767851113660295218798348204458681759708604009302458367  
114337210553401940381248586215345181395352291968856442645433498049115947270143  
113132164423024130524861324700541901994761614219594546867231550518525100556287  
87573369636447305634810130191662880279253378877472597764884782705048037621759  
97688165580846721473336607176896791547837105552735426964639609867380937719807  
94152706151049299102154845345967452697831838458138855773058514508866595586047  
115685797338166999808875553336391719076684166999082844641625499694658805039103

### Параметри криптосистеми RSA для абонента А

n	9462095097831636066924818252030951902635633375143550149914645472073184338365857677199186649274844786918823381298433704284810517399391532373375784163737601
e	65537
d	2999306705561048074435756509570029064884762008258420599269524772828300524058291994745966730589828201164392160360124561296887281209005487815869342285317273
P	101355380124076268587266947719799178486423436012413065988713643571576687296511
q	93355627360367236277216721331318062881555725369696294768340295769881619464191

### Параметри криптосистеми RSA для абонента В

n	9962862030447587087632624336902366945145757431062767543183325110640289073800892324667756487843542231388423213184859503880488421332524367114413251615522817
e	65537
d	2201993629721123929449907129103114045507671946975665469322832357715864126127271797901798505483404724089418117842573949556990215043217963577054829359129173
p	88265247073978377837946878484926123522413904627216961245034438951970524889087
q	112874119324645891945540799327642322036507848386732622064580935769023464865791

### Чисельні значення прикладів ВТ, ШТ, цифрового підпису для А і В

А	ВТ	32928132483970987330052044412339911689184459972724647077783954592507856158720
	ШТ	6969791709667043378391415573171013419150400774760490172249971218102254550088086498346221995078434740027994908403398553525003050210801709726090365386391295
	ЦП	9325181590664658794189696255036890489817257086677498759790508247857011262652032189829308185302529451684240862485094733014826051958880763370445284004310744
В	ВТ	45900184735641708205069153162809714020347439441410191795089558925584170483711
	ШТ	7189564306336296841605501100411338500006751628862199387390177163214935949754902613622855615883559481185059691531528712382706613471731088053748492028063115
	ЦП	5954724688053983393353920653269570948763055667850050113159272013140363057324630270182688569907180541108556148949845884148229707791278614154138698755157233

Підтвердження справжності

## RSA Testing Environment

Server Key

Encryption

Decryption

Signature

Verification

Send Key

Receive Key

### Receive key

Clear

Key BE7702A9A2D96534

Signature 37DFDB704C62FDDDBA85AB10022C985BF14A3B9EB2DEB9393F96276752B6EA1847F5BF6EE96E6EC630EF

Modulus D4BD8B63336E8F7CDEF1DE89ED86A3EDD69A1EFB17B3284106FEFEC8E8E123D9E485EFC3FF398392FF8

Public exponent 10001

Receive

Key 37DFDB704C62FDDDBA85AB10022C985BF14A3B9EB2DEB9393F96276752B6EA1847F5BF6EE96E6EC630EF

Verification true

Received k1 and s1:

PairKeySignature(k=5242256003952150571933719331407066541463963742395669419144720855684495742719244945119398326639604690764807812748541084926468400700444515584157422614932213, s=57050171970150719701508050220384189094379795605929102026980852956075341913939674646695808035469841522843003758

Verification successfully!

Decrypted signature: 0926381984824204352635590896876017201986334894792870133929551149406995595893223534332409312876184236653057644744128361593161073960839253214080961910007125

Decrypted message: 13724441317039760692

Process finished with exit code 0

```
BigInteger myd = new BigInteger( val: "2797695775595802171167112510911934805599509306793885509170628158000787680318698602503524875793794
BigInteger myn = new BigInteger( val: "9149330740779545254080791049283207153421908255456630469786200478587705698854809941220692663749114
BigInteger mye = E_FOR_GENERATING_KEYPAIR;

try {
    receiveKey(new PairKeySignature(new BigInteger( val: "524225600395215057193371933140706654146396374239566941914472085568449574271924
        new BigInteger( val: "570501719701508050220384189094379795605929102026980852956075341913939674646695808035469841522843003758
        myd, myn, mye, new BigInteger( val: "111421192625201846186462335571442971576369691605448545669228452667241653615449670369260
    } catch (Exception e) {
        e.printStackTrace();
    }
}
```

### Висновок:

Ознайомилися з тестами перевірки чисел на простоту і методами генерації ключів для асиметричної криптосистеми типу RSA; практично ознайомилися з системою захисту інформації на основі криптосхеми RSA, організували з використанням цієї системи засекречений зв'язок й електронний підпис, вивчили протоколи розсилання ключів.

### Код

```
import java.math.BigDecimal;
```

```
import java.math.BigInteger;
```

```
import java.util.Random;
```

```
public class Main {
```

```
    public static void main(String[] args) {
```

```
        BigInteger p =
        generatePseudoPrimeNumber(255, 256);
```

```
        BigInteger p1 =
        generatePseudoPrimeNumber(255, 256);
```

```
        BigInteger q =
        generatePseudoPrimeNumber(255, 256);
```

```
        BigInteger q1 =
        generatePseudoPrimeNumber(255, 256);
```

```
        if
        (p.multiply(q).compareTo(p1.multiply(q1)) > 0) {
```

```
            BigInteger tempP = p;
```

```
            BigInteger tempQ = q;
```

```
            p = p1;
```



```

        q = q1;

        p1 = tempP;

        q1 = tempQ; }

System.out.println("\np = " + p);
System.out.println("q = " + q);
System.out.println("p1 = " + p1);
System.out.println("q1 = " + q1);

KeyPair keyPairA = generateKeyPair(p, q);

KeyPair keyPairB = generateKeyPair(p1,
q1);

doingGenrateMessageEncryptedSignedDecr
ypted(keyPairA, "A");

doingGenrateMessageEncryptedSignedDecr
ypted(keyPairB, "B");

        System.out.println("\n\nSending and
Received protocols:\n");

        //Emulate sending and retrieving message
BigInteger k =
generateRandomBigIntegerFromRange(new
BigInteger(String.valueOf(2)).pow(254),

        new
BigInteger(String.valueOf(2)).pow(255));

        PairKeySignature pairKeySignatureSend =
sendKey(keyPairA.getOpenKey().getE(),
keyPairA.getOpenKey().getN(),
keyPairA.getPrivateKey().getD(),

        keyPairB.getOpenKey().getE(),
keyPairB.getOpenKey().getN(), k);

        System.out.println();

        try { PairKeySignature
pairKeySignatureAfterDecryption =
receiveKey(pairKeySignatureSend,
keyPairB.getPrivateKey().getD(),

        keyPairB.getOpenKey().getN(),
keyPairA.getOpenKey().getE(),
keyPairA.getOpenKey().getN());

        } catch (Exception e) {

            System.out.println(e.getMessage());

            System.out.println("end"); }

```

```

static class PairKeySignature {

    private BigInteger k;

    private BigInteger s;

    public PairKeySignature(BigInteger k,
BigInteger s) {

        this.k = k;

        this.s = s; }

    public BigInteger getK() {

        return k; }

    public BigInteger getS() {

        return s; }

    @Override

    public String toString() {

        return "PairKeySignature{" +

            "k=" + k +

            ", s=" + s +

            '}'; }

    public static PairKeySignature
sendKey(BigInteger e, BigInteger n, BigInteger
d, BigInteger e1, BigInteger n1, BigInteger k) {

        System.out.println("Text for sending: " + k);

        BigInteger k1 = encrypt(k, e1, n1);

        System.out.println("Encrypted text for
sending: " + k1);

        BigInteger s = sign(k, d, n);

        System.out.println("Signature of sender: " +
s);

        BigInteger s1 = encrypt(s, e1, n1);

        System.out.println("Encrypted signature of
sender: " + s);

        return new PairKeySignature(k1, s1); }

    public static PairKeySignature
receiveKey(PairKeySignature pair, BigInteger d1,
BigInteger n1, BigInteger e, BigInteger n) throws
Exception { System.out.println("Recieved k1 and
s1:\n" + pair.toString());

```

```

        BigInteger k = decrypt(pair.getK(), d1, n1);

        BigInteger s = decrypt(pair.getS(), d1, n1);

        boolean verify = verify(k, s, e, n);

        if (!verify) {

            throw new Exception("Сообщение было
повреждено и его содержание восстановить
не удастся");}

        System.out.println("Verification
successfully!");

        System.out.println("Decrypted signature: "
+ s);

        System.out.println("Decrypted message: " + k);

        return new PairKeySignature(k, s);}

public static void
doingGenerateMessageEncryptedSignedDecr
ypted(KeyPair keyPair, String user) {

    System.out.println("\n" + keyPair);

    BigInteger m =
generateRandomBigIntegerFromRange(new
BigInteger(String.valueOf(2)).pow(254),
new BigInteger(String.valueOf(2)).pow(255));

    System.out.println("Message from user " +
user + " : " + m);

    BigInteger encrypted = encrypt(m,
keyPair.openKey().getE(),
keyPair.getOpenKey().getN());

    System.out.println("Encrypted text from user "
+ user + " : c = " + encrypted);

    BigInteger signature = sign(m,
keyPair.getPrivateKey().getD(),
keyPair.getOpenKey().getN());

    System.out.println("Digital signature from
user " + user + " : s = " + signature);

    System.out.println("Verification from user " +
user + " : " + verify(m, signature,
keyPair.getOpenKey().getE(),
keyPair.getOpenKey().getN())); BigInteger
decrypt = decrypt(encrypted,
keyPair.getPrivateKey().getD(),
keyPair.getOpenKey().getN());

```

```

        System.out.println("Decrypt encrypted
message from user " + user + " : " + decrypt);

        System.out.println("Звірено з
результатами роботи ресурса
http://asymcryptwebservice.appspot.com/?sect
ion=rsa, всі результати збігаються.");

    }

    public static BigInteger encrypt(BigInteger m,
BigInteger e, BigInteger n) {

        return m.modPow(e, n);

    }

    public static BigInteger decrypt(BigInteger c,
BigInteger d, BigInteger n) {

        return c.modPow(d, n);

    }

    public static BigInteger sign(BigInteger m,
BigInteger d, BigInteger n) {

        return m.modPow(d, n);

    }

    public static boolean verify(BigInteger m,
BigInteger s, BigInteger e, BigInteger n) {

        return m.equals(s.modPow(e, n));

    }

    public static BigInteger
reverseElement(BigInteger a, BigInteger n) {

        BigInteger x = new BigInteger("0"), y = new
BigInteger("1"),

        lastx = new BigInteger("1"), lasty = new
BigInteger("0"), temp;

        while (!n.equals(new BigInteger("0"))) {

            BigInteger q = a.divide(n);

            BigInteger r = a.mod(n);

            a = n;

            n = r;

            temp = x;

            x = lastx.subtract(q.multiply(x));

```

```

        lastx = temp;

        temp = y;

        y = lasty.subtract(q.multiply(y));

        lasty = temp;
    }

    //      System.out.println("Roots  x : "+ lastx +" y
    :"+ lasty);

    return lastx; }

    public static KeyPair
generateKeyPair(BigInteger p, BigInteger q) {

    KeyPair keyPair = new KeyPair(p, q);

    BigInteger n = p.multiply(q);

    BigInteger fi = p.subtract(new
BigInteger("1")).multiply(q.subtract(new
BigInteger("1")));

    BigInteger e = new
BigInteger("2").pow(16).add(new
BigInteger("1"));

    BigInteger d = reverseElement(e, fi);

    keyPair.openKey.setE(e);

    keyPair.openKey.setN(n);

    keyPair.privateKey.setD(d);

    return keyPair; }

static class KeyPair {

    private OpenKey openKey;

    private PrivateKey privateKey;

    KeyPair(BigInteger p, BigInteger q) {

        openKey = new OpenKey();

        privateKey = new PrivateKey(p, q);

    }

    static class PrivateKey {

        private BigInteger d;

        private BigInteger p;

        private BigInteger q;

```

```

        public PrivateKey(BigInteger p, BigInteger
q) {

            this.p = p;

            this.q = q;

        }

        public BigInteger getD() {

            return d; }

        public void setD(BigInteger d) {

            this.d = d; }

        public BigInteger getP() {

            return p; }

        public void setP(BigInteger p) {

            this.p = p; }

        public BigInteger getQ() {

            return q; }

        public void setQ(BigInteger q) {

            this.q = q; }

        @Override

        public String toString() {

            return "d=" + d +

                ", p=" + p +

                ", q=" + q +

                '}; }

        static class OpenKey {

            private BigInteger n;

            private BigInteger e;

            public BigInteger getN() {

                return n; }

            public void setN(BigInteger n) {

                this.n = n; }

            public BigInteger getE() {

                return e; }

            public void setE(BigInteger e) {

```

```

        this.e = e;}

@Override
    public String toString() {
        return "n=" + n +
            ", e=" + e +
            '};'
    }

    public OpenKey getOpenKey() {
        return openKey;
    }

    public PrivateKey getPrivateKey() {
        return privateKey;
    }

    @Override
    public String toString() {
        return "KeyPair{" +
            "openKey=" + openKey +
            "\nprivateKey=" + privateKey +
            '};'
    }

    public static BigInteger
    generatePseudoPrimeNumber(int
    amountMinBit, int amountMaxBit) {

        int count = 0;

        BigInteger number = null;

        do {
            count++;

            System.out.println("Attempt number : "
            + count);

            boolean satisfact = false;

            while (!satisfact) {

                number =
                generateRandomBigIntegerFromRange(new
                BigInteger(String.valueOf(2)).pow(amountMinBi
                t),

                    new
                BigInteger(String.valueOf(2)).pow(amountMaxB
                it));

                satisfact =
                checkSimpleConstraints(number);

```

```

        }

        System.out.println("number: " +
        number);

        } while
        (!millerRabinProbabilityTest(number));

        System.out.println("result of Miller-Rabin
        test for this number: " +
        millerRabinProbabilityTest(number) + ", amount
        of attempt to find = " + count);

        System.out.println("Pseudo prime
        number:" + number);

        System.out.println();

        return number;
    }

    public static BigInteger
    generateRandomBigIntegerFromRange(BigInteger
    min, BigInteger max) {

        BigDecimal minD = new BigDecimal(min);

        BigDecimal maxD = new BigDecimal(max);

        BigDecimal randomBigInteger =
        minD.add(new
        BigDecimal(Math.random()).multiply(maxD.subt
        ract(minD.add(BigDecimal.valueOf(1)))));

        return new
        BigInteger(String.valueOf(randomBigInteger.set
        Scale(0, BigDecimal.ROUND_HALF_UP)));
    }

    public static boolean
    checkSimpleConstraints(BigInteger number) {

        boolean result = true;

        String text = String.valueOf(number);

        int length = text.length();

        int sum = 0;

        for (int i = 0; i < length; i++) {

            sum += Integer.valueOf(text.substring(i, i
            + 1));

            if (text.substring(length - 1,
            length).equals("0") ||

                text.substring(length - 1,
            length).equals("2") ||

```

```

        text.substring(length - 1,
length).equals("4") ||

        text.substring(length - 1,
length).equals("5") ||

        text.substring(length - 1,
length).equals("6") ||

        text.substring(length - 1,
length).equals("8") ||

        sum % 3 == 0) {

        result = false; }

        return result; }

    public static boolean
millerRabinProbabilityTest(BigInteger p) {

        int counter = 0;

        boolean changeCounter = false;

        int k = new Random().nextInt(46) + 5;

        //Кпок 0

        BigInteger numberS1 =
p.subtract(BigInteger.valueOf(1));

        BigInteger d;

        int s = 0;

        while
(numberS1.mod(BigInteger.valueOf(2)).equals(n
ew BigInteger(String.valueOf(0)))) {

            s++;

            numberS1 =
numberS1.divide(BigInteger.valueOf(2));

        }

        d = numberS1;

        System.out.println("d = " + d + "\ns = " + s);

        while (counter < k) {

            changeCounter = false;

            //Кпок 1

            BigInteger x =
generateRandomX(BigInteger.valueOf(2), p);

```

```

        BigInteger gcd = new
BigInteger(String.valueOf(x)).gcd(new
BigInteger(String.valueOf(p)));

        if (!gcd.equals(new
BigInteger(String.valueOf(1)))) {

            return false;

        }

        //Кпок 2

        if (x.modPow(d, p).equals(new
BigInteger("1")) || x.modPow(d, p).equals(new
BigInteger("-1").mod(p))) {

            counter++;

            changeCounter = true;

        } else {

            for (int r = 1; r < s - 1; r++) {

                BigInteger xr =
x.modPow(d.subtract(new
BigInteger("2").pow(r)), p);

                if (xr.equals(new BigInteger("1"))) {

                    return false; }

                if (xr.equals(new BigInteger("-1")) ||
xr.equals(new BigInteger("-1").mod(p))) {

                    counter++;

                    changeCounter = true;

                    break; }}}

        if (!changeCounter) {

            return false;} } return true; }

    public static BigInteger
generateRandomX(BigInteger minLimit,
BigInteger maxLimit) {

        BigInteger bigInteger =
maxLimit.subtract(minLimit);

        Random randNum = new Random();

        int len = maxLimit.bitLength();

        int randomBitLength = new
Random().nextInt(len) + 1;

```

```
    BigInteger res = new  
    BigInteger(randomBitLength, randNum);  
    if (res.compareTo(minLimit) < 0)  
        res = res.add(minLimit);  
    if (res.compareTo(BigInteger) >= 0)  
        res = res.mod(BigInteger).add(minLimit);  
    return res;}}
```