

**LUA DASTURLASH TEXNOLOGIYASIDA MATEMATIK FUNKSIYALAR  
BILAN ISHLASH BANDIDA SHART APERATORLARI BILAN  
TANISHISH QISMINI O'RGANISH**

*Zokirova Nargiza Sadriddin qizi*

*Namangan davlat Universiteti, Informatika kafedrasii  
stajor o'qituvchisi*

*Ibrohimov Sanjar Rustam o'g'li*

*Namangan davlat Universiteti, Informatika kafedrasii  
stajor o'qituvchisi*

**Annotatsiya:** Bu maqolada Lua dasturlash texnologiyasi yordamidan matematik funksiyalar bilan ishlash va ularni qanday qilib dasturlarga kiritish haqida hamda matematik funksiyalar ustida amallar bajarish hamda shart operatorlari haqida ma'lumotlar ko'rsatib o'tilgan.

**Kalit so'zlar:** Matematik funksiya, matematika masalalarni yaxlitlash, matematik atamalarning Lua texnologiyasida qo'llanilishi ko'rinishlari, oddiy matematik amal.

**ТЕХНОЛОГИЯ ПРОГРАММИРОВАНИЯ LUA РАБОТА С  
МАТЕМАТИЧЕСКИМИ ФУНКЦИЯМИ В РАЗДЕЛЕ ЗНАКОМСТВА С  
УСЛОВНЫМИ ОПЕРАТОРАМИ**

*Закирова Наргиза Садриддиновна.*

*Преподаватель Наманганского государственного университета  
Иброхимов Санжар Рустамович*

*Преподаватель Наманганского государственного университета*

**Аннотация:** В этой статье показано, как работать с математическими функциями с помощью технологии программирования Lua и как включать их в программы, а также информация о выполнении операций над математическими функциями и условными операторами.

**Ключевые слова:** Математические функции, округление математических задач, примеры использования математических терминов в технологии Lua, простые математические операции.

**LUA PROGRAMMING TECHNOLOGY LUA WORKING WITH  
MATHEMATICAL FUNCTIONS IN THE SECTION OF FAMILIARITY WITH  
CONDITIONAL OPERATORS.**

*Zakirova Nargiza Sadriddin kizi*

*Teacher of Namangan State University*

*Ibrokhimov Sanjar Rustam o'gli*

*Teacher of Namangan State University*

**Annotation:** This article shows how to work with mathematical functions using Lua programming technology and how to include them in programs, as well as information about performing operations on mathematical functions and conditional operators.

**Keywords:** Mathematical functions, rounding of mathematical problems, examples of the use of mathematical terms in Lua technology, simple mathematical operations.

Ushbu kutubxona qatorlarni manipulyatsiya (moslashtirish) bajarish uchun umumiy funksiyalarni taqdim etadi, masalan, pastki satrlarni topish va chiqarish va naqsh (belgilarni) moslashtirish uchun xizmat qiladi. Lua dasturlash texnologiyasidada satrni indekslashda birinchi belgi 1-pozitsiyada (C++ dagi kabi 0da emas). Indekslar manfiy bo'lishiga ruxsat beriladi va ular qator oxiridan boshlab orqaga indekslash sifatida talqin etiladi. Shunday qilib, oxirgi belgi -1 pozitsiyasida boshlanib cheksiz davom etishi mumkin.

String kutubxonasi jadval ichidagi barcha funksiyalarini ta'minlaydi. Shuningdek, `u__index` maydon stringjadvalga ishora qiladigan satrlar uchun meta- jadvalni o'rnatadi. Shuning uchun siz string funktsiyalaridan ob'ektga yo'naltirilgan uslubda foydalanishingiz mumkin. Masalan, `string.byte(s,i)` sifatida yozilishi mumkin `s:byte(i)`.

String kutubxonasi bir baytlik belgilar kodlashini nazarda tutadi.

`string.byte (s [, i [, j]])`

`s[i], s[i+1], ...,` belgilarning ichki raqamli kodlarini qaytaradi `s[j]`. shuning uchun standart qiymat `i[1]`; uchun standart qiymat `j` hisoblanadi. Ushbu indekslar bir xil funksiya qoidalariga rioya qilgan holda tuzatiladi [string.sub](#) shartini ishlatilgan xolda.

Raqamli kodlar platformalarda ko'chma bo'lishi shart emas.

`string.char (···)`

Nol yoki undan ortiq butun sonlarni oladi. Uzunligi argumentlar soniga teng bo'lgan qatorni qaytaradi, unda har bir belgi o'zining tegishli argumentiga teng ichki raqamli kodga ega.

Raqamli kodlar platformalarda ko'chma bo'lishi shart emas.

`string.dump (function [, strip])`

Berilgan funktsiyaning *ikkilik ko'rinishini (ikkilik bo'lagi)* o'z ichiga olgan qatorni qaytaradi, shuning uchun keyingi [load](#) satr funktsiyaning nusxasini qaytaradi

(lekin yangi yuqori qiymatlar bilan). Agar striphaqiqiy qiymat bo'lsa, bo'sh joyni tejash uchun ikkilik tasvir funksiya haqidagi barcha disk raskadrovka ma'lumotlarini o'z ichiga olmaydi.

Yuqori qiymatli funksiyalar faqat yuqori qiymatlar sonini saqlaydi. (Qayta) yuklanganda, bu yuqori qiymatlar yangi misollarni oladi. ([load](#)Ushbu yuqori qiymatlar qanday ishga tushirilganligi haqida batafsil ma'lumot uchun funktsiyaga qarang. Funktsiyaning yuqori qiymatlarini o'z ehtiyojlaringizga mos keladigan tarzda seriyalashtirish va qayta yuklash uchun disk raskadrovka kutubxonasidan foydalanishingiz mumkin.)

`string.find (s, pattern [, init [, plain]])`

pattern [satrda](#) birinchi [mosligini qidiradi](#)s . Agar u moslikni topsa, bu hodisaning boshlanishi va tugashi findindekslarini qaytaradi s; aks holda, u **muvaffaqiyatsiz** qaytaradi . Uchinchi, ixtiyoriy raqamli argument initqidiruvni qaerdan boshlashni belgilaydi; uning standart qiymati 1 va salbiy bo'lishi mumkin. To'rtinchi, ixtiyoriy argument sifatida **true** qiymati plain naqshga mos keladigan moslamalarni o'chirib qo'yadi, shuning uchun funksiya oddiy "pastki qatorni topish" operatsiyasini bajaradi, hech qanday belgilar patternsehrli hisoblanmaydi.

Agar naqshda suratga olishlar bo'lsa, muvaffaqiyatli o'yinda olingan qiymatlar ikkita indeksdan keyin ham qaytariladi.

`string.format (formatstring, ...)`

Birinchi argumentda berilgan tavsifga muvofiq o'zgaruvchan argumentlar sonining formatlangan versiyasini qaytaradi, bu qator bo'lishi kerak. Format satri ISO C funksiyasi bilan bir xil qoidalarga amal qiladi sprintf. Faqat farqlar aylantirish ma'lumoti va nitelemeler bor \*, h, L, l, va nqo'llab-quvvatlanmaydi va qo'shimcha belirticisi borligini q.

Belirticisi q natija Lua kodida bir amal doimiy ekanligini bir tarzda formatlar Bağlaçlar, nol, sonlar va belgilar qatori bo'ladi. Mantiqiy va nil aniq shaklda yoziladi ( true, false, nil). Floats to'liq aniqlikni saqlash uchun o'n oltilik tizimda yoziladi. Lua tarjimoni tomonidan xavfsiz o'qilishini ta'minlash uchun kerak bo'lganda qochish ketma-ketliklaridan foydalangan holda qator qo'sh tirnoq orasiga yoziladi. Masalan, qo'ng'iroq

`string.format('%q', "'tirnoq" va \n yangi qatorli satr')`  
qatorni ishlab chiqishi mumkin:

"\" tirnoq \" va \" bilan bir qator yangi qator"

Bu spetsifikator modifikatorlarni (bayroqlar, kenglik, uzunlik) qo'llab-quvvatlamaydi.

A, a, E, e, f, G, va konversiya spetsifikacilari gargument sifatida raqamni kutadi. Ma'lumoti c, d, i, o, u, X, va x tamsayı kutish. Lua C89 kompilyatori bilan kompilyatsiya qilinganda, spetsifikatsiyalar Avaa (o'n oltilik suzuvchilar) modifikatorlarni qo'llab-quvvatlamaydi.

Belirticisi bir qator kutadi; agar uning argumenti satr bo'lmasa, u bir xil qoidalarga rioya qilgan holda o'zgartiriladi [tostring](#). Agar spetsifikatsiyada biron bir modifikator bo'lsa, tegishli qator argumenti o'rnatilgan nollarni o'z ichiga olmaydi.

Belirticisi ptomonidan qaytib ko`rsatgichini formatlaydi [lua\\_topointer](#). Bu jadvallar, foydalanuvchi ma'lumotlari, iplar, satrlar va funksiyalar uchun noyob satr identifikatorini beradi. Boshqa qiymatlar uchun (raqamlar, nil, mantiqiy qiymatlar) bu ko'rsatgich ko'rsatgichni ifodalovchi qatorga olib keladi NULL.

[string.gmatch \(s, pattern \[, init\]\)](#)

Har safar chaqirilganda satr ustidagi pattern keyingi suratlarni qaytaradigan iterator funksiyasini qaytaradi s. Agar patternqo'lga kiritilmagan bo'lsa, har bir qo'ng'iroqda butun o'yin ishlab chiqariladi. Uchinchi, ixtiyoriy raqamli argument initqidiruvni qaerdan boshlashni belgilaydi; uning standart qiymati 1 va salbiy bo'lishi mumkin.

Misol tariqasida, quyidagi tsikl string dan barcha so'zlarni takrorlaydi va har sbir satrda bittadan chop etadi:

```
s = "Luadan salom dunyo"
string.gmatch(s, "%a+") da w uchun
  chop etish (w)
oxiri
```

Keyingi misol berilgan satrdagi barcha juftlarni key=valuejadvalga to'playdi :

```
t = {}
s = "dan=dunyoga, to=Lua"
k, v uchun string.gmatch(s, "(%w+)=(%w+)") bajaring
  t[k] = v
oxiri
```

Bu funksiya uchun naqsh boshidagi ' ' belgisi langar sifatida ishlamaydi, chunki bu iteratsiyani oldini oladi.

`string.gsub (s, pattern, repl [, n])`

Satr , jadval yoki funksiya bo'lishi mumkin s bo'lgan barcha (yoki naga berilgan bo'lsa , birinchisi ) pattern tomonidan belgilangan almashtirish qatori bilan almashtirilgan nusxasini qaytaradi repl. gsubshuningdek, ikkinchi qiymat sifatida, sodir bo'lgan o'yinlarning umumiy sonini qaytaradi. Nomi gsubkelgan *Global narsa* .

Agar replsatr bo'lsa, uning qiymati almashtirish uchun ishlatiladi. Belgi %qochish belgisi sifatida ishlaydi: 1 dan 9 gacha bo'lgan drepl shaklidagi har qanday ketma-ketlik d -chi olingan pastki qatorning qiymatini bildiradi ; ketma-ketlik butun o'yinni anglatadi; ketma-ketlik bittani bildiradi . %d%0%% %

Agar repljadval bo'lsa, u holda jadval har bir moslik uchun so'raladi, kalit sifatida birinchi qo'lga olish.

Agar replfunksiya bo'lsa, u holda bu funksiya har safar mos kelganda chaqiriladi va olingan barcha pastki qatorlar tartibda argument sifatida uzatiladi.

Har qanday holatda, agar naqsh hech qanday suratga olishni belgilamasa, u o'zini butun naqsh suratga olishning ichida bo'lgandek tutadi.

Agar jadval so'rovi yoki funksiya chaqiruvi tomonidan qaytarilgan qiymat satr yoki raqam bo'lsa, u holda o'rnini bosuvchi qator sifatida ishlatiladi; aks holda, agar u **noto'g'ri** yoki **nil bo'lsa** , unda almashtirish yo'q (ya'ni, asl moslik satrda saqlanadi).

Mana bir nechta misollar:

```
x = string.gsub("salom dunyo", "(%w+)", "%1 %1")
--> x="salom salom dunyo dunyosi"
```

```
x = string.gsub("salom dunyo", "%w+", "%0 %0", 1)
--> x="salom salom dunyo"
```

```
x = string.gsub("Luadan salom dunyo", "(%w+)%s*(%w+)", "%2 %1")
--> x="dunyodan salom Lua"
```

```
x = string.gsub("uy = $HOME, foydalanuvchi = $USER", "%$(%w+)",
os.getenv)
```

```
--> x="uy = /home/roberto, foydalanuvchi = roberto"
```

```
x = string.gsub("4+5 = $return 4+5$", "%$(.)%", funksiya(lar)
```

```
  Qaytish yuk(lar)i()
```

```
  oxiri)
```

```
--> x="4+5 = 9"
```

```
mahalliy t = {ism = "lua", versiya = "5.4"}
```

```
x = string.gsub("$name-$version.tar.gz", "%$(%w+)", t)
```

```
--> x="lua-5.4.tar.gz"
```

[string.len \(s\)](#)

Satni qabul qiladi va uning uzunligini qaytaradi. Bo'sh satr ""uzunligi 0 ga ega. O'rnatilgan nollar hisoblanadi, shuning "a\000bc\000" uchun uzunligi 5 ga teng.

[string.lower \(s\)](#)

Satni qabul qiladi va barcha katta harflar kichik harflarga almashtirilgan holda ushbu satrning nusxasini qaytaradi. Boshqa barcha belgilar o'zgarishsiz qoldiriladi. Katta harf nima ekanligini aniqlash joriy mahalliy tilga bog'liq.

[string.match \(s, pattern \[, init\]\)](#)

Birinchi qidiradi *o'yinda* ning pattern. Agar u bitta topsa match, naqshdan olingan suratlarni qaytaradi; aks holda qaytaradi **muvaqqiyatsiz**. Agar pattern qo'lga kiritilmagan bo'lsa, butun o'yin qaytariladi. Uchinchi, ixtiyoriy raqamli argument init qidiruvni qaerdan boshlashni belgilaydi; uning standart qiymati 1 va salbiy bo'lishi mumkin.

[string.pack \(fmt, v1, v2, ...\)](#)

Qaytadi qadriyatlarini o'z ichiga olgan ikkilik string v1, v2 va boshqalar format kanop ko'ra (qadoqlangan) ikkilik shaklida serileştirilir fmt.

[string.packsize \(fmt\)](#)

[string.pack](#) Berilgan formatdan olingan satr hajmini qaytaradi. Format satrida o'zgaruvchan uzunlikdagi "s" yoki "z" variantlari bo'lishi mumkin emas.

[string.rep \(s, n \[, sep\]\)](#)

Satr bilan ajratilgan nqator nusxalarining birlashmasi bo'lgan qatorni qaytaradi . Standart qiymat bo'sh satrdir (ya'ni, ajratuvchi yo'q). Agar ijobiy bo'lmasa, bo'sh qatorni qaytaradi . ssepsepn

(E'tibor bering, ushbu funktsiyaga bitta qo'ng'iroq qilish bilan mashinangiz xotirasini tugatish juda oson.)

`string.reverse (s)`

Satr steskari bo'lgan satrni qaytaradi .

`string.sub (s, i [, j])`

dan sboshlanadigan i va gacha davom etadigan pastki qatorni qaytaradi j; iva jsalbiy bo'lishi mumkin. Agar jyo'q bo'lsa, u -1 ga teng deb qabul qilinadi (bu qator uzunligi bilan bir xil). Xususan, chaqiruv with length string.sub(s,1,j)prefiksini qaytaradi va (musbat uchun ) with length qo'shimchasini qaytaradi . sjstring.sub(s, -i)isi

Agar manfiy indeksni tarjima qilgandan keyin il dan kichik bo'lsa, u 1 ga tuzatiladi. Agar jsatr uzunligidan katta bo'lsa, u shu uzunlikka tuzatiladi. Agar ushbu tuzatishlardan keyin idan katta bo'lsa j, funktsiya bo'sh qatorni qaytaradi.

`string.unpack (fmt, s [, pos])`

Qaytaring kanop joylangan qadriyatlar s(qarang [string.pack](#)format satrlari ko'ra) fmt. Majburiy emas pos, o'qishni qaerdan boshlash kerakligini belgilaydi s(standart 1). O'qilgan qiymatlardan so'ng, bu funktsiya birinchi o'qilmagan bayt indeksini ham qaytaradi s.

`string.upper (s)`

Satrni qabul qiladi va barcha kichik harflar katta harfga almashtirilgan holda ushbu qatorning nusxasini qaytaradi. Boshqa barcha belgilar o'zgarishsiz qoldiriladi. Kichik harf nima ekanligini aniqlash joriy mahalliy tilga bog'liq.

Lua tilidagi naqshlar oddiy satrlar bilan tavsiflanadi, ular naqsh moslashtirish funktsiyalari orqali naqsh sifatida talqin etiladi [string.find](#), [string.gmatch](#), [string.gsub](#), va [string.match](#). Ushbu bo'limda ushbu satrlarning sintaksisi va ma'nosi (ya'ni, ular nimaga mos kelishi) tavsiflanadi.

Belgilar sinfi:



A *belgi sinf* belgilar majmuini ifodalaydi uchun ishlatiladi. Belgilar sinfini tavsiflashda quyidagi kombinatsiyalarga ruxsat beriladi:

• **x** : (bu erda *x* *sehrli* belgilardan biri emas)  $\$()%.[]*+-$ ?) *x* belgisining o'zini ifodalaydi.

- **.** : (nuqta) barcha belgilarni ifodalaydi.
- **%a** : barcha harflarni ifodalaydi.
- **%c** : barcha boshqaruv belgilarini ifodalaydi.
- **%d** : barcha raqamlarni ifodalaydi.
- **%g** : bo'sh joydan tashqari barcha chop etiladigan belgilarni ifodalaydi.
- **%l** : barcha kichik harflarni ifodalaydi.
- **%p** : barcha tinish belgilarini ifodalaydi.
- **%s** : barcha bo'sh joy belgilarini ifodalaydi.
- **%u** : barcha bosh harflarni ifodalaydi.
- **%w** : barcha alfanumerik belgilarni ifodalaydi.
- **%x** : barcha o'n oltilik raqamlarni ifodalaydi.
- **%x** : (bu erda *x* har qanday harf-raqamli bo'lmagan belgi) *x* belgisini ifodalaydi . Bu *sehrli* belgilardan qochishning standart usuli. Har qanday alfanumerik bo'lmagan belgi (shu jumladan, barcha tinish belgilari, hatto *sehrli* bo'lmagan belgilar) %o'zini naqshda ko'rsatish uchun " " belgisini qo'yishi mumkin .

• **[set]** : *to'plamdagi* barcha belgilar birlashmasi bo'lgan sinfni ifodalaydi . Belgilar diapazoni diapazonning oxirgi belgilarini ortib boruvchi tartibda “ -” bilan ajratish orqali belgilanishi mumkin . Yuqorida tavsiflangan barcha sinflar %*x* ham *to'plamdagi* komponentlar sifatida ishlatilishi mumkin . Barcha boshqa belgilar *to'plami* o'zlarini vakili. Masalan, [%w\_](yoki [\_%w]) barcha alfanumerik belgilar va pastki [0-7]chiziqni ifodalaydi, sakkizlik raqamlarni va [0-7%1%-]sakkizlik raqamlarni va ' -' belgisini ifodalaydi.

Yopuvchi kvadrat qavsni to'plamdagi birinchi belgi sifatida joylashtirish orqali to'plamga qo'yishingiz mumkin. Siz uni to'plamdagi birinchi yoki oxirgi belgi sifatida joylashtirish orqali to'plamga defis qo'yishingiz mumkin. (Shuningdek, ikkala holatda ham qochishdan foydalanishingiz mumkin.)

Diapazonlar va sinflar o'rtasidagi o'zaro ta'sir aniqlanmagan. Shuning uchun, naqshlar yoqadi [%a-z]yoki [a-%] hech qanday ma'noga ega emas.

• **[^set]** : *to'plamning* to'ldiruvchisini ifodalaydi , bu erda *to'plam* yuqoridagi kabi izohlanadi.



Bitta harflar ( %a, %c, va hokazo) bilan ifodalangan barcha sinflar uchun mos keladigan bosh harf sinfning to'ldiruvchisini ifodalaydi. Masalan, %Sbo'sh joy bo'lmagan barcha belgilarni ifodalaydi.

Harf, bo'sh joy va boshqa belgilar guruhlarini ta'riflari joriy lokalizatsiyaga bog'liq. Xususan, sinf ga [a-z]ekvivalent bo'lmasligi mumkin %l.

### Naqshli element:

A *namuna element* bo'lishi mumkin

- sinfdagi istalgan bitta belgiga mos keladigan bitta belgi sinfi;
- sinfdagi \*nol yoki undan ortiq belgilar ketma-ketligiga mos keladigan bitta belgi sinfi, undan keyin " ". Ushbu takrorlash elementlari har doim mumkin bo'lgan eng uzun ketma-ketlikka mos keladi;
- sinfdagi +bir yoki bir nechta belgilar ketma-ketligiga mos keladigan bitta belgi sinfidan keyin " ". Ushbu takrorlash elementlari har doim mumkin bo'lgan eng uzun ketma-ketlikka mos keladi;
- bitta belgi sinfidan keyin ' -', bu sinfdagi nol yoki undan ortiq belgilar ketma-ketligiga ham mos keladi. ' \*' dan farqli o'laroq , bu takrorlash elementlari har doim mumkin bo'lgan eng qisqa ketma-ketlikka mos keladi;
- bitta belgi sinfidan keyin ' ?' belgisi bo'lib, u sinfdagi belgilarning nolga yoki bitta ko'rinishiga mos keladi. Agar iloji bo'lsa, u har doim bitta hodisaga mos keladi;
- %n, 1 dan 9 gacha bo'lgan n uchun ; bunday element qo'lga kiritilgan n -chi qatorga teng bo'lgan pastki qatorga mos keladi (pastga qarang);
- %bxy, bu erda x va y ikkita alohida belgidir; Bunday element bilan boshlash skriptlarini o'yinlari x bilan oxirida y , qaerda va x va y etiladi *muvozanatli* . Bir sanab, chapdan o'ngga satrni o'qiydi bo'lsa, bu vosita +1 bir uchun x va -1 bir uchun y , so'nggi y birinchi y soni Masalan 0. etib, element %b()muvozanatli Qavslar ifodalarini o'yinlari .
- %f[set], *chegara namunasi* ; bunday element bo'sh satrga istalgan pozitsiyada mos keladi, shunda keyingi belgi *to'plamga* tegishli bo'ladi va oldingi belgi *to'plamga* tegishli emas . Belgilangan *to'plami* ilgari tasvirlangan sifatida talqin etiladi. Mavzuning boshi va oxiri xuddi " \0" qahramoni kabi ishlanadi .

### Naqsh:

A *o'rnak* o'rnak unsurlar bir qator. ^Naqsh boshidagi ' ' karetsi mavzu satrining boshidagi moslikni bog'laydi. \$Naqsh oxiridagi ' ' belgisi mavzu satrining oxiridagi

moslikni bog'laydi. Boshqa pozitsiyalarda " ^" va " \$" alohida ma'noga ega emas va o'zlarini ifodalaydi.

### Suratga olish:

Naqsh qavs ichiga olingan pastki naqshlarni o'z ichiga olishi mumkin; *qo'lga* olishlarni tasvirlaydilar. Muvaffaqiyatli mos kelsa, tasvirga mos keladigan mavzu satrining pastki qatorlari kelajakda foydalanish uchun saqlanadi (*qo'lga olinadi*). Qo'lga olishlar chap qavslarga ko'ra raqamlangan. Misol uchun, naqshda "(a\*(.)%w(%s\*))"satrning mos keladigan qismi "a\*(.)%w(%s\*)"birinchi suratga olish sifatida saqlanadi va shuning uchun 1 raqamiga ega; "." ga mos keladigan belgi 2 raqami bilan, "%s\*" mos keladigan qism esa 3 raqamiga ega.

Maxsus holat sifatida ()suratga olish joriy satr holatini (raqamni) oladi. Biz naqsh amal Misol uchun, agar "()aa()"kanop kuni "flaaap"3 va 5: ikki egallashga mavjud bo'ladi.

### Bir nechta moslik:

Funktsiya [string.gsub](#) va iterator [string.gmatch](#) mavzudagi berilgan naqshning bir nechta takrorlanishiga mos keladi. Bu funksiyalar uchun yangi moslik avvalgi moslik tugaganidan keyin kamida bir bayt tugasagina haqiqiy hisoblanadi. Boshqacha qilib aytganda, naqsh mashinasi hech qachon bo'sh satrni boshqa o'yindan keyin darhol o'yin sifatida qabul qilmaydi. Misol sifatida, quyidagi kod natijalarini ko'rib chiqing:

```
> string.gsub("abc", "()a*()", chop etish);
-> 1 2
-> 3 3
-> 4 4
```

Ikkinchi va uchinchi natijalar Lua ning b' ' dan keyin bo'sh qatorga va ' c' dan keyin boshqasiga mos kelishidan kelib chiqadi . Lua ' a' dan keyin bo'sh qatorga mos kelmaydi , chunki u oldingi o'yinning bir xil pozitsiyasida tugaydi.

### Adabiyotlar:

1. “Lua dasturlash texnologiyasi” qo'llanma N.Zokirova G.Yunusova
2. Dasturlash texnologiyalari Nazarov Sh G.Ivanova, S.Gaynazarov

### Internet manbalar:

1. <https://www.youtube.com/watch?v=5UCg7ca8ogE>
2. <https://www.youtube.com/watch?v=SBL0POpUYvQ>