

9. Pythonda siklik operatorlar.



Takrorlanuvchi algoritm yoki dasturning bir necha marta (shu jumladan nol marta) takrorlanishi mumkin bo'lgan qismidir. Har bir siklik tashkil etuvchilar sikl sarlavhasi bilan boshlanadi va yakuniy tasdiq bilan tugaydi. Ularning orasida "sikl tanasi" deb nomlangan operatorlar mavjud. Siklning tanasini tashkil etuvchi buyruqlar (operatorlar) bajarilishining takroriy soni siklni tugatish sharti bilan belgilanadi. To'xtatish sharti sikl parametrlari (sikl o'zgaruvchisi) deb nomlangan maxsus o'zgaruvchining ma'lum bir qiymatiga erishishi yoki ma'lum bir shartning bajarilishi (bajarilishini tugatish) bo'lishi mumkin.

Dasturlash tillarida parametrli sikllarni tashkil qilish uchun FOR ("uchun") operatoridan foydalaniladi va shartli sikllarda WHILE ("aslo") aralash operatoridan foydalaniladi.

Parametrli sikl holatida siklning takrorlanish soni ("takrorlanishlar") oldindan ma'lum va sikl sarlavhasida maxsus ifoda bilan belgilanadi,

shartli sikl bo'lsa, har bir keyingi takrorlash bilan siklning tugatish holatini tekshirish kerak.

Agar sikldan tanasida operatorlarni yozish paytida xatolik yuz bersa, siklni tugatish sharti hech qachon bajarilmaydi va sikl cheksiz bo'ladi ("dastur Takrorlanuvchi").

Bir o'lchovli massivlar bilan ishlash uchun parametrli sikllardan foydalanish tavsiya etiladi, chunki takroriy sonlarni sikl boshlanishidan oldin aniqlash mumkin. Bunday holda, massiv elementlarini kiritish uchun parametrli sikl talab qilinadi va sikllar ushbu elementlar bilan har qanday harakatlarni bajarish va natijalarni namoyish qilish uchun ham talab qilinishi mumkin.

Sikllarni biron bir hodisa ro'y bermaguncha ma'lumot kiritiladigan (ba'zi bir manbadan kelgan) vaziyatda ishlatish tavsiya etiladi. Bundan tashqari, barcha ishlov berishlarni ko'pincha massiv yaratmasdan bajarish kerak, chunki elementlarning soni oldindan ma'lum emas.

Oddiy vazifalarni ko'rib chiqing, ularning yechimi siklda hisoblashni talab qiladi.

1 topshiriq. Berilgan A bir o'lchovli massiv N elementlari sonlardan iborat. Massiv elementlarining o'rta arifmetik qiymatini toping.

Masalani qo'yilishi:

Berilgan:

- N - massivdagi elementlar soni;
- i - massiv elementining indeksleri (sikl parametri).
- $A[i]$ -massiv elementi;

Aniqlash:

- S - massiv elementlarining yig'indisi
- C - massiv elementlarining o'rta arifmetik qiymati, $C = S / N$.

Dasturning matni:

kirish N

$S = 0$

bs uchun 1 dan N gacha

A [i] kiritish

$S = S + A [i]$

ys

$C = S / N$

chiqarish C

Bu erda bs va ys mos ravishda siklning boshini va oxirini bildiradi, bs bilan chiziq siklning sarlavhasini anglatadi. Matndan ko'rinib turibdiki, sikl o'zgaruvchisining boshlang'ich va yakuniy qiymatlari ko'rsatilgan, ular butun son bo'lishi kerak. Bu erda ko'rsatilgan izohda, sikl o'zgaruvchisi har bir takrorlash uchun 1 ga ko'paytiriladi ("sikl o'zgaruvchan qadam" 1). Agar 1 ga teng bo'lmagan qadam kerak bo'lsa, bu aniq ko'rsatiladi

Sikl tanasi ikkita operatordan iborat - keyingi raqamni kiritish va ushbu raqamni summaning joriy qiymatiga qo'shish.

Python-da siz deyarli bir xil narsani yozishingiz mumkin (range () funksiyasidan foydalanish bilan bog'liq xususiyatlarni hisobga olgan holda).

- * - kodlash: utf-8 - * -

#

N = int(input ('**Elementlar soni:**'))

S = 0

for i **in** range (N):

 a =int(input ('**Raqamni kiriting:**'))

 S = S + a

C = S / N

print ('**Natija :** ', C)

Elementlar soni:3

Raqamni kiriting:2

Raqamni kiriting:4

Raqamni kiriting:6

Natija : 4.0

Takrorlanuvchi tanasi ":" belgisidan keyin boshlanadi va Python-dagi barcha takrorlanish tanasi operatorlari satr boshidan bir xil bo'sh joy qoldirilishi kerak. Bo'shliq tugagandan Python sikl tanasi tugadi deb hisoblaydi.

Va bu erda ro'yxat va ro'yxat usullaridan foydalanib Python-da xuddi shu vazifani hal qilish uchun variant mavjud.

```
# - * - kodlash: utf-8 - * -
```

```
#
```

```
N = int(input ( 'Elementlar soni:'))
```

Elementlar soni:3

```
S = 0
```

Raqamni kiriting:2

```
lst = []
```

```
for i in range (N):
```

Raqamni kiriting:4

```
    a = int(input ( 'Raqamni kiriting:'))
```

Raqamni kiriting:6

```
    lst.append (a)
```

```
C = sum (lst) / N
```

Natija: 4.0

```
print ('Natija: ', C)
```

Ushbu parametrda ro'yxat tuziladi va ro'yxat elementlarining yig'indisi o'rnatilgan funksiyadan foydalanib hisoblanadi. Dastur bitta qatorga ko'paydi (bo'sh ro'yxatni yaratish), ammo biz qanday qilib sikl qanday qilib ro'yxat tuzishni bilib oldik.

2 topshiriq. Kiritilgan satr ABBA, kazak va boshqalar tipidagi palindrom (shakl o'zgartiruvchi) ekanligini aniqlang.

Masalani qo'yilishi: S satrni boshidan va oxiridan juft-juft taqqoslash talab etiladi (birinchi va oxirgi, ikkinchi va penultimate va boshqalar). Agar har bir bunday juftlik bir xil belgilarga ega bo'lsa, satr palindromdir. Shunga ko'ra, har bir belgi juftligini tekshirish ba'zi bir atributlarni olishi kerak (flag - "bayroq"), bu juftlikdagi belgilar mos keladigan bo'lsa 1 ga va agar ular bir-biriga mos kelmasa 0 ga teng bo'ladi. Satrni qayta ishlashning yakuniy natijasi "bayroqlar" ning barcha qiymatlarining mahsulidir. Agar hech bo'lmaganda bir marta "bayroq" nolga teng bo'lsa, satr palindrom emas

va barcha "bayroqlar" ning mahsuloti 0 ga teng bo'ladi. Juftlar soni L satr uzunligining yarmidan oshmaydi (teng belgilar soni bo'lgan satrlar uchun uzunlikning yarmiga teng va satr uzunligining butun sonini bo'lish natijasi) toq sonli belgilar bilan chiziqlar uchun 2 ga teng, chunki toq belgilar soni bo'lgan chiziqning "markaziy" belgisi aniq bir xil).

Dasturning matni:

kirish S

flag = 1

L = uzunlik (S)

N = L div 2

bs i uchun 1 dan N gacha

agar $S[i] = S[L-i + 1]$ bo'lsa

k = 1

aks holda

k = 0

agar tamom

flag = flag * k

ys

agar flag = 1 bo'lsa

"Palindrom" chiqishi

aks holda

"Palindrom emas!" xulosasi

agar tugadi

Har bir juftlikni tekshirishda k koeffitsient belgilanadi, u keyinchalik "bayroq" ning joriy qiymatiga ko'paytiriladi. Yakuniy xulosa "bayroq" ning umumiy qiymati bo'yicha amalga oshiriladi.

Python dasturidagi matn soxta til matniga juda o'xshash bo'lishi mumkin.

```
# - * - kodlash: utf-8 - * -
```

```
#
```

```
s1 = input( 'Manba qatori:' )
```

```
# Qator uzunligini aniqlang
```

```
L = len (s1)
```

```
flag=1
```

```
for i in range(L//2):
```

```
    if s1[i]==s1[-i-1]:
```

```
        k=1
```

```
    else:
```

```
        k=0
```

```
    flag=flag*k
```

```
if flag==1:
```

```
    print ("Palindrom")
```

```
else:
```

```
    print ("Palindrom emas!")
```

Manba qatori: abba

Palindrom

Satr kiritish uchun input () operatoridan foydalanilgan va satrni qo'shtirnoq ichiga yozish shart emas.

Shunga qaramay, ba'zi bir kichik sintaktik xususiyatlar mavjud - ikkita o'zgaruvchining tengligi sharti "==" belgisi bilan yozilgan, har bir aralash gapning boshi ":" belgisi bilan ko'rsatilgan va har doimgidek, siz yozuvni belgilashingiz kerak. Bunga qo'shimcha ravishda, chiziq oxiridagi belgilarni hisoblash uchun chiziq elementlarining "salbiy" ko'rsatkichlari qo'llaniladi.

Biroq, Python-da satrli funksiyalardan foydalanish, ularning funksiyalari va usullari bizga ushbu muammoni yanada oqilona tarzda hal qilishga imkon beradi. Masalan, shunga o'xshash.

```
# - * - kodlash: utf-8 - * -
```

```
#
```

```
s1 = input ( 'Manba qatori:')
```

```
lst = list (s1)
```

```
lst.reverse ()
```

```
s2 = ''.join(lst)
```

```
if s1 == s2:
```

```
    print(" Palindrom ")
```

```
else:
```

```
    print (" Palindrom emas! ")
```

Manba qatori:abba

Palindrom

Agar operatsiyalarni takrorlash soni oldindan ma'lum bo'lmasa, lekin operatsiyalarni to'xtatish sharti ma'lum bo'lsa, WHILE (murakkab operator) sikl ishlatiladi. Biz undan foydalanishni quyidagi misolda ko'rsatamiz.

3 topshiriq. Nolga teng bo'lmagan raqamlar ketma-ket ravishda kiritiladi. Musbat va manfiy sonlarning yig'indisini aniqlang. 0 raqamini kiritganingizdan so'ng raqamlarni kiritishni tugating.

Vazifa shunchalik osonki, muammoning bayoni sifatida qo'shimcha aniqlashtirish talab qilinmaydi. Musbat sonlarning yig'indisi SP, manfiy sonlarning yig'indisi esa SN deb aytilsin.

Algoritmning blok sxemasi [rasm. 3.13](#).

Dasturning matni:

SP = 0

SN = 0

kiritish son

bs agar son <> 0

agar son > 0 bo'lsa

SP = SP + son

aks holda

SN = SN + son

end agar

kiritish son

ys

SP chiqishi

SN chiqishi

Paskal va BASIC dasturlash tillaridagi "tengsizlik" holati " <> " deb belgilanadi, shuning uchun bu erda bu belgi saqlanadi.

Shuni ta'kidlash kerakki, tekshirilgan raqam sikl boshlanishidan oldin aniqlanishi kerak, chunki noma'lum qiymat 0 ga aylanishi mumkin va dastur ishga tushmasdan oldin tugaydi. Va keyin raqamlar siklga kiritiladi va har bir yangi qabul qilingan raqam 0 bilan taqqoslanadi (har bir raqam kiritilgandan so'ng, shart tekshiruvi kuzatiladi). WHILE Takrorlanuvchiidagi operatsiyalar va tekshirishlar tartibi to'g'ri natijaga erishish uchun muhim bo'lishi mumkin.

Python-dagi dastur matni muhim xususiyatlarga ega emas. O'qish qulayligi uchun dastur sharh belgisini ishlatgan holda "blokлар" ga bo'linadi.

```
# - * - kodlash: utf-8 - * -
#
SP = 0
SN = 0
#
son = int(input ( 'Keyingi raqam:'))
#
while son != 0:
    if son>0:
        SP = SP + son
    else:
        SN = SN + son
    son = int(input ( 'Keyingi raqam :'))
#
print ('Musbat sonlar yig`indisi: ', SP)
print ('Manfiy sonlar yig`indisi: ', SN)
```

Keyingi raqam:4

Keyingi raqam :5

Keyingi raqam :-2

Keyingi raqam :-4

Keyingi raqam :7

Keyingi raqam :1

Keyingi raqam :0

Musbat sonlar yig`indisi: 17

Manfiy sonlar yig`indisi: -6

4 topshiriq. for operatori va input() funktsiyasini jamlab, ro'yxatni foydalanuvchidan olingan qiymatlar bilan to'ldiring.

```
dostlar = [] # bo'sh ro'yxat

print("5 ta eng yaqin do'stingiz kim?")

for n in range(5):

    # n bu yerda 0 dan 4 gacha qiymatlar oladi

    dostlar.append(input(f"{n+1}-do'stingizning ismini kiriting: "))

print(dostlar)
```

5 ta eng yaqin do'stingiz kim?

1-do'stingizning ismini kiriting: Aziz

2-do'stingizning ismini kiriting: Salim

3-do'stingizning ismini kiriting: Karim

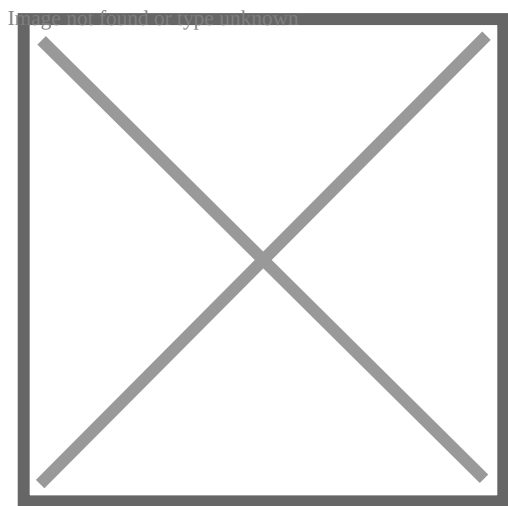
4-do'stingizning ismini kiriting: Toir

5-do'stingizning ismini kiriting: Zoir

['Aziz', 'Salim', 'Karim', 'Toir', 'Zoir']

5 topshiriq. N natural son berilgan $1+2+3+\dots+n$ hisoblang.

Python dasturi:

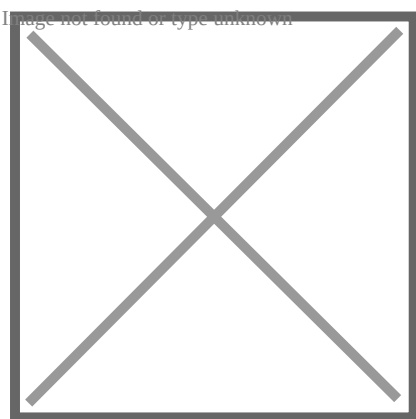


N natural son kiriting

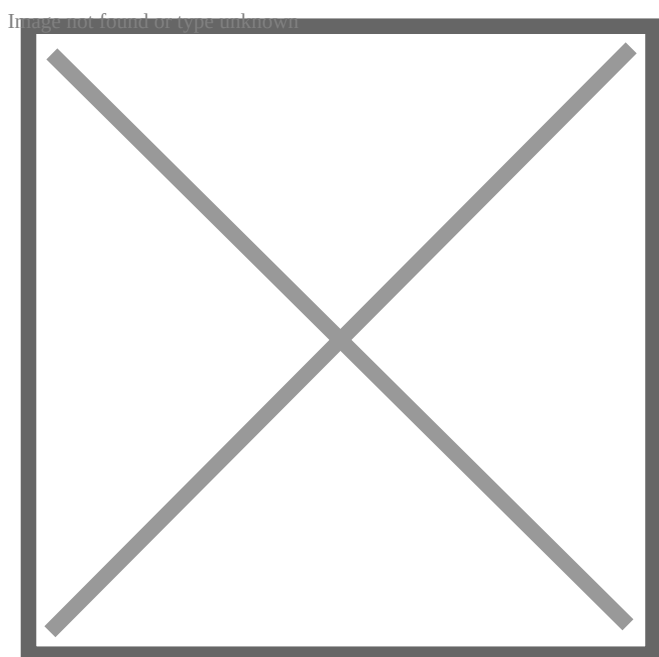
$n=50$

Yig'indi qiymati= 1225

6 topshiriq. N va x natural sonlar berilgan. Quyidagi yig'indini hisoblang:



Python dasturi:



n natural son qiymatini kiriting

$n=4$

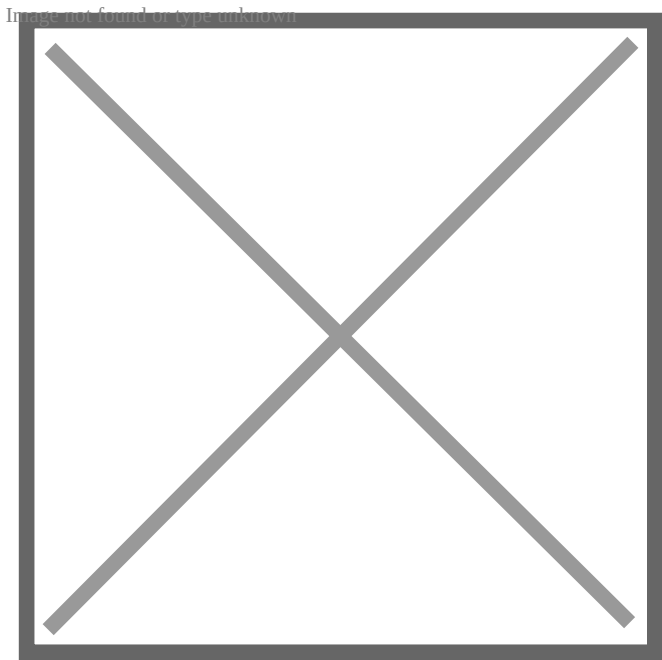
x ning qiymatini kiriting

$x=1$

Yig'indi qiymati $S= 2.00547186266$

7 topshiriq. Ixtiyoriy berilgan N natural sonning raqamlar yig'indisi nechaga teng.

Python dasturi:



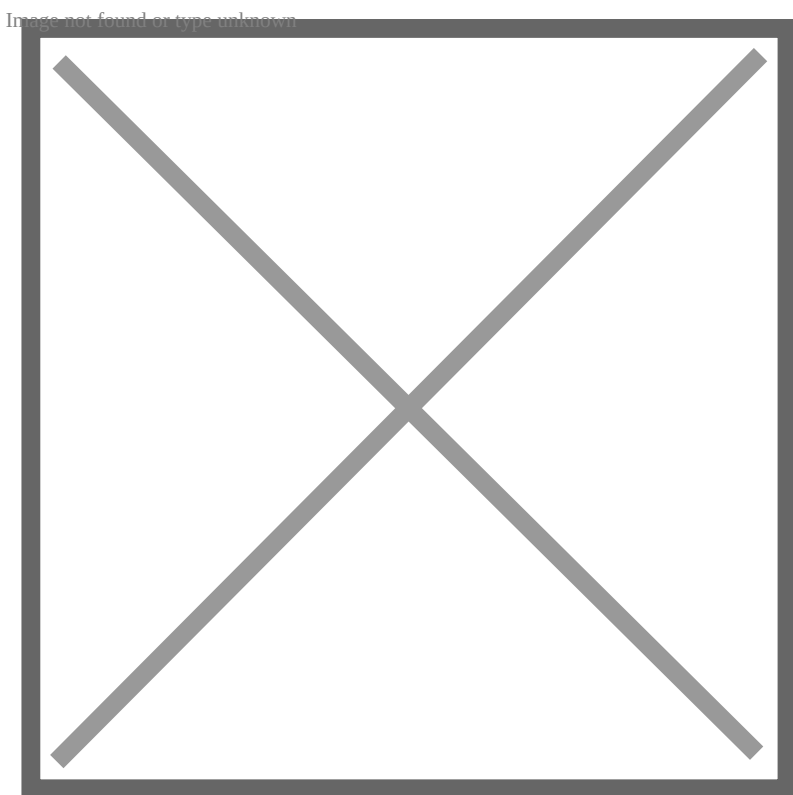
n natural son qiymatini kiritin

n=4789

Yig'indi qiymati S= 28

8 topshiriq. a natural soniga ko'paytirish karra jadvalini chiqaruvchi dastur tuzing.

Python dasturi:



Karra jadvalini hisoblash son

a=6

$6 \times 1 = 6$

$6 \times 2 = 12$

$6 \times 3 = 18$

$6 \times 4 = 24$

$6 \times 5 = 30$

$6 \times 6 = 36$

$6 \times 7 = 42$

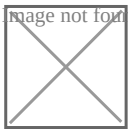
$6 \times 8 = 48$

$6 \times 9 = 54$

$6 \times 10 = 60$

9 topshiriq. M>1 natural son berilgan bo'sin.

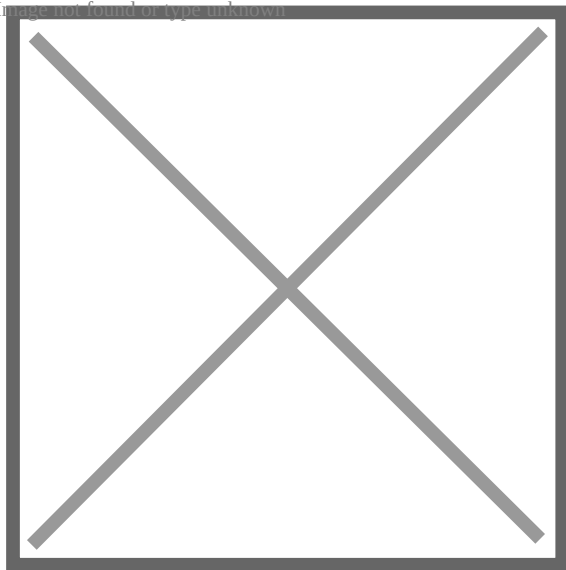
Image not found or type unknown



Shartlarni o'qitiradigan eng katta k sonini toping.

Python dasturi:

Image not found or type unknown



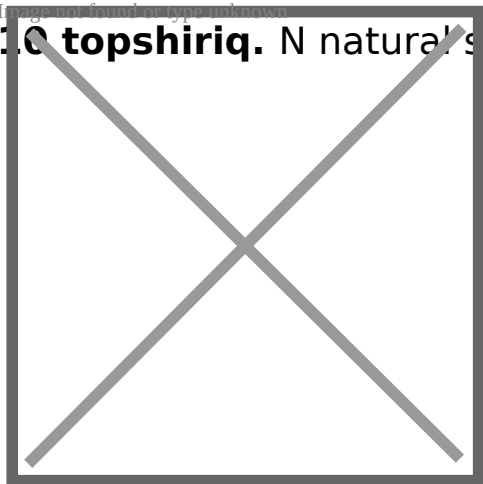
$M > 1$ natural sonini kiriting

$M = 80$

$k = 3$

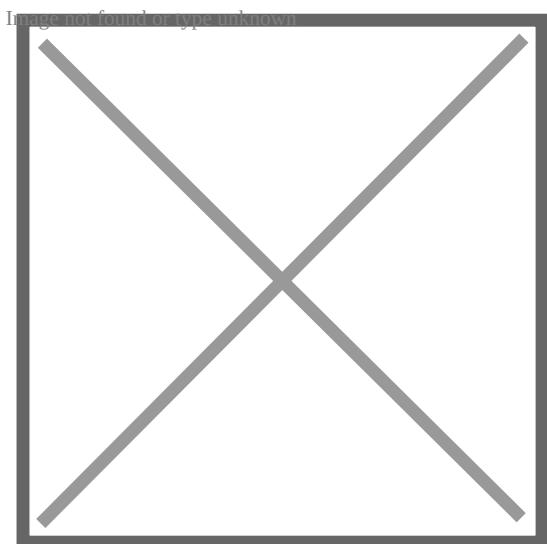
Image not found or type unknown

10 topshiriq. N natural soni berilgan.



hisoblovchi dastur tuzing.

Python dasturi:



n natural sonini kiriting

$n=30$

Yig'indini qiymati $S= 9920$

Mustaqil yechish uchun topshiriqlar

Masala 1. N -butun son. k kattalikga shu sonni tashkil etuvchi raqamlar yig'indisini o'zlashtiring.

Masala 2. N - natural son. Uning tub buluvchilari sonini aniqlang

Masala 3. N - natural son berilgan. Uning raqamlari yig'indisini toping

Masala 4. N ta haqiqiy son berilgan. Ularni orta borish tartibida joylashtiring

Masala 5. Barcha uch xonali natural sonlardan raqamlari yig'indisi berilgan n ($1 \leq n \leq 27$) ga tenglarini chop eting

Masala 6. N - natural soni berilgan. Uning kvadrati tarkibiga 3 raqami kirishi yoki kirmasligini aniqlang

Masala 7. Ketma-ketlik quyidagi tartibda yaratilgan. 1, 3, 5, 7, 9, 10, 12, 14, 16, ... ya'ni natural sonni tashkil etuvchi raqamlar yig'indisi toq bulsa, ketma-ketlik tarkibiga kiritilgan. SHu ketma-ketlikdagi 10, 15, 2008 o'rindagi raqamlarni aniqlang.

Masala 8. Berilgan n ($n > 10$) sonidagi, birinchi va oxirgi raqamlari o'rnini almashtirish va natijani chop etish dasturini tuzing

Masala 9. Berilgan n ($n > 10$) natural sonini raqamlarini teskarisidan yozib, yangi son hosil qiling.

Masala 10. k va n ($n > 0$) butun sonlar berilgan. n marta k soni chiqarilsin

Masala 11. a va b butun sonlar berilgan ($a > b$). a va b sonlari orasidagi sonlarni o'sish tartibida chiqarilsin (a va b sonlari ham kiradi) hamda shu sonlar miqdori (soni) n chiqarilsin.

Masala 12. a va b butun sonlar berilgan ($a < b$). a va b sonlari orasidagi sonlarni kamayish tartibida chiqarilsin (a va b sonlari ham kiradi) hamda shu sonlar miqdori (soni) n chiqarilsin

Masala 13. 2 ta a va b butun sonlar berilgan. ($a < b$) a dan b gacha bo'lgan butun sonlar yig'indisi topilsin.

Masala 14. 2 ta a va b butun sonlar berilgan. ($a < b$) a dan b gacha bo'lgan sonlarning ko'paytmasi topilsin.

Masala 15. n ($n > 0$) butun soni berilgan (Yig'indi haqiqiy son). Yig'indi hisoblansin

Masala 16. a haqiqiy va n butun sonlari berilgan ($n > 0$). (a , n marta ko'paytirilgan) a ning n - darajasi hisoblansin.

Masala 17. a va n sonlari berilgan. . Bitta sikldan foydalanib yig'indi hisoblansin.

Masala 18. n ($n > 0$) butun son berilgan. (n -faktorial) ko'paytma hisoblansin.

Masala 19. n butun soni n butun soni berilgan ($n > 0$). . Bitta sikldan foydalanib yig'indi hisoblansin. ($n > 0$). $1! + 2! + \dots + n!$. Bitta sikldan foydalanib yig'indi hisoblansin.

Masala 20. n butun soni berilgan ($n > 0$). . Bitta sikldan foydalanib yig'indi hisoblansin.

Masala 21. N va K butun musbat sonlari berilgan. Faqat ayirish va qo'shish amallarini ishlatib N sonini K soniga bo'lgandagi qoldiq va butun qismini aniqlovchi dastur tuzing.

Masala 22. n butun soni berilgan ($n > 0$). Agar n soni 3 ning darajasi bo'lsa $*3$ - ning darajasi'. aks xolda $*3$ - ning darajasi emas" degan natija chiqaruvchi dastur tuzing. Qoldikli bo'lish va bo'lish amallarini ishlatmang.

Masala 23. n natural soni berilgan ($n > 0$). Kvadrati n dan katta bo'ladigan eng kichik butun k sonini ($k^2 > n$) aniqlovchi dastur tuzing. Ildizdan chiqaruvchi funksiyadan foydalanmang.

Masala 24. n natural soni berilgan ($n > 1$). $3^k \leq n$ shartni qanoatlantiruvchi eng katta butun k sonini aniqlovchi dastur tuzing.

Masala 25. n va m butun musbat sonlari berilgan ($n > m$). n sonini m soniga bo'lib butun hamda qoldiq qismlarini bo'lish va qoldiqni olish amallarini ishlatmasdan topuvchi dastur tuzing

Masala 26. n butun soni berilgan ($n > 0$). Uni bo'lib butun va qoldiq qismlarini aniqlash orqali, berilgan son raqamlarini teskari tartibda chiqaruvchi dastur tuzing.

Masala 27. n butun soni berilgan ($n > 1$). n sonining tub yoki tub emasligini aniqlovchi dastur tuzing.

Masala 28. a va b butun musbat sonlari berilgan. Berilgan sonlarning eng katta umumiy bo'luvchisini aniqlovchi dastur tuzing.

Masala 29. n butun soni berilgan ($n > 0$). Uni bo'lib butun va qoldiq qismlarini aniqlash orqali, berilgan son raqamlarining orasida 2 raqami bor yoki yo'qligini aniqlovchi dastur tuzing.