

4-MA'RUZA. TAKRORLANUVCHI ALGORITMLAR

Takrorlanuvchi algoritmlar. Agar biror masalani yechish uchun tuzilgan zarur bo'lgan amallar ketma-ketligining ma'lum bir qismi biror parametrga bog'liq ko'p marta qayta bajarilsa, bunday algoritmlar takrorlanuvchi algoritmlar yoki siklik algoritmlar deyiladi. Takrorlanuvchi algoritmlarga tipik misol sifatida odatda qatorlarning yig'indisi yoki ko'patmasini hisoblash jarayonlarini qarash mumkin. Quyidagi yig'indini hisoblash algoritmini tuzaylik.

$$S = 1 + 2 + 3 + + N = \sum_{i=1}^N i$$

Bu yig'indini hisoblash uchun $i=0$ da $S=0$ deb olamiz va $i=i+1$ da $S=S+i^2$ ni hisoblaymiz. Bu yerda birinchi va ikkinchi qadamlar uchun yig'indi hisoblandi va keyingi qadamda i parametr yana bittaga orttiriladi va navbatdagi raqam avvalgi yig'indi S ning ustiga qo'shiladi va bu jarayon shu tartibda to $i \leq N$ sharti bajarilmaguncha davom ettiriladi va natijada izlangan yig'indiga ega bo'lamiz. Bu fikrlarni quyidagi algoritmlar sifatida ifodalash mumkin:

1-qadam: N –berilgan bo'lsin,

2-qadam: $i=0$ berilsin,

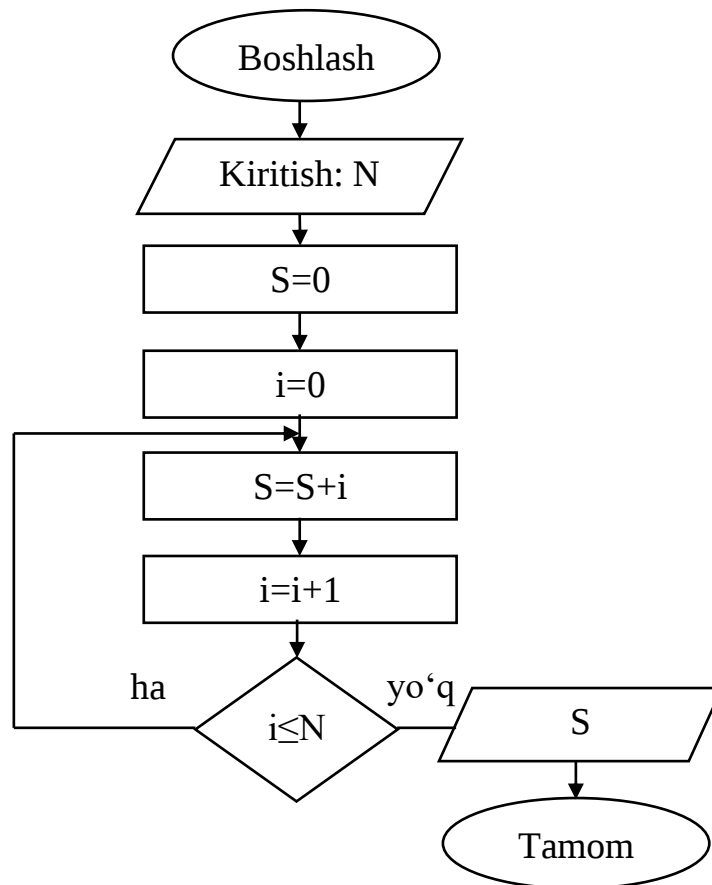
3-qadam: $S=0$ berilsin,

4-qadam: $i=i+1$ hisoblansin,

5-qadam: $S=S+i$ hisoblansin,

6-qadam: $i \leq N$ tekshirilsin va bu shart bajarilsa, 4-satrga qaytilsin, aks holda keyingi qatarga o'tilsin,

7-qadam: S ning qiymati chop etilsin.



4.1-rasm. 1 dan n gacha bo'lgan sonlar yig'indisini hisoblash algoritmi.

Yuqorida keltirilgan algoritmi va blok sxemadan ko'rinib turibdiki, amallar ketma-ketligining ma'lum qismi parametr i ga nisbatan N marta takrorlanayapti.

Endi quyidagi ko'paytmaning algoritmini va blok sxemasini tuzib ko'raylik (1 dan N bo'lgan sonlarning ko'paytmasini odatda $P!$ kabi belgilanadi va faktorial deb ataladi)

$$P = 1 \cdot 2 \cdot 3 \dots N = P!$$

$P!$ - faktorialni quyidagi ko'rinishda ham yozish mumkin: $P = \prod_{i=1}^N i$

Ko'paytmani hosil qilish algoritmi ham yig'indini hosil qilish algoritmiga juda o'xshash, faqat ko'paytmani hosil qilish uchun $i=1$ da $P=1$ deb olamiz va keyin $i=i+1$ da $P=P \cdot i$ ni hisoblaymiz. Keyingi qadamda i parametr yana bittaga orttiriladi va navbatdagi raqam avvalgi hosil bo'lgan ko'paytma P ga ko'paytiriladi va bu jarayon shu tartibda to $i < N$ sharti bajarilmaguncha davom ettiriladi va natijada

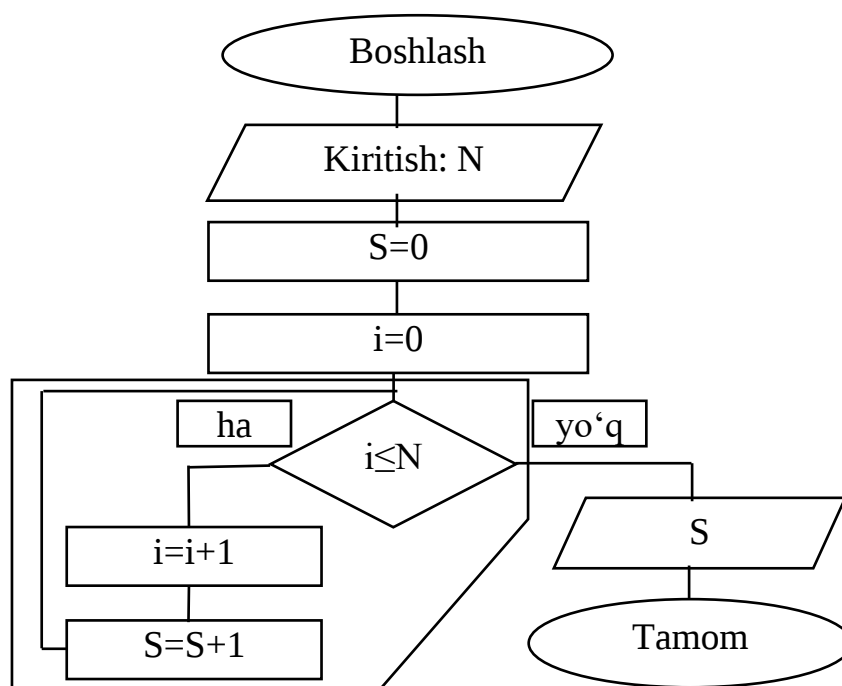
izlangan ko‘paytmaga ega bo‘lamiz. Quyidagi algoritmda bu fikrlar o‘z aksini topgan.

Matn shakli	Blok-sxemasi
1-qadam: N–berilgan bo‘lsin, 2-qadam: $i=1$ berilsin, 3-qadam: $P=1$ berilsin, 4-qadam: $i=i+1$ hisoblansin, 5-qadam: $P=P*i$ hisoblansin, 6-qadam: $i < N$ tekshirilsin va bu shart bajarilsa, 4-satrga qaytilsin, aks holda keyingi qatorga o‘tilsin, 7-qadam: P ning qiymati chop etilsin.	<pre> graph TD Start([Boshlash]) --> Input[/Kiritish: N/] Input --> P1[P=1] P1 --> i1[i=1] i1 --> LoopStart(()) subgraph Loop [] direction TB Pmult[P=P*i] --> iinc[i=i+1] end iinc --> Decision{i <= N} Decision -- ha --> LoopStart Decision -- yo‘q --> Output[/P/] Output --> End([Tamom]) style LoopStart fill:none,stroke-dasharray: 5 5 style Loop fill:none,stroke-dasharray: 5 5 style End fill:none,stroke-dasharray: 5 5 </pre>

4.2-rasm. 1 dan n gacha bo‘lgan sonlar ko‘paytmasini hisoblash algoritmi.

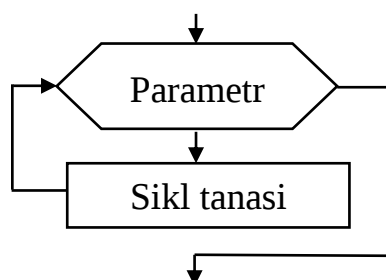
Yuqorida ko‘rilgan yig‘indi va ko‘paytmalarning blok sxemalaridagi takrorlanuvchi qismlariga (aylana ichiga olingan) quyidagi sharti keyin berilgan siklik struktura mos kelishini ko‘rish mumkin.

Yuqoridagi blok sxemalarda shartni oldin tekshiriladigan holatda chizish mumkin edi. Masalan, yig‘indining algoritmini qaraylik. Bu blok sxemaning takrorlanuvchi qismiga quyidagi, sharti oldin berilgan siklik strukturaning mos kelishini ko‘rish mumkin.



4.3-rasm. 1 dan n gacha bo‘lgan sonlar yig‘indisini hisoblash algoritmi.

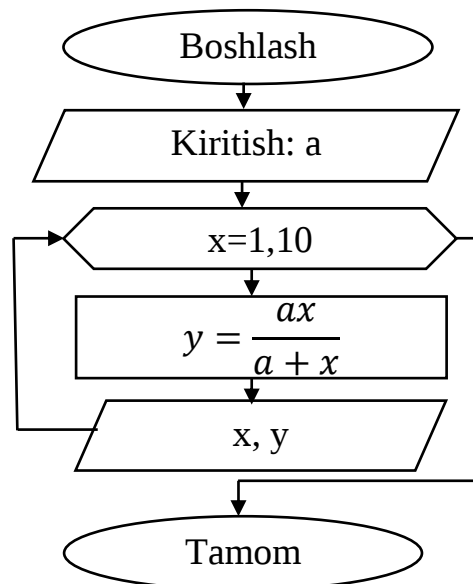
Blok sxemalarining takrorlanuvchi qismlarini, quyidagi parametrlilik takrorlash strukturasi ko‘rinishida ham ifodalash mumkin.



4.4-rasm. Parametrlilik takrorlash operatorining umumiy ko‘rinishi

Parametrlilik takrorlash operatoriga misol sifatida berilgan $x=1,2,3,\dots,10$ larda

$y = \frac{ax}{a+x}$ funksiyasining qiymatlarini hisoblash blok sxemasini qarash mumkin.



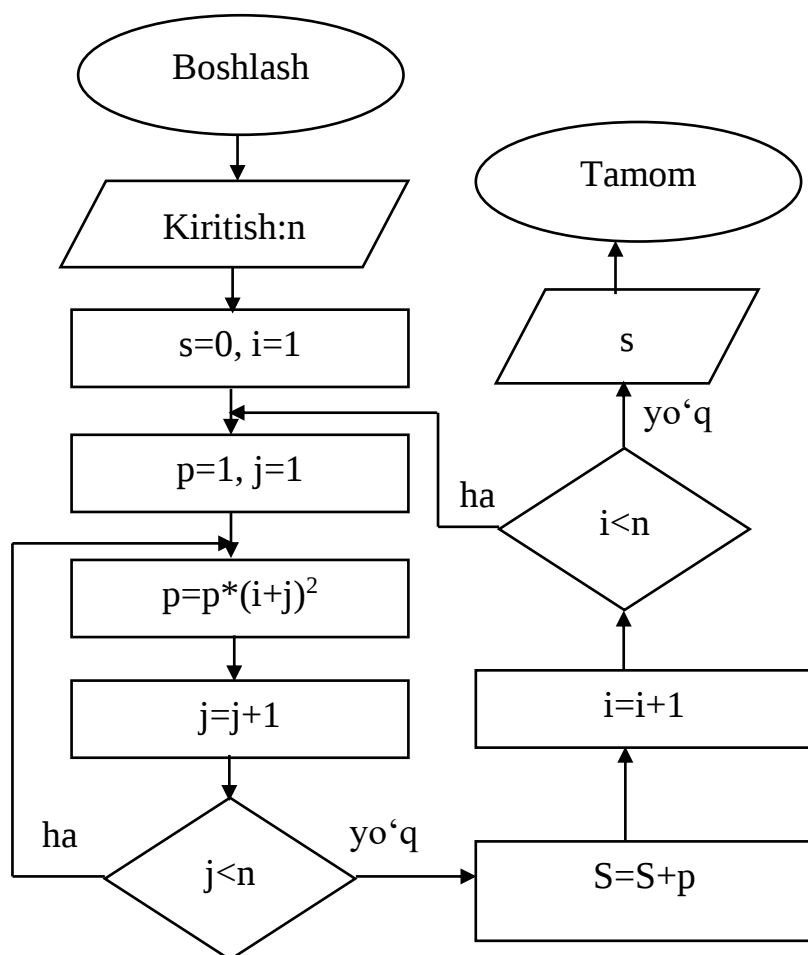
4.5-rasm. Parametrli takrorlash operatoriga doir algoritm.

Ichma-ich joylashgan siklik algoritmlar. Ba'zan, takrorlanuvchi algoritmlar bir nechta parametrlarga bog'liq bo'ladi. Odatda bunday algoritmlarni ichma-ich joylashgan algortmlar deb ataladi.

Misol sifati berilgan $n \times m$ o'lchovli a_{ij} –matritsa elementlarining yig'indisini hisoblash masalasini qaraylik.

4.1-misol. $S = \sum_{i=1}^n \prod_{j=1}^n (i + j)^2$ Bu yig'indi hisoblash uchun, i ning har bir

qiymatida j bo'yicha ko'paytmani hisoblab, avval yig'indi ustiga ketma-ket qo'shib borish kerak bo'ladi. Bu jarayon quyidagi blok–sxemada aks ettirilgan. Bu yerda i -tashqi sikl - yig'indi uchun, j -esa ichki sikl-ko'paytmani hosil qilish uchun foydalanilgan.

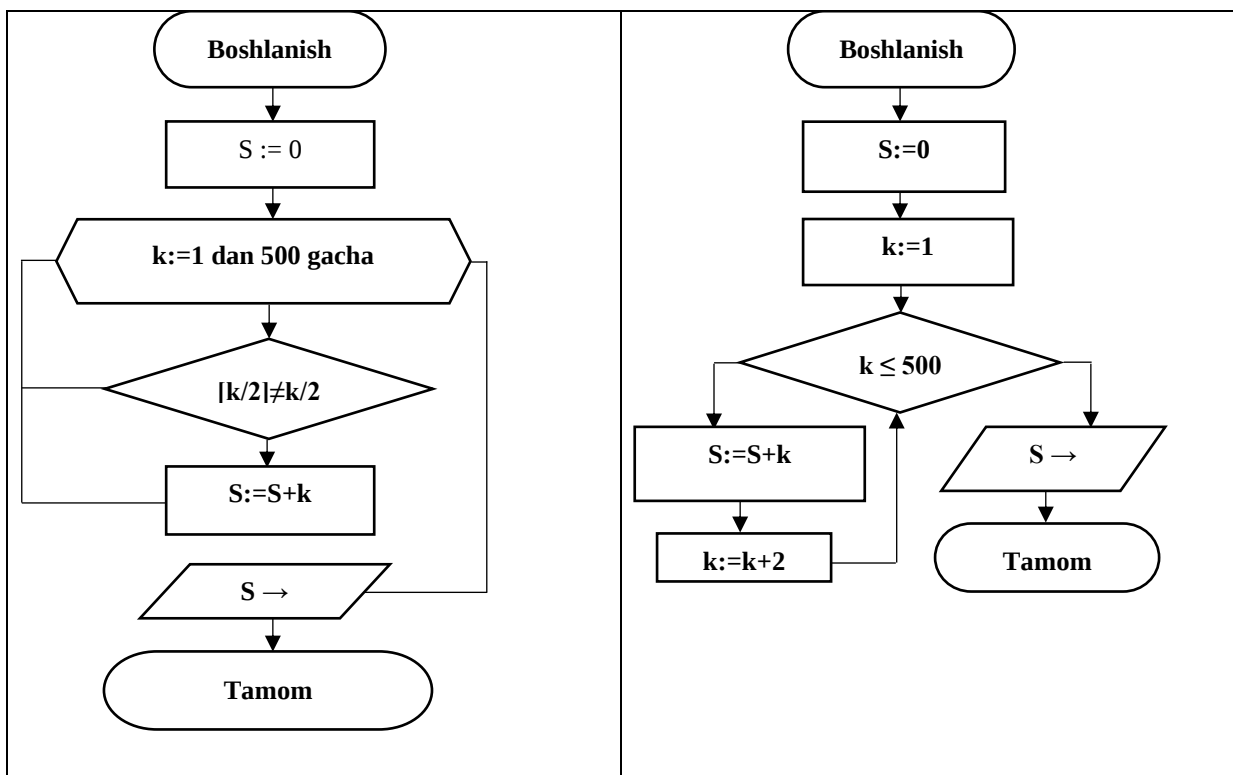


4.2-misol. 1 dan 100 gacha bo'lgan sonlar ichida 11 ga karrali sonlarning P ko'paytmasini topish algoritmini so'zlar yordamida tuzing.

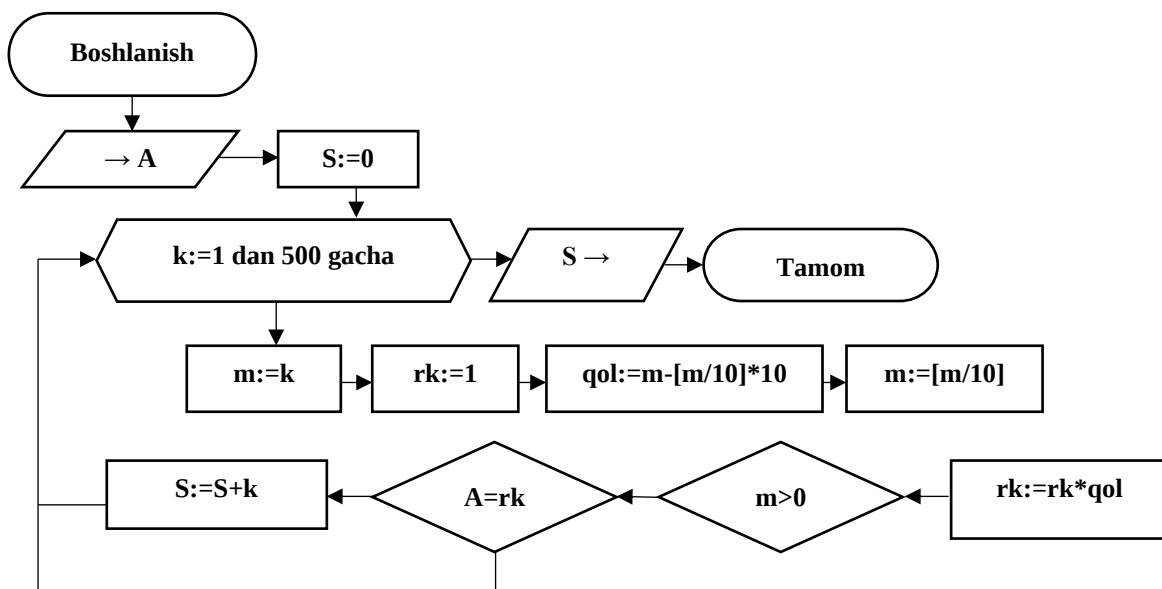
1-usul:	2-usul:	3-usul:
boshlanshi	boshlansin	boshlansin
P:=1	P:=1	P:=1
K:=1	K:=1 dan 100 gacha	K:=11
toki $K \leq 100$	agar $[K/11]=K/11$	toki $K \leq 100$
agar $[K/11]=K/11$	u holda	P:=P*K
u holda	P:=P*K	K:=K+11
P:=P*K	oxiri	oxiri
oxiri	oxiri	P chiqarilsin

K:=K+1	P chiqarilsin	tamomlansin
Oxiri	tamomlansin	
P chiqarilsin		
tamomlansin		

4.3-misol. 1 dan 500 gacha oraliqda bo'lgan toq sonlarning S yig'indisini topish algoritmini blok-sxema shaklida yozing.



4.4-misol. A natural son berilgan. 1 dan 1000 gacha bo'lgan sonlar ichida raqamlari ko'paytmasi A songa teng bo'lgan sonlar S yig'indisini topish algoritmini blok-sxema shaklida yozing.



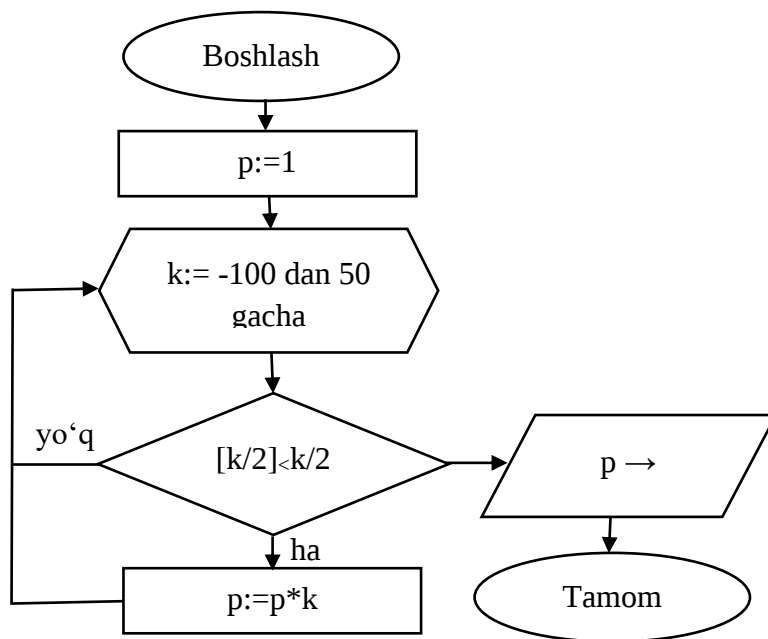
4.5-misol. -100 dan 50 gacha bo'lgan sonlar oralig'idagi toq sonlar ko'paytmasini hisoblash algoritmini tuzing.

Masalani ikki xil usulda hal etamiz:

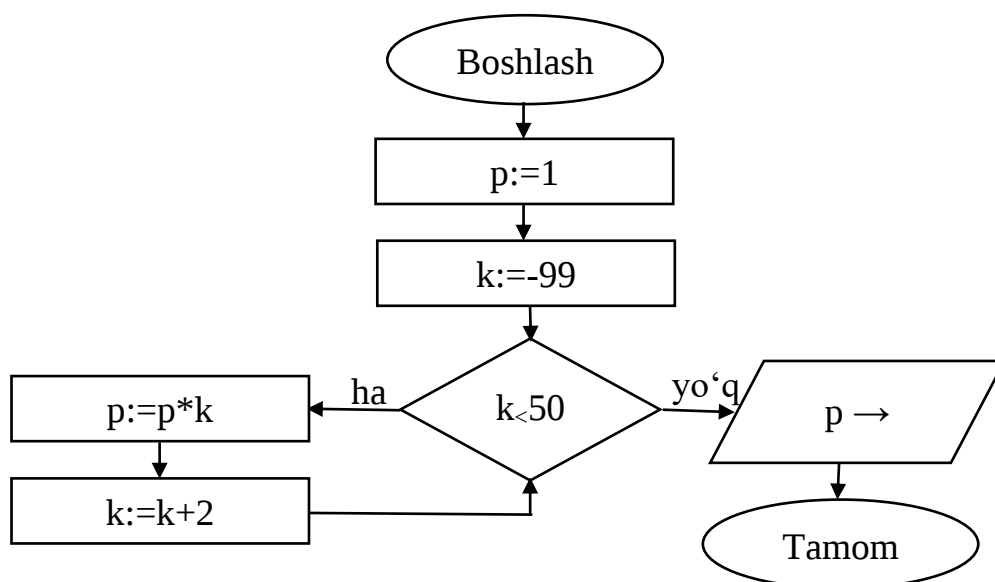
1-usul. Masalada takrorlanishlar soni avvaldan ma'lum , shuning uchun parametrli takrorlash tuzilmasidan foydalanish mumkin.

2-usul. Shart bo'yicha takrorlash tuzilmasidan foydalanish mumkin. Ko'paytma hisoblanayotgani uchun ikkala usulda ham p ning boshlang'ich qiymati 1 deb olinadi.

1-usulning blok-sxemasi:



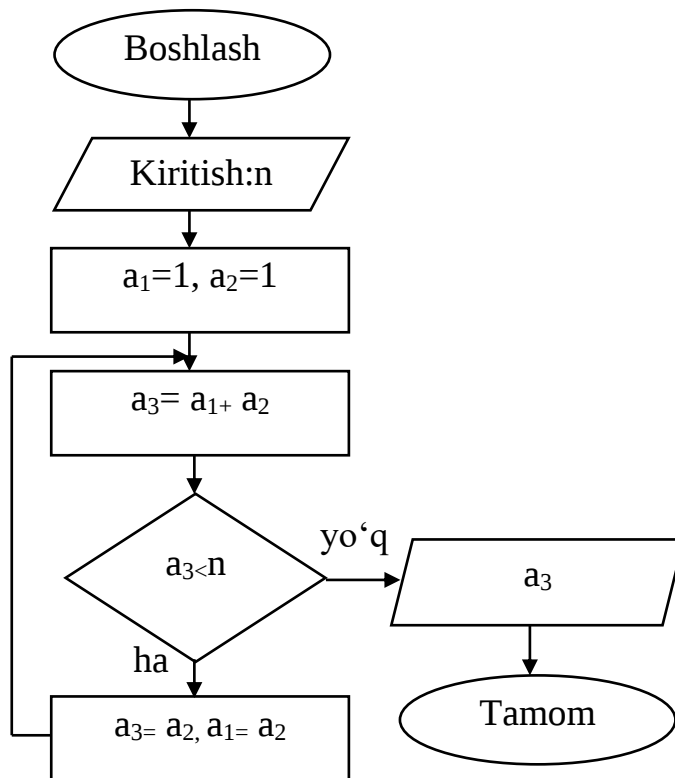
2-usul blok-sxemasi:



Rekurrent algoritmlar. Hisoblash jarayonida ba'zi bir algoritmlarning o'ziga qayta murojaat qilishga to'g'ri keladi. O'ziga-o'zi murojaat qiladigan algoritmlarga rekurrent algoritmlar yoki rekursiya deb ataladi.

Bunday algoritmgaga misol sifatida Fibonacci sonlarini keltirish mumkin. Ma'lumki, Fibonacci sonlari quyidagicha aniqlangan.

4.6-misol. $a_0=a_1=1$, $a_i=a_{i-1}+a_{i-2}$ $i=2,3,4,\dots$ Bu rekurrent ifoda 4.1-misolning algoritmgaga mos keluvchi blok-sxemada keltirilgan. Eslatib o'tamiz formuladagi i -indeksga hojat yo'q, agar Fibonacci sonining nomerini ham aniqlash zarur bo'lsa, birorta parametr-kalit kiritish kerak bo'ladi.



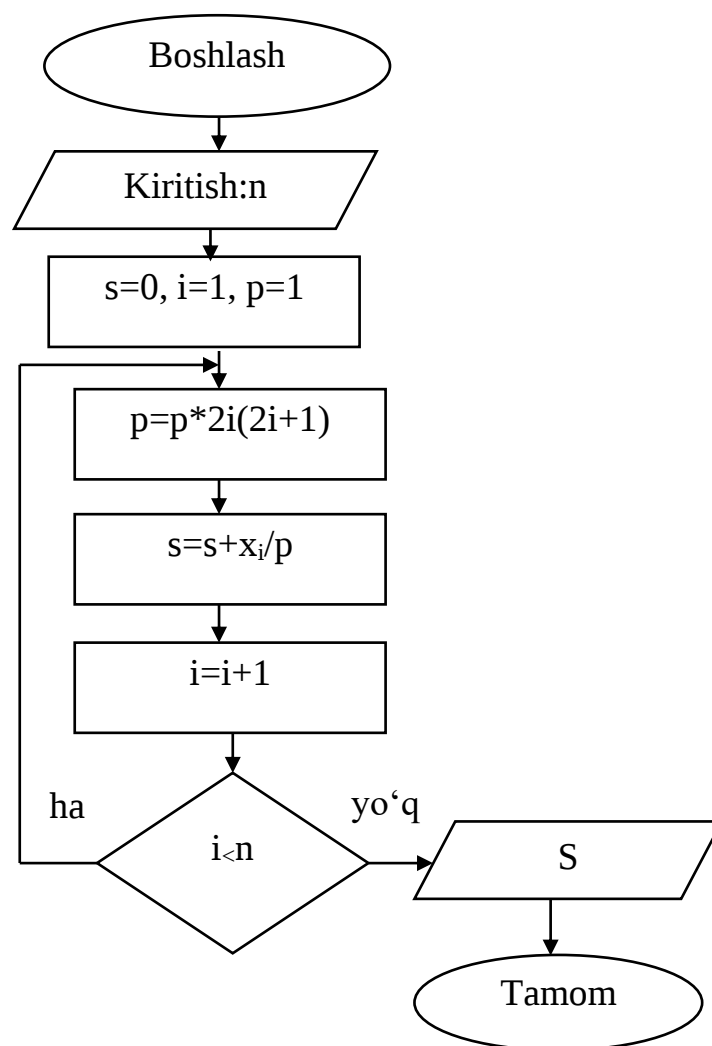
4.7-misol.
$$S = \sum_{i=1}^n \frac{x_i}{(2i+i)!}$$

Bu ifoda i ning har bir qiymatida faktorialni va yig'indini hisoblashni taqozo etadi. Shuning uchun avval faktorialni hisoblashni alohida ko'rib chiqamiz. Quyidagi rekkurent ifoda faktorialni kam amal sarflab qulay usulda hisoblash imkonini beradi:

$$R=1$$

$$R=R*2i*(2i+1)$$

Haqiqatan ham, $i=1$ da $3!$ ni, $i=2$ da $R=3!*4*5=5!$ ni va hokazo tarzda $(2i+1)!$ ni yuqoridagi rekkurent formula yordamida hisoblash mumkin bo'ladi. Bu misolga mos keluvchi blok-sxema quyida keltirilgan.

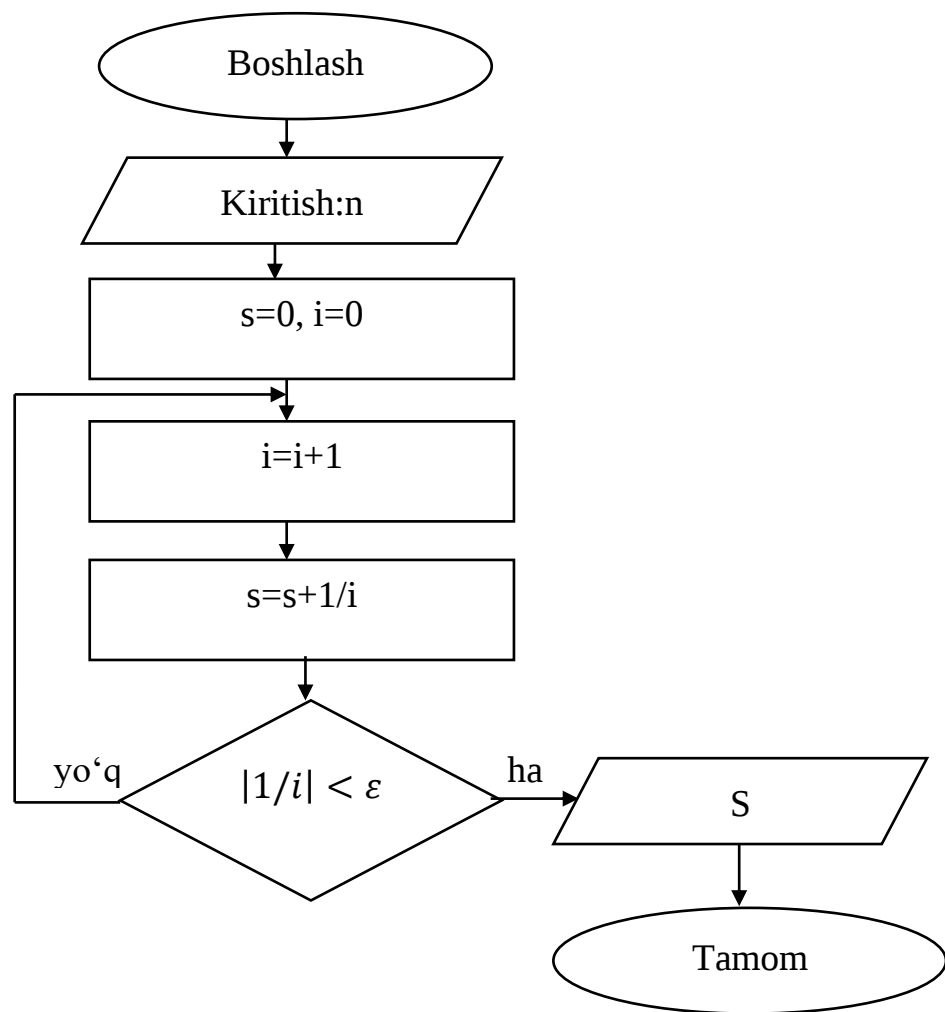


Takrorlanishlar soni noma'lum bo'lgan algoritmlar. Amalda shunday bir masalalar uchraydiki, ularda takrorlanishlar soni oldindan berilmagan-noma'lum bo'ladi. Ammo, bu jarayonni tugatish uchun biror bir shart berilgan bo'ladi.

Masalan, quyidagi $S = 1 + \frac{1}{2} + \frac{1}{3} + \dots = \sum_{i=1}^{\infty} \frac{1}{i}$ qatorda nechta had bilan chegaralanish

berilmagan. Lekin qatorni ε aniqlikda hisoblash zarur bo'ladi. Buning uchun

$\left| \frac{1}{i} \right| < \varepsilon$ shartni olish mumkin.



4.6-rasm. Takrorlanishlar soni oldindan noma'lum bo'lgan algoritmlarga doir blok-sxema.

Ketma-ket yaqinlashuvchi yoki iteratsion algoritmlar. Yuqori tartibli algebrayik va transsendent tenglamalarni yechish ususllari yoki algoritmlari ketma-ket yaqinlashuvchi – iteratsion algoritmlarga misollar bo'la oladi. Ma'lumki, transsendent tenglamalarni yechishning quyidagi asosiy usullari mavjud:

- Urinmalar usuli (Nyuton usuli),
- Ketma-ket yaqinlashishi usuli,
- Vatarlar usuli,
- Teng ikkiga bo'lish usuli.

Bizga

$$f(x)=0 \quad (1)$$

transsendent tenglama berilgan bo'lsin. Faraz qilaylik bu tenglama $[a,b]$ oraliqda uzluksiz va $f(a)*f(b)<0$ shartni qanoatlantirsin. Ma'lumki, bu holda berilgan

tenglama $[a,b]$ orilaqda kamida bitta ildizga ega bo‘ladi va u quyidagi formula orqali topiladi.

$$X_{n+1} = X_n - \frac{f(X_n)}{f'(X_n)} \quad n = 0,1,2,\dots \quad (2)$$

Boshlang‘ich X_0 qiymat $f(x_0)f''(x_0) < 0$ shart asosida tanlab olinsa, (2) iteratsion albatta yaqinlashadi. Ketma-ketlik

$$|X_{n+1} - X_n| < \varepsilon$$

shart bajarilgunga davom ettiriladi.

4.8-misol. Berilgan musbat a xaqiqiy sondan kvadrat ildiz chiqarish algoritmi tuzilsin.

Bu masalani yechish uchun kvadrat ildizni x deb belgilab olib,

$$\sqrt{a} = x \quad (3)$$

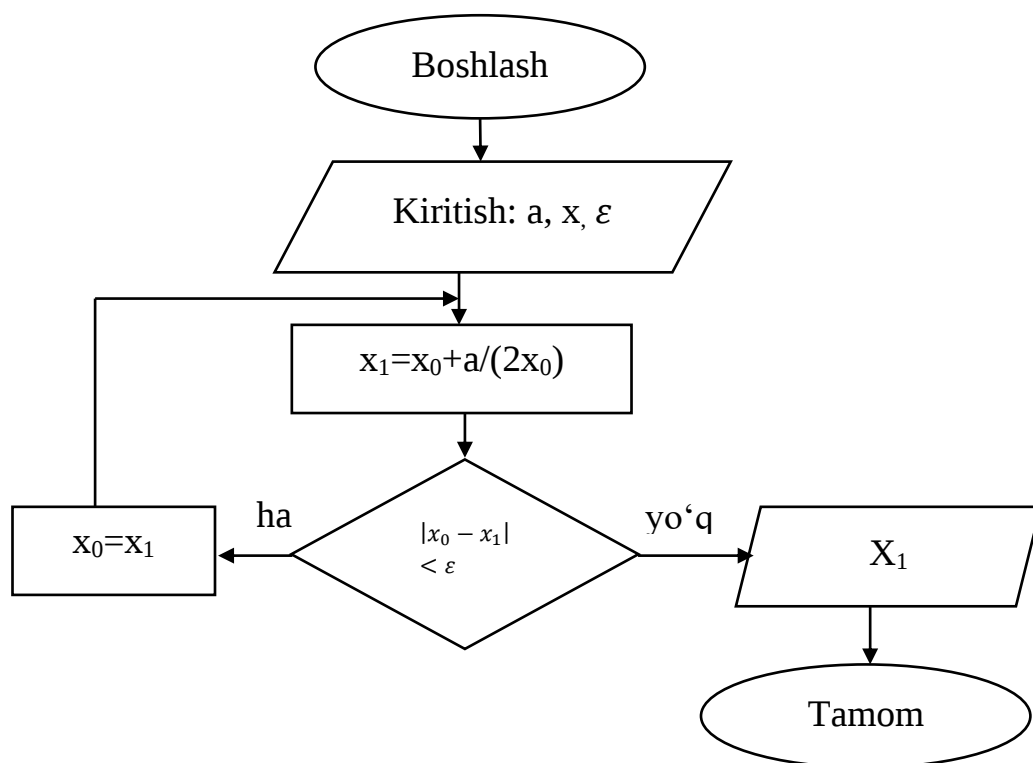
ifodalash yozib olamiz. U holda (1) tenglamaga asosan

$$f(x) = x^2 - a \quad (4)$$

ekanligini topish mumkin (4) ifodani (2) ga qo‘yib, quyidagi rekurrent formulani topish mumkin:

$$X_{n+1} = \frac{1}{2} \left(X_n + \frac{a}{X_n} \right) \quad (5)$$

Bu formulaga mos blok-sxema 4.6-rasmda keltirilgan. ε - kvadrat ildizni topishning berilgan aniqligi. Eslatib o‘tamiz, algoritmda indeksli o‘zgaruvchilarga zarurat yo‘q.



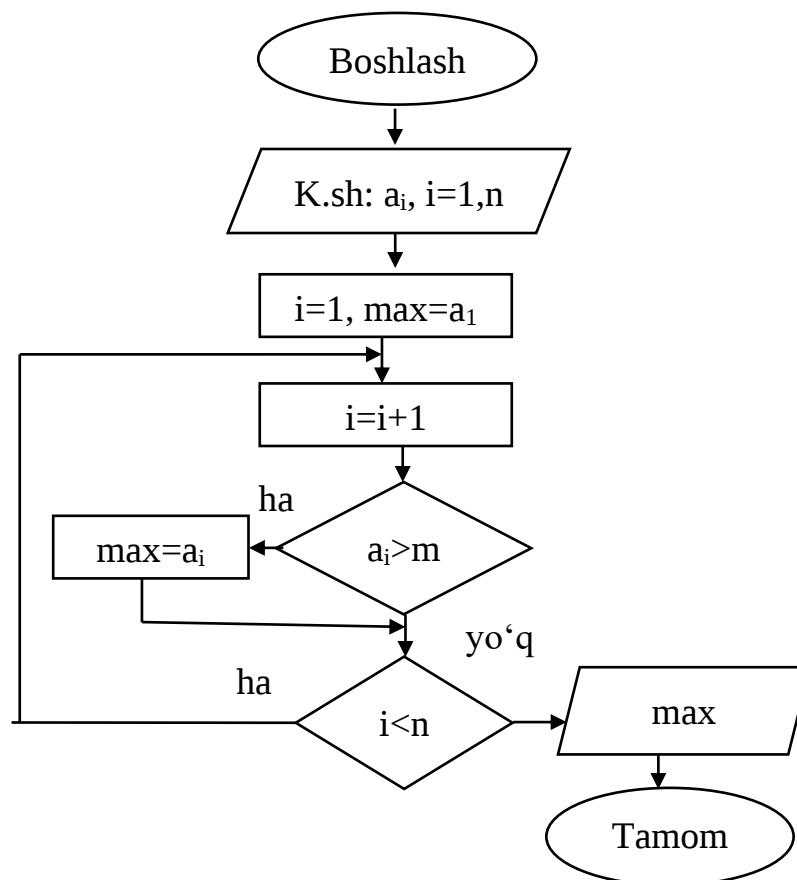
4.7-rasm. Berilgan musbat a haqiqiy sondan kvadrat ildiz chiqarish algoritmi (iteratsion algoritmgaga doir blok-sxema).

Algoritm ijrosini tekshirish. Kompyuter uchun tuzilgan algoritm ijrochisi-bu kompyuterdir. Biror programmalash tilida yozilgan algoritm kodlashtirilgan oddiy koʻrsatmalar ketma-ketliliga oʻtadi va mashina tomonidan avtomatik ravishda bajariladi. Metodik nuqtayi-nazardan qaraganda algoritmning birinchi ijrochisi sifatida oʻquvchining oʻzini olish muhim ahamiyatga ega. Oʻquvchi tomonidan biror masalani yechish algoritmi tuzilganda bu algoritmni toʻgʻri natija berishini tekshirish juda muhimdir. Buning yagona usuli oʻquvchi tomonidan algoritmni turli boshlangʻich maʼlumotlarda qadam - qadam bajarib (ijro etib) koʻrishdir. Algoritmni bajarish natijasida xatolar aniqlanadi va toʻgʻrilanadi. Ikkinchi tomonidan, masalani yechishga qiynalayotgan oʻquvchi uchun tayyor algoritmni bajarish – masalani yechish yoʻllarini tushunishga xizmat qiladi.

Algoritm ijrosini quyidagi misolda koʻraylik.

Berilgan a_i , $i=1, n$ sonlarning eng kattasini topish algoritmini tuzaylik. Buning uchun, berilgan sonlardan birinchisi a_1 ni $i=1$ eng katta qiymat deb faraz qilaylik va uni max nomli yangi oʻzgaruvchiga uzataylik: $\max=a_1$. Parametr i ning

qiymatini bittaga oshirib, ya'ni $i=i+1$ a_1 ni a_2 bilan taqqoslaymiz va qaysi biri katta bo'lsa uni max o'zgaruvchisiga uzatamiz va jarayonni shu tarzda to $i=n$ bo'lguncha davom ettiramiz. Bu fikrlar quyidagi blok-sxemada o'z aksini topgan.



4.8-rasm. Vektor elementlarining eng kattasini topish algoritmi.

Endi bu blok-sxema yoki algoritmnining ijrosini $n=3$ $a_1=3$, $a_2=5$, $a_3=1$ aniq sonlarda ko'rib o'taylik:

1-qadam. $i=1$ da $max=3$ bo'ladi.

2-qadam. $i=i+1=2$ ni topamiz,

3-qadam. $a_2 > max$, ya'ni $5 > 3$ ni tekshiramiz, shart bajarilsa, $max=5$ bo'ladi.

4-qadam. $i < n$, ya'ni $2 < 3$ ni tekshiramiz. Shart bajarilsa, i ni yana bittaga oshiramiz, va $i=3$ bo'ladi, va

5-qadam. $a_3 > max$, ya'ni $1 > 5$, ni tekshiramiz. Shart bajarilmadi, demak, keyingi

6-qadam. $i < n$ shartni, ya'ni $3 < 3$ ni tekshiramiz. Shart bajarilmadi. Demak $max=5$ chop etiladi. Biz blok-sxemani tahlil qilish davomida uning to'g'riligiga

ishonch hosil qildik. Endi ixtiyoriy n lar uchun bu blok-sxema bo'yicha eng katta elementni topish mumkin.

Savol va topshiriqlar

1. Yig'indining $S = \sum_{i=1}^n a_i$ algoritmi va blok sxemasi tuzilsin.
2. Ko'paytmaning algoritmi va blok sxemasi tuzilsin $P = \prod_{i=1}^n a_i$
3. Berilgan sonlarning eng kattasini topadigan algoritm va blok-sxema tuzilsin.
4. 6. Ikkita n va m natural sonning eng katta umumiy bo'luvchisini topish algoritmi (Evklid algoritmi) tuzilsin.
5. 4 xonali sonlar orasidan avvalgi ikkita raqamli yig'indisi, keyingi 2 raqamli yig'indisiga teng bo'lgan sonlar chop etilsin va miqdori aniqlansin.
6. Berilgan a_{ij} $n \times n$ o'lchovli matritsaning satr elementlarining yig'indisi chop etilsin.
7. $n \times m$ o'lchovli a_{ij} matritsaning elementlarining eng katta va kichik elementlari topilsin.
8. Berilgan $a_1, a_2, a_3, \dots, a_n$ sonlarning eng katta va kichik elementlarini bir vaqtda topadigan blok-sxema tuzing va uni $n=3$ da tekshiring.
9. Berilgan $a_1, a_2, a_3, \dots, a_n$ sonlarni qiymatlari bo'yicha o'sish tartibida yozing.
10. Ikkita n va m natural sonlarining eng katta umumiy bo'luvchisini topish (Yevklid) algoritmiga blok-sxema tuzilsin.