## 1-mavzu. Algoritm tushunchasi va ulardan foydalanish.

**Tayanch so’zlar:** Algoritm tushunchasi va ular haqida ma’lumotlar. Algoritmlar tarixi. Algoritm xossalari, turlari va uning berilish usullari.

**Reja:**

1. Amaliy masalalarni kompyuterda yechish jarayoni

2. “Algoritm” tushunchasi

3. Algoritmning asosiy xossalari

4. Algoritmni tasvirlash usullari

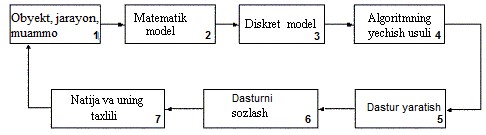
5. Chiziqli algoritmlar

6. Tarmoqlanuvchi algoritmlar

7. Takrorlanuvchi algoritmlar

**1. Amaliy masalalarni kompyuterda yechish jarayoni**

Kundalik faoliyatimizda biz turli xil muammolar, vazifalarga duch kelamiz. Biz uchun ular talab yoki savol shaklida bo‘ladi. Masalan, «Kvadrat tenglamani yechish», «Dengizda nechta tomchi suv bor?», «Ikki karra ikki necha bo‘ladi?» kabi. Masalani yechish uchun kerakli amallarni bajarish, ma’lum bir ketmaketlikda qator harakatlarni amalga oshirish darkor. Aynan shu harakatlar ketmaketligini to‘liq tasavvur etishimiz va tasvirlab berishimiz kerak. Turli muammo, masala yoki jarayonlarni o‘rganishni kompyuter yordamida amalga oshirish uchun, birinchi navbatda, tadqiq qilinayotgan masala, jarayon - obyektning matematik ifodasi, ya’ni matematik modelini qurish kerak bo‘ladi. Matematik model real obyektning tasavvurimizdagi mavhum ko‘rinishi bo‘lib, u matematik belgilar va ba’zi-bir qonun-qoidalar bilan ifodalangan bo‘ladi. Qurilayotgan obyektning matematik modelini yaratish juda murakkab jarayon bo‘lib, o‘rganilayotgan obyektga bog‘liq ravishda turli soha mutaxassislarining ishtiroki talab etiladi. Umuman, biror masalani kompyuter yordamida yechishni quyidagi bosqichlarga ajratish mumkin (1.1.-rasm).



1-rasm. Hisoblash eksperimentining sxemasi

Misol sifatida, kosmik kemani Erdan Zuhro sayyorasiga eng qulay trayektoriya bo‘yicha uchirish masalasini hal qilish talab qilingan bo‘lsin.

Birinchi navbatda, qo‘yilgan masala turli soha mutaxassislari tomonidan atroflicha o‘rganilishi va bu jarayonni ifodalaydigan eng muhim bo‘lgan asosiy parametrlarni aniqlash kerak bo‘ladi (1.1.- rasm, 1-blok). Masalan, fizik-astronom muhandis tomonidan masala qo‘yilishining o‘rinli ekanligi, ya’ni sayyoralar orasidagi masofa va atmosfera qatlamlarining ta’siri, yerning tortish kuchini engib o‘tish va kemaning og‘irligi, zarur bo‘lgan yoqilg‘ining optimal miqdori va kosmik kemani qurishda qanday materiallardan foydalanish zarurligi, inson sog‘lig‘iga ta’siri va sarflanadigan vaqt va yana turli tuman ta’sirlarni hisobga olgan holda shu masalaning matematik modelini tuzish zarur bo‘ladi. Zikr etilgan ta’sirlarni va fizika qonunlarini hisobga olgan holda bu masalani ifodalaydigan muayyan differensial yoki boshqa ko‘rinishdagi modellovchi tenglama hosil qilish mumkin bo‘ladi. Balki bu masalani bir nechta alohida masalalarga ajratib o‘rganish maqsadga muvofiqdir. Bu matematik modelni o‘rganish asosida mazkur masalani ijobiy hal qilish yoki hozirgi zamon sivilizatsiyasi bu masalani yechishga qodir emas degan xulosaga ham kelish mumkin. Ushbu fikrlar, yuqorida keltirilgan sxemaning 2-blokiga mos keladi.

Faraz qilaylik, matematik model ishlab chiqildi. Endi uni kompyuter vositasida yechish masalasi paydo bo‘ladi. Bizga ma’lumki, kompyuter faqat 0 va 1 diskret qiymatlar va ular ustida arifmetik va mantiqiy amallarni bajara oladi xolos. SHuning uchun matematik modelga mos diskret modelni qurish zaruriyati tug‘iladi (3-blok). Odatda, matematik modellarga mos keluvchi diskret modellar ko‘p noma’lumli murakkab chiziqsiz algebraik tenglamalar sistemasi (chekli ayirmali tenglamalar-sxemalar) ko‘rinishida bo‘ladi (4-blok). Endi, hosil bo‘lgan diskret modelni sonli yechish usuli – algoritmini yaratish zarur bo‘ladi. Algoritm esa tuziladigan dastur uchun asos bo‘ladi (5-blok). Odatda, tuzilgan dasturni ishchi holatga keltirish uchun uning xato va kamchiliklarini tuzatish – sozlash zarur bo‘ladi (6-blok). Olingan sonli natijalar hali dasturning to‘g‘ri ishlayotganligiga kafolatini bermaydi. SHuning uchun olingan natijalarni masalaning mohiyatidan kelib chiqqan holda tahlil qilish kerak bo‘ladi (7-blok). Agar olingan natija o‘rganilayotgan jarayonni ifodalay olmasa, masalani 1.1- rasmdagi sxema asosida qaytadan ko‘rib chiqish va zarur bo‘lgan joylarda o‘zgartirishlar kiritish kerak bo‘ladi. Bu jarayon to kutilgan ijobiy yoki salbiy natija olinguncha davom ettiriladi va bu takrorlanuvchi jarayon hisoblash eksperimenti deb ataladi. Odatda, hisoblash eksperimenti deganda, soddaroq holda, model, algoritm va dastur uchligini (triadasini) tushunish mumkin (1.2-rasm).



1.2 – rasm. Hisoblash eksperimenti uchligi

**2. “Algoritm” tushunchasi**

Yuqorida qayd etganimizdek, qo‘yilgan biror masalani kompyuterda yechish uchun, avval uning matematik modelini, keyin esa yechish algoritmi va dasturini tuzish kerak bo‘ladi. Ushbu uchlikda algoritm bloki muhim ahamiyatga ega. Endi algoritm tushunchasining ta’rifi va xossalarini bayon qilamiz. Masala echimini cheklangan qadamlar natijasida hosil qiladigan, oldindan tayinlangan va aniq belgilangan qoidalar yoki buyruqlar ketma-ketligi *algoritm* deyiladi. Soddaroq qilib aytganda, algoritm bu - oldimizga qo‘yilgan masalani yechish uchun zarur bo‘lgan amallar ketma-ketligidir. Algoritm tuzish - bu dasturlashdir, algoritmni tuzuvchilar esa dasturchilardir.

Masalan, *a*x2+*b*x+*c*=0 kvadrat tenglamani yechish uchun quyidagi amallar ketma-ketligi zarur bo‘ladi:

*a, b, c*  koeffitsiyentlar berilgan bo‘lsin.

Berilgan *a, b, c*  koeffitsiyentlar yordamida discriminant.

*D*= *b2-4ac* hisoblanadi.

*D>0* bo‘lsa *X*12 = (−*b*± *D*)/ 2\*( *a*)*-* hisoblanadi.

*D<0* bo‘lsa, haqiqiy echimi yo‘q.

Misol sifatida berilgan *a, b, s* tomonlari bo‘yicha uchburchakning yuzasini Gerron formulasi bo‘yicha hisoblash masalasini ko‘rib chiqaylik.

*a, b, c* uchburchak tomonlari uzunliklari.

*r* = *(a*+*b*+*c)*/*2* - perimetrning yarmi hisoblansin.

T= p(r-a) (r-b) (r-c) – hisoblansin.

*S*= *T* hisoblansin.

Yuqoridagi misollardan ko‘rinib turibdiki, algoritmning har bir qadamida bajariladigan amallar tushunarli va aniq tarzda ifodalangan hamda chekli sondagi amallar bajarilgandan keyin aniq natija olish mumkin.

Zikr etilgan, tushinarlilik, aniqlilik, cheklilik va natijaviylik tushunchalari algoritmning asosiy xossalarini tashkil etadi. Bu tushunchalar keyingi paragraflarda alohida ko‘rib o‘tiladi.

“Algoritm” so‘zi va tushunchasi IX asrda yashab ijod etgan buyuk alloma Muhammad al-Xorazmiy nomi bilan uzviy bog‘liq. Algoritm so‘zi Al-Xorazmiy nomini Evropa olimlari tomonidan buzib talaffuz qilinishidan yuzaga kelgan. AlXorazmiy birinchi bo‘lib o‘nlik canoq sistemasining tamoyillarini va undagi to‘rtta amalni bajarish qoidalarini asoslab bergan.

**3. Algoritmning asosiy xossalari**

Algoritmning 5 ta asosiy xossasi bor.

***Diskretlilik (Cheklilik).*** Bu xossaning mazmuni algoritmlarni doimo chekli qadamlardan iborat qilib bo‘laklash imkoniyati mavjudligida. Ya’ni uni chekli sondagi oddiy ko‘rsatmalar ketma-ketligi shaklida ifodalash mumkin. Agar kuzatilayotgan jarayonni chekli qadamlardan iborat qilib qo‘llay olmasak, uni algoritm deb bo‘lmaydi.

***Tushunarlilik.*** Biz kundalik hayotimizda berilgan algoritmlar bilan ishlayotgan elektron soatlar, mashinalar, dastgohlar, kompyuterlar, turli avtomatik va mexanik qurilmalarni kuzatamiz.

Ijrochiga tavsiya etilayotgan ko‘rsatmalar uning uchun tushinarli mazmunda bo‘lishi shart, aks holda, ijrochi oddiygina amalni ham bajara olmaydi. Bundan tashqari, ijrochi har qanday amalni bajara olmasligi ham mumkin.

Har bir ijrochining bajarishi mumkin bo‘lgan ko‘rsatmalar yoki buyruqlar majmuasi mavjud, u ijrochining ko‘rsatmalar tizimi (sistemasi) deyiladi. Demak, ijrochi uchun berilayotgan har bir ko‘rsatma ijrochining ko‘rsatmalar tizimiga mansub bo‘lishi lozim.

Ko‘rsatmalarni ijrochining ko‘rsatmalar tizimiga tegishli bo‘ladigan qilib ifodalay olishimiz muhim ahamiyatga ega. Masalan, quyi sinfning a’lochi o‘quvchisi "son kvadratga oshirilsin" degan ko‘rsatmani tushunmasligi natijasida bajara olmaydi, lekin "son o‘zini o‘ziga ko‘paytirilsin" shaklidagi ko‘rsatmani bemalol bajaradi, chunki u ko‘rsatma mazmunidan ko‘paytirish amalini bajarish kerakligini anglaydi.

***Aniqlik*.** Ijrochiga berilayotgan ko‘rsatmalar aniq va mazmunli bo‘lishi zarur. Chunki ko‘rsatmadagi noaniqliklar mo‘ljaldagi maqsadga erishishga olib kelmaydi. Inson uchun tushunarli bo‘lgan "3-4 marta silkitilsin", "5-10 daqiqa qizdirilsin", "1-2 qoshiq solinsin", "tenglamalardan biri echilsin" kabi noaniq ko‘rsatmalar kompyuterni qiyin ahvolga solib qo‘yadi.

Bundan tashqari, ko‘rsatmalarning qaysi ketma-ketlikda bajarilishi ham muhim ahamiyatga ega. Demak, ko‘rsatmalar aniq berilishi va faqat algoritmda ko‘rsatilgan tartibda bajarilishi shart ekan.

***Ommaviylik*.** Har bir algoritm mazmuniga ko‘ra bir turdagi masalalarning barchasi uchun ham o‘rinli bo‘lishi kerak. Ya’ni masaladagi boshlang‘ich ma’lumotlar qanday bo‘lishidan qat’i nazar algoritm shu xildagi har qanday masalani yechishga yaroqli bo‘lishi kerak. Masalan, ikki oddiy kasrning umumiy maxrajini topish algoritmi, kasrlarni turlicha o‘zgartirib bersangiz ham ularning umumiy maxrajlarini aniqlab beraveradi. Yoki uchburchakning yuzini topish algoritmi, uchburchakning qanday bo‘lishidan qat’i nazar, uning yuzasini hisoblab beraveradi.

***Natijaviylik*.** Har bir algoritm chekli sondagi qadamlardan so‘ng, albatta, natija berishi shart. Bajariladigan amallar ko‘p bo‘lsa ham baribir natijaga olib kelishi kerak. Chekli qadamdan so‘ng qo‘yilgan masalae chimga ega emasligini aniqlash ham natija hisoblanadi. Agar ko‘rilayotgan jarayon cheksiz davom etib natija bermasa, uni algoritm deb atay olmaymiz.

**4. Algoritmni tasvirlash usullari**

Yuqorida ko‘rilgan misollarda, odatda, biz masalani yechish algoritmini so‘zlar va matematik formulalar orqali ifodaladik. Lekin algoritm boshqa ko‘rinishlarda ham berilishi mumkin. Biz endi algoritmlarning eng ko‘p uchraydigan turlari bilan tanishamiz.

*Algoritmning so‘zlar orqali ifodalanishi.*Bu usulda ijrochi uchun beriladigan har bir ko‘rsatma jumlalar, so‘zlar orqali buyruq shaklida beriladi.

*Algoritmning formulalar bilan ifodalanish*usulidan matematika, fizika, kimyo kabi aniq fanlardagi formulalarni o‘rganishda foydalaniladi. Bu usul ba’zan analitik ifodalash deyiladi.

*Algoritmlarning maxsus geometrik shakllar yordamida ifodalanishida* masala yechish jarayoni aniq va ravon tasvirlanadi va bu ko‘rinish blok-sxema deyiladi.

*Algoritmning jadval ko‘rinishda berilishi*.Algoritmning bunday ifodasidan ham ko‘p foydalanamiz. Masalan, maktabda qo‘llanib kelinayotgan to‘rt xonali matematik jadvallar yoki turli xil lotereyalar jadvali. Funksiyalarning grafiklarini chizishda ham algoritmlarning qiymatlari jadvali ko‘rinishlaridan foydalanamiz. Bu kabi jadvallardan foydalanish algoritmlari sodda bo‘lgani tufayli ularni o‘zlashtirib olish oson.

Yuqorida ko‘rilgan algoritmlarni tasvirlash usullarining asosiy maqsadi, qo‘yilgan masalani yechish uchun zarur bo‘lgan amallar ketma-ketligining eng qulay holatini aniqlash va shu bilan inson tomonidan dastur yozishni yanada osonlashtirishdan iborat. Aslida, dastur ham algoritmning boshqa bir ko‘rinishi bo‘lib, u insonning kompyuter bilan muloqotini qulayroq amalga oshirish uchun mo‘ljallangan.

Blok-sxemalarni tuzishda foydalaniladigan asosiy sodda geometrik figuralar quyidagilardan iborat:

|  |  |
| --- | --- |
| Figura shakli | Vazifasi |
|  | oval, algoritmning boshlanishi yoki tugallanishini belgilaydi |
|  | parallelogramm, ma’lumotlarni kiritish yoki chop etishni belgilaydi |
|  | to‘g‘ri to‘rtburchak, amal bajarish jarayonini belgilaydi |
|  | romb, shart bajarilishi tekshirilishini belgilaydi |
|  | yordamchi algoritmga murojaatni belgilaydi |
|  | oltiburchak, takrorlash operatorini ifodalashni belgilaydi |
|  | strelka, amallar bajarilish ketma-ketligini aniqlaydi |
| => (n) | so‘zlar orqali ifodalangan algoritmda *n* - chi amalga o‘tishni ko‘rsatadi |

Blok-sxemalar bilan ishlashni yaxshi o‘zlashtirib olish zarur, chunki bu usul algoritmlarni ifodalashning eng qulay usullaridan biri bo‘lib, dastur tuzishni osonlashtiradi, dasturlash qobiliyatini mustahkamlaydi. Algoritmik tillarda blok - sxemaning asosiy strukturalariga maxsus operatorlar mos keladi.

Shuni aytish kerakni, blok-sxemalardagi yozuvlar odatdagi yozuvlardan katta farq qilmaydi.

Masalan, misol sifatida 1.2 punktda keltirilgan *a*x2+*b*x+*c*=0 kvadrat tenglamaning haqiqiy echimlarini hisoblash uchun quyidagi amallar ketma-ketligi zarur bo‘ladi:

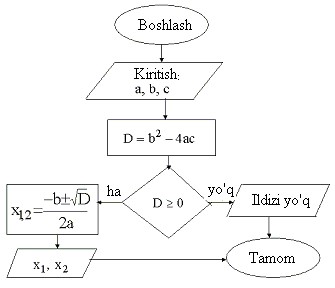
berilganlarni kiritish (*a, b, c*);

*D*=*b2-4ac* diskriminantni hisoblash;

*agar D>0* bo‘lsa *X*12 = (−*b*± *D*)/ 2\*( *a*)*;*

*aks holda, D<0* bo‘lsa haqiqiy echimi yo‘q.

Bu amallar ketma-ketligiga mos algoritm 1.3-rasmda blok-sxema ko‘rinishida keltirilgan.



1.3-rasm. Kvadrat tenglamani yechish blok-sxemasi

**5. Chiziqli algoritmlar**

Har qanday murakkab algoritmni ham uch asosiy struktura yordamida tasvirlash mumkin. Bular ketma-ketlik, ayri va takrorlash strukturalaridir. Ushbu strukturalar asosida chiziqli, tarmoqlanuvchi va takrorlanuvchi hisoblash jarayonlarining algoritmlarini tuzish mumkin. Umuman olganda, algoritmlarni shartli ravishda quyidagi turlarga ajratish mumkin:

*chiziqli* algoritmlar;

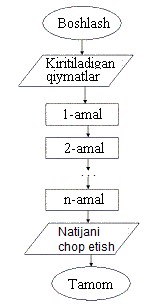
*tarmoqlanuvchi* algoritmlar;

*takrorlanuvchi* algoritmlar;

ichma-ich joylashgan takrorlanuvchi algoritmlar;

*rekurrent* algoritmlar;

takrorlanishlar soni oldindan no’malum algoritmlar; - ketma-ket yaqinlashuvchi algoritmlar.



1.4-rasm. Chiziqli algoritmlar blok–sxemasining umumiy tuzilishi

Faqat ketma-ket bajariladigan amallardan tashkil topgan algoritmlarga - *chiziqli* algoritmlar deyiladi. Bunday algoritmni ifodalash uchun ketma-ketlik strukturasi ishlatiladi. Strukturada bajariladigan amal mos keluvchi shakl bilan ko‘rsatiladi. Chiziqli algoritmlar blok-sxemasining umumiy tuzilishi 1.4-rasmda keltirilgan.

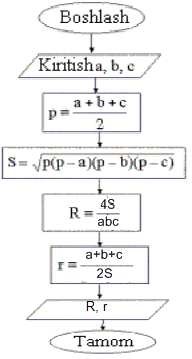
1*-misol.* Uchburchak tomonlarining uzunligi bilan berilgan. Uchburchakka ichki *r* va tashqi *R* chizilgan aylanalar radiuslarini hisoblang.

Ichki chizilgan aylana radiusi *r = (a+b+c)/2S*, tashqi chizilgan aylana

4*S* radiusi *R=*  formulalar orqali hisoblanadi. Bu yerda *S -* uchburchakning yuzi, *a, abc*

*b, c* – uchburchak tomonlarining uzunliklari. Masala echimining blok-sxemasi

1.5-rasmda keltirilgan.



1.5-rasm. Uchburchakka ichki va tashqi chizilgan aylanalar radiuslarini hisoblash bloksxemasi

2-*misol*. Quyida keltirilgan munosabatni hisoblash algoritmini ko‘rib chiqaylik. Jarayon amallarni ketma-ket bajarilishidan iborat.

z x= 2 + sin(x + y) ,

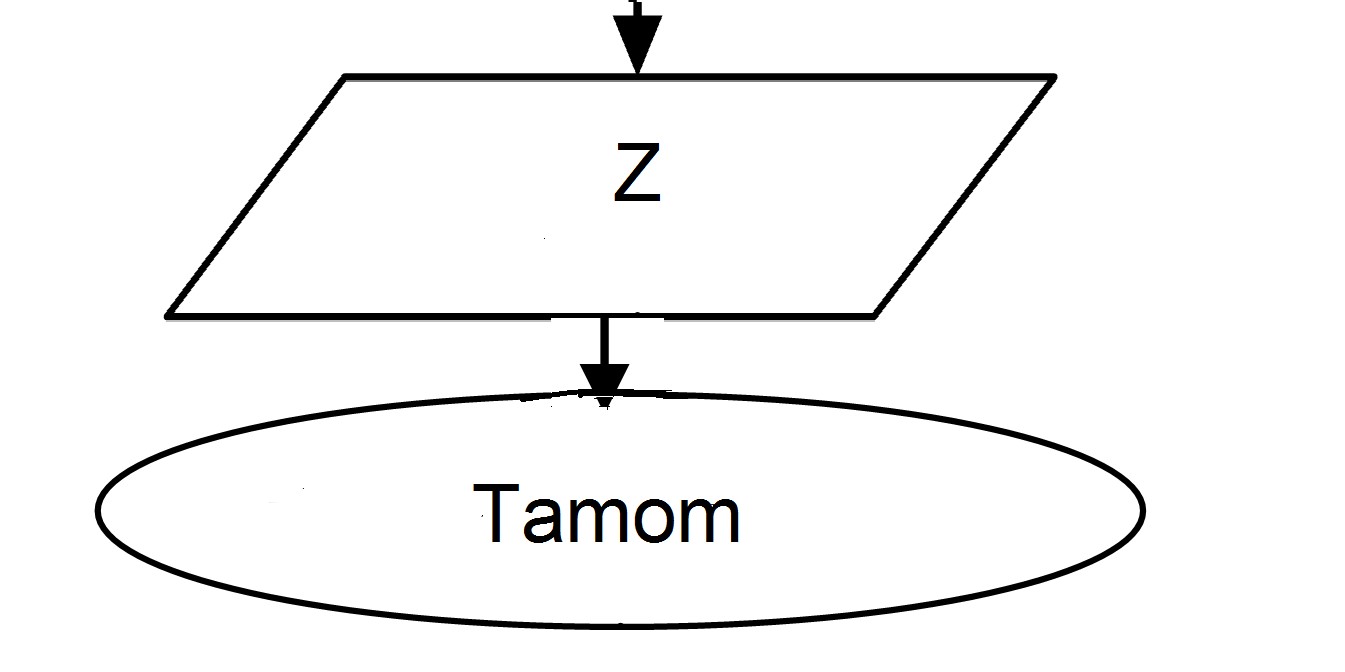
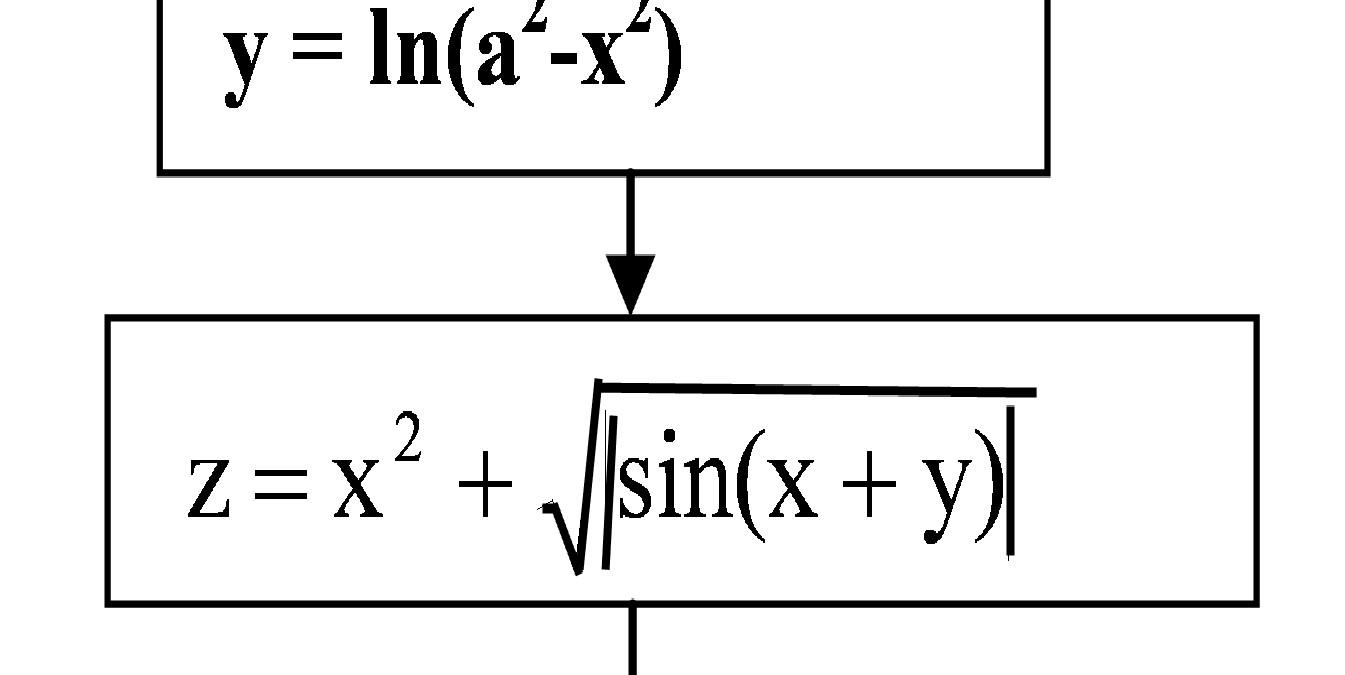
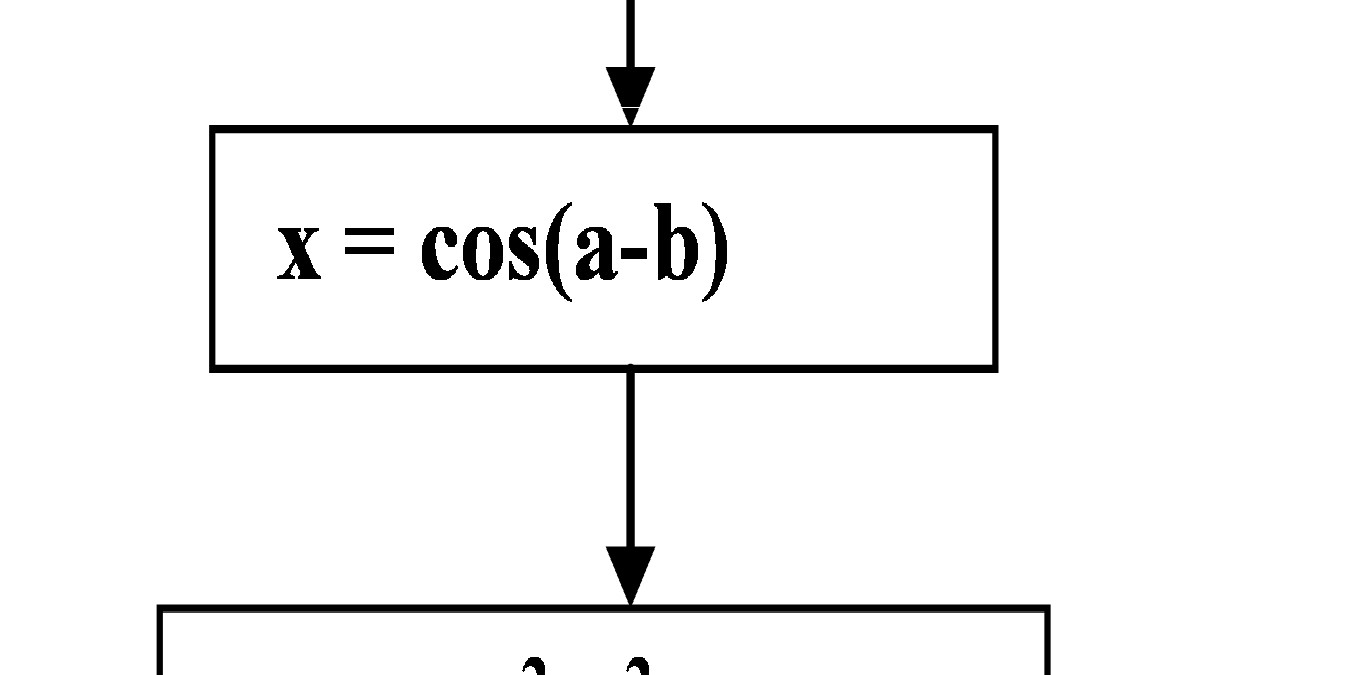
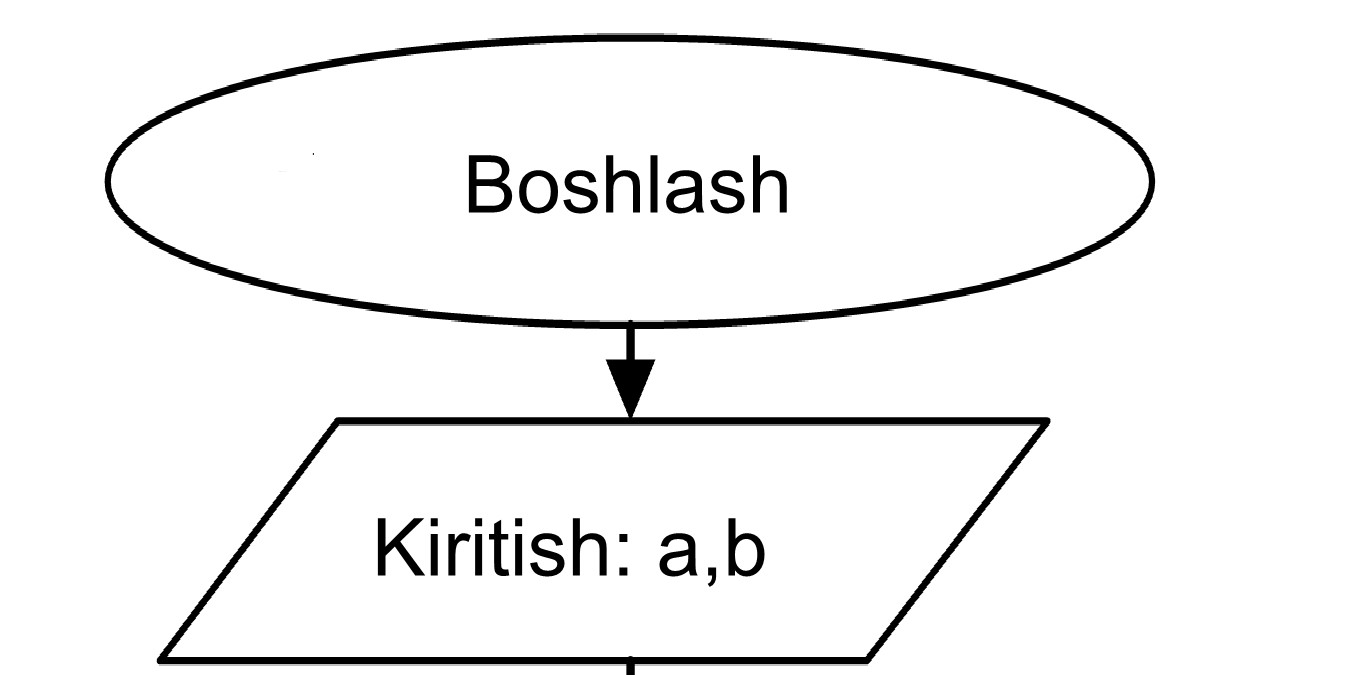
bu yerda x = cos ( a – b ), y = ln ( a2 - x2 ), a = 0.7, b = 2.1.

Bunda:

a, b - aniq qiymatga ega bo‘lgan boshlang‘ich ma’lumotlar; x, y - oraliq ma’lumotlar; z - natija.

Masalani yechish jarayoni chiziqli hisoblanadi, chunki boshlang‘ich ma’lumotlar kiritilgach, munosabatlarning qiymati dasturda joylashgan tartibda hisoblanadi, ya’ni dastlab *x*, so‘ng - *y* qiymati va nihoyat *z* natija hisoblanadi.

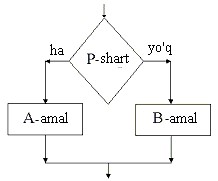
Mazkur jarayonning blok-sxemasi 1.6-rasmda keltirilgan.



1.6-rasm. Hisoblash blok-sxemasi

**6. Tarmoqlanuvchi algoritmlar**

Agar hisoblash jarayoni biror-bir berilgan shartning bajarilishiga qarab turli tarmoqlar bo‘yicha davom ettirilsa va hisoblash jarayonida har bir tarmoq faqat bir marta bajarilsa, bunday hisoblash jarayonlari tarmoqlanuvchi algoritmlar deyiladi. Tarmoqlanuvchi algoritmlarni tasvirlash uchun “ayri” tuzilmasi ishlatiladi. Tarmoqlanuvchi tuzilmasi berilgan shartning bajarilishiga qarab ko‘rsatilgan tarmoqdan faqat bittasining bajarilishi ta’minlanadi (1.7-rasm).



1.7-rasm. Tarmoqlanishning umumiy ko‘rinishi

Berilgan R-*shart* romb figurasi ichida tasvirlanadi. Agar *shart* bajarilsa, "ha" tarmoq bo‘yicha A-amal, aks holda (*shart* bajarilmasa) "yo‘q" tarmoq bo‘yicha V-amal bajariladi.

1-*misol*. Tarmoqlanuvchi algoritmga *misol* sifatida quyidagi sodda *masala* keltiriladi:

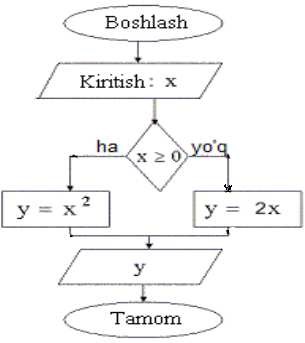
*x*2 , *agar x* ≥ 0 *у*=

2x, aks holda.

Natijaviy qiymat *y* berilgan *x* ning qiytmatiga bog‘liq holda bo‘ladi: agar x≥0 shart *rost* bo‘lsa, tarmoq bo‘yicha *y = x2* munosabatning qiymati, aks holda, *y* = *2\*x*munosabatning qiymati hisoblanadi. Bu masala bajarilishining so‘z bilan ifodalangan algoritmi quyidagicha:

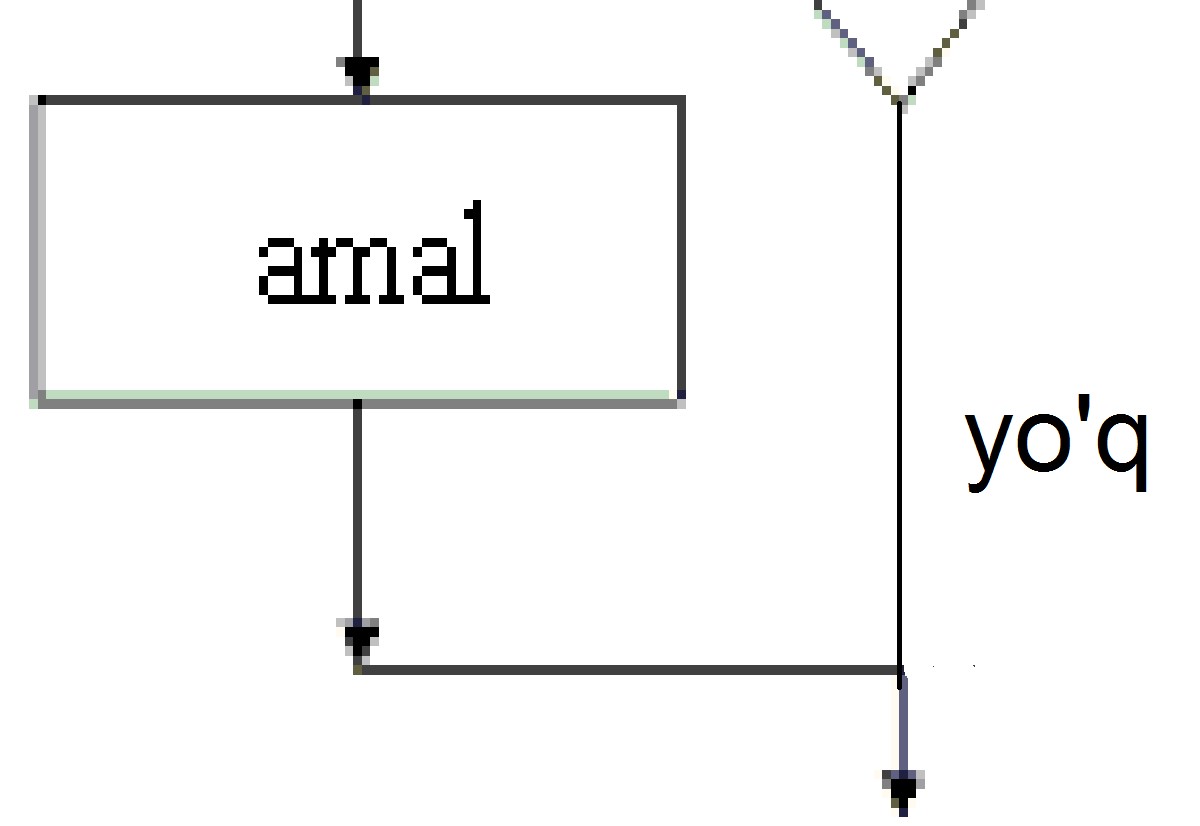
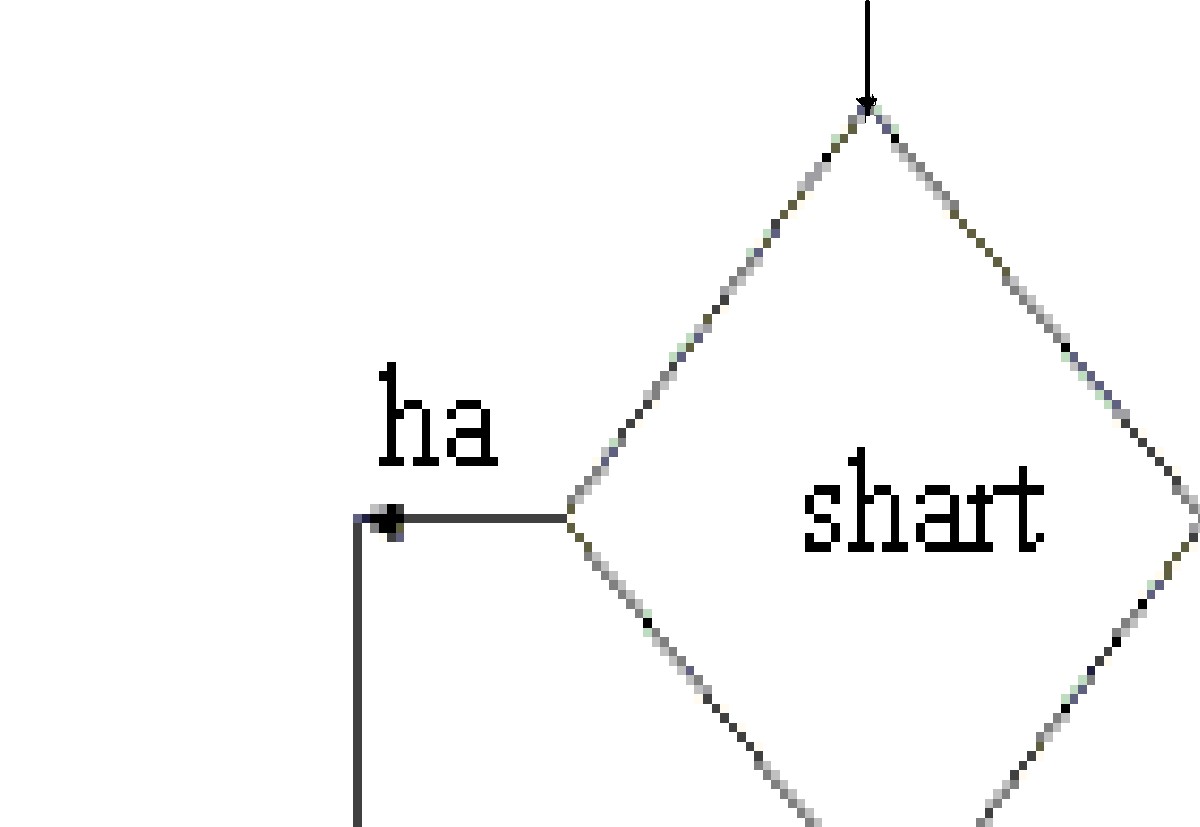
agar ( x ≥ 0 ) shart bajarilsa, u holda u=x2, aks holda u=2\*x.

Masala echimining blok-sxemasi 1.8-rasmda keltirilgan.



1.8-rasm. Interval ko‘rinishidagi funksiya qiymatini hisoblash blok-sxemasi

Ko‘pgina masalalarni yechishda, *shart* asosida tarmoqlanuvchi algoritmning ikki tarmog‘idan biri, ya’ni «rost» yoki «yolg‘on»ning bajarilishi etarli bo‘ladi. Bu holat tarmoqlanuvchi algoritmning xususiy holi sifatida qisqartirilgan strukturasi deb atash mumkin. Qisqartirilgan struktura blok-sxemasi quyidagi ko‘rinishga ega (1.9-rasm).



1.9-rasm. Qisqartirilgan strukturaning umumiy ko‘rinishi

2-*misol*.Berilgan *x, u, z* sonlari ichidan eng kattasini aniqlang. Ushbu masalaga mos matematik modelni quyidagicha tasvirlash mumkin: *p*= max{*x*, *y*,*z*}.

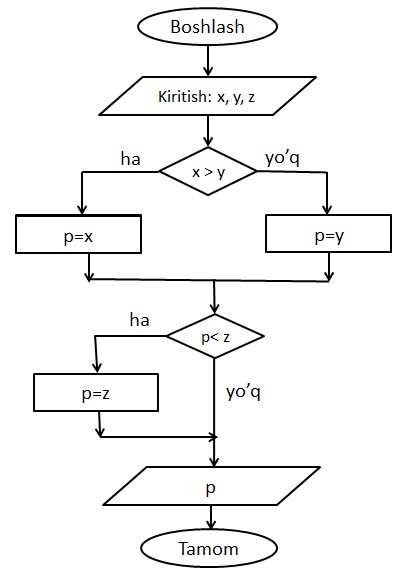
Berilgan *x, y, z* sonlardan eng kattasi *p* deb belgilangan. So‘zlar orqali ifodalangan algoritm asosida masala echimini quydagicha tasvirlash mumkin: 1) kiritish (*x, y, z*);

agar ( x > u ) bo‘lsa, u holda p = x, aks holda p = u;

agar (*r < z* *)* bo‘lsa, u holda *p = z; 4)* muhrlash (*r*).

Keltirilgan algoritmga mos blok-sxema 1.10-rasmda tasvirlangan.

Bu algoritmda, avva, *x* va *y*  o‘zaro solishtiriladi, katta qiymatligi ega r ga yuklanadi. So‘ngra *x* va *y*  larning kattasi deb aniqlangan *r* va *z* o‘zarosolishtiriladi. Agar  *r < z* sharti bajarilsa, u holda eng katta qiymat *p= z* deb olinadi, aks holda boshqarish navbatdagi amalga uzatiladi. Natijada *p* da uchta qiymatdan eng kattasi aniqlanadi.



1.10-rasm. Berilgan *x, y, z* sonlar ichidan eng kattasini topish blok-sxemasi

Ushbu masalani yechish algoritmining yana bir usulini ko‘rib chiqamiz.

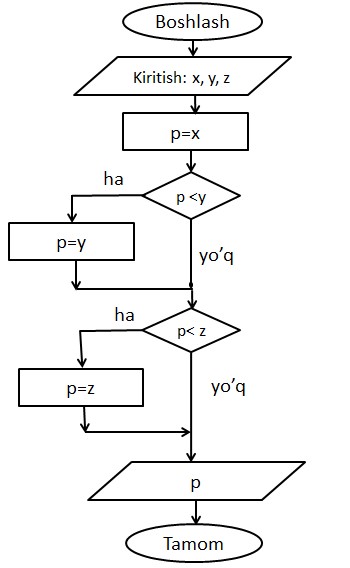
kiritish (*x, y, z*);

*p* = *x*;

agar (*p < y )* bo‘lsa, u holda *p= y;* 4) agar (*p < z* *)* bo‘lsa, u holda *p= z;* *5)* muhrlash (*r*).

Bu algoritmga mos blok-sxema 1.11-rasmda tasvirlangan.

Bu usulga asosan, avvalo sonlar ichida birinchisi eng kattasi deb faraz qilinadi, ya’ni *p* = *x.* So‘ngra har bir qadamda navbatdagi son *– r* ning qiymati bilan solishtiriladi va shart bajarilsa, u eng kattasi deb qabul qilinadi.Bu algoritmning afzalligi shundaki, uning asosida uchta va undan ko‘p sonlar ichidan eng kattasini (kichigini) topishning qulay imkoniyati mavjud.



1.11-rasm. Hisoblash blok-sxemasi

3-*misol*. Quyidagi ifoda bilan berilgan munosabatni hisoblang [2, 52 b.].

*b* − *x*, *agar x* > 0,



*Y* = *x* + *a*, *agar x* < 0,

a + b, aks holda .

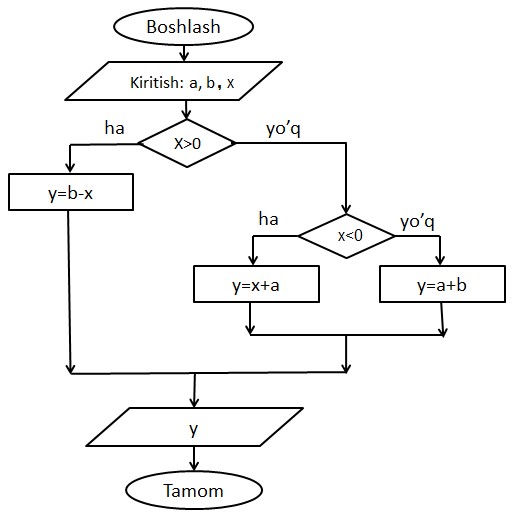
Bu misol natija *x* ning qiymatiga bog‘liq shart bilan berilgan va masala quyidagicha so‘zlar orqali ifodalangan algoritm asosida aniqlanadi: *agar x > 0 bo‘lsa, u holda u = b - x bo‘ladi, aks holda; agar x < 0 bo‘lsa, u holda u = x + a, aks holda u = a+ b.*

Avvalo, birinchi shart tekshiriladi va agar u bajarilsa, *y = b - x* amal

*x* + *a*, *agar x* < 0, bajariladi, aks holda *Y* =  munosabat hisoblanadi.

a + b, aks holda .

Bu fikrlar quyidagi blok-sxemada o‘z aksini topgan (1.12-rasm).



1.12-rasm. Hisoblash blok-sxemasi

**7. Takrorlanuvchi algoritmlar**

Agar biror masalani yechish uchun zarur bo‘lgan amallar ketma-ketligining ma’lum bir qismi biror parametrga bog‘liq holda ko‘p marta qayta bajarilsa, bunday jarayon takrorlanuvchi algoritm deyiladi. Takrorlanuvchi algoritmlarga misolsifatida odatda qatorlarning yig‘indisi yoki ko‘paytmasini hisoblash jarayonlarini qarash mumkin.

*1-misol*. Birdan *n* gacha bo‘lgan natural sonlarning yig‘indisini hisoblash algoritmini tuzaylik. Masalaning matematik modeli quyidagicha:

n

*S* =1+ 2 + 3+...+ *n* =∑*i*

*i*=1

Bu yig‘indini hisoblash uchun, avvalo, natiga boshlangich qiymatini S=0 va indeksning boshlangich qiymatini *i* =1 deb olamiz va joriy amallar S = S + *i* va *i* = *i* + 1 hisoblanadi. Bu erda birinchi va ikkinchi qadamlar uchun yig‘indi hisoblandi va keyingi qadamda *i* parametr yana bittaga orttiriladi va navbatdagi qiymat avvalgi yig‘indi *S* ga qo‘shiladi. Mazkur jarayon shu tartibda indeksning joriy qiymati *i≤ n* sharti bajarilmaguncha davom ettiriladi va natijada, izlangan yig‘indiga ega bo‘lamiz. Ushbu fikrlarni quyidagi so‘zlar orqali ifodalangan algoritm bilan ifodalash mumkin:

kiritish (n);

S= 0 - natijaning boshlang‘ich qiymati;

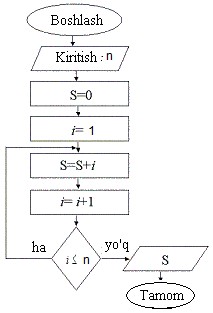
i= 1 - indeksning boshlang‘ich qiymati;

S= S + i - natijaning joriy qiymatini hisoblang;

i= i+ 1- indeksning joriy qiymatini hisoblang;

agar (i ≤ n) sharti tekshirilsin va u bajarilsa => (4) ; 7) muhrlash (S).

Bu jarayonga mos keladigan blok-sxemaning ko‘rinishi 1.13-rasmda tasvirlangan.



1.13-rasm. 1 dan n-gacha bo‘lgan sonlar yig‘indisini hisoblash blok-sxemasi

Yuqorida keltirilgan so‘zlar asosida ifodalangan algoritm va blok-sxemadan ko‘rinib turibdiki, amallar ketma-ketligining ma’lum qismi parametr *i* ga nisbatan *n* marta takrorlanadi.

*2-misol.* Quyidagi ko‘paytmani hisoblash algoritmi va blok-sxemasini tuzaylik: *P= 1*⋅2⋅3⋅⋅⋅*n = n!* (odatda, 1 dan *n* gachabo‘lgan natural sonlarning

ko‘paytmasi *n!* ko‘rinishda belgilanadi va “en” faktorial deb ataladi: P=n!

n

( jarayonning matematik modeli: *P* =∏*i* ) [2, 57-58 b.].

*i*=1

Ko‘paytmani hosil qilish algoritmi ham yig‘indini hosil qilish algoritmiga o‘xshash, faqat ko‘paytmani hosil qilish uchun, avvalo, *i*=*1* da *P*=1deb olinadi, so‘ngra *i*= *i* +1da *P*= *P*∗*i* munosabatlar hisoblanadi. Keyingi qadamda *i* parametrning qiymati yana bittaga orttiriladi va navbatdagi qiymat avvalgi hosil bo‘lgan ko‘paytma - *P* ga ko‘paytiriladi. Bu jarayon shu tartibda to *i ≤* n sharti bajarilmaguncha davom ettiriladi va natijaviy ko‘paytmaning qiymatiga ega bo‘lamiz. Quyidagi so‘zlar orqali ifodalangan algoritmda bu fikrlar o‘z aksini topgan:

kiritish *(n);*

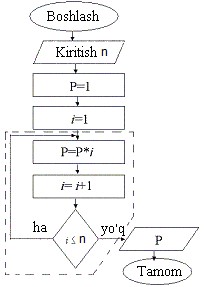
*P*=1 *-* natijaning boshlang‘ich qiymati; 3) *i*=1- indeksning boshlang‘ich qiymati;

*P*= *P*∗*i -* natijaning joriy qiymatini hisoblash;

*i*= *i*+1 *-* indeksning joriy qiymatini hisoblash;

*agar (i <= n)* shart bajarilsa, u holda => (4); 7) muhrlash *(P).*

Bu algoritmga mos blok-sxema 1.14-rasmda keltirilgan.



1.14-rasm. 1 dan n gacha bo‘lgan sonlar ko‘paytmasini hisoblash blok-sxemasi

Yuqorida ko‘rilgan yig‘indi va ko‘paytmalarning blok-sxemalaridagi takrorlanuvchi qismlariga (punktir chiziqlar ichiga olingan) 1.14 rasmdagi sharti keyin berilgan takrorlanuvchi struktura mos kelishini ko‘rish mumkin.

*3-misol*. Yuqoridagi blok-sxemalarda shartni oldin tekshiriladigan holatda

n

chizish mumkin edi. Masalan, *S* =∑*i* yig‘indini xisoblash algoritmi tadqiqi *i*=1

keltiriladi. Bu masalani yechishda algoritmning takrorlanuvchi qismiga quyidagi sharti oldin berilgan takrorlanuvchi strukturaning mos kelishini ko‘rish mumkin.

kiritish (n);

S=0;

i = 0;

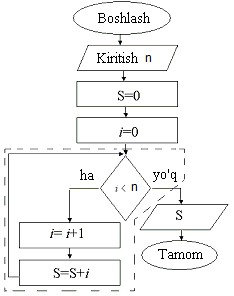
agar ( i > n ) => (8);

i = i + 1;

S = S + i ;

shartsiz o‘tish=> (4); 8) muhrlash (S).

Bu algoritmga mos blok-sxema 1.15- rasmda keltirilgan.



1.15-rasm. 1 dan *n* gacha bo‘lgan sonlar yig‘indisini hisoblash blok-sxemasi

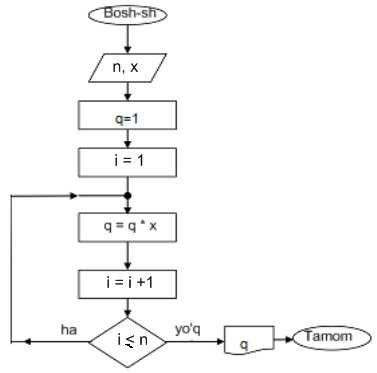
4-*misol.* Haqiqiy *x* sonining n chi darajasini hisoblash masalasi ko‘riladi.

Uning matematik modeli: *q* = *xn* ko‘rinishga ega.

Takrorlanuvchi jarayonni tashkil etish quyidagidan farqli, yuqoridagilar bilan bir xil:

- ko‘paytirish jarayoni uchun boshlang‘ich qiymat berilishi: q = 1 ; - joriy natijani hisoblash: q = q \* *x*  ifoda bo‘yicha amalga oshiriladi.

Shunday qilib, *x* ning *n* chi darajasini hisoblash uchun takrorlanuvchi jarayonni tashkil etish blok-sxemasi 1.16-rasmda keltirilgan.



1.16-rasm. Hisoblash blok-sxemasi

5-*misol*. Quyidagi munosabatni hisoblash kerak bo‘lsin [2, 55-56 b.]:

n x i

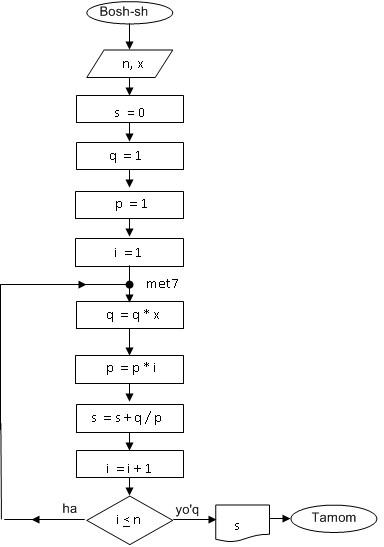
*S* = ∑ .

*i* =1 *i*!

Munosabatni ochib, quyidagi ko‘rinishda yozish mumkin: s = x1 /1! + x 2 /2! + … + xn / n! .

Masalani yechish algoritmida boshlang‘ich qiymat sifatida *s=*0 ni olamiz, chunki ifodada yig‘indi belgisi mavjud. Yig‘indi belgisi ostidagi munosabat kasr sonni anglatadi: suratda - *x i* , mahrajda - *i* **!**. Ularning har biri uchun boshlang‘ich va joriy munosabatlar shakllantiriladi:

|  |  |  |  |
| --- | --- | --- | --- |
|  | surat | mahraj | natija |
| boshlang‘ich munosabat | q = 1 | p = 1 | s=0 |
| joriy munosabat | q = q \* *x* | p = p \* *i* | s = s + q / p |



1.17-rasm. Hisoblash blok-sxemasi

Bu jarayonni shakllantirish uchun ***i*** indeks-parametri ishlatiladi.

Indeks-parametrni boshqarish amallari quyidagicha:

*i =* 1–parametrning boshlang‘ich qiymati,

*i = i +* 1 – parametrning orttirmasi (orttirma h=1),

*i ≤ n –* jarayon yakunlanish sharti.

Bunga muvofiq, masalani yechish blok-sxemasi quyidagi 1.17-rasmdagi ko‘rinishga ega bo‘ladi.

6-*misol*. A={ai} (i=1, 2, …, n) massiv elementlarining yig‘indisini hisoblash jarayonini aks ettiradigan algoritm yarating.

n

Masalaning matematik modeli quyidagidan iborat: *S*=∑*ai* .

*i*=1

Yig‘indini hisoblash uchun *S* o‘zgaruvchidan foydalanamiz va uning boshlang‘ich qiymati deb *S* = 0 olinadi. So‘ngra indeksning *i* = 1 qiymatidan boshlab, uning *i* = *i* + 1 orttirmasi bilan to ( *i* <= n ) shart bajarilguncha *S* = *S* + a *i* munosabat qiymati ketma-ket hisoblanadi.

Quyidagi algoritmda jarayon amallari bajarilishi ketma-ketligi keltiriladi:

kiritish (n, a i );

*S* = 0,

*i* = 1,

*S* = *S* + a *i* ,

*i* = *i* + 1,

agar ( *i* <= n ) shart bajarilsa, u holda => (4), 7) muhrlash (*S*) .

*7-misol*. Massiv elementlari o‘rta qiymatini hisoblash. Masalaning

1 *n*

matematik modeli : *Ð*= ∑*ai* .Yuqoridagi masaladan farqi –

n i=1

elementlar yig‘indisini elementlar soniga bo‘lish amali bilan algoritm to‘ldiriladi, ya’ni:

kiritish (n, a i );

S = 0;

*i* = 1;

S = S + a *i* ;

*i* = *i* + 1;

agar ( *i* <= n ) shart bajarilsa, u xolda => (4); 7) P =S / n ; 8) muhrlash (P) .

8-*misol*. Massiv elementlari qiymatlarining ko‘paytmasini hisoblash

*n* algoritmini tuzing. Masalaning matematik modeli quyidagidan iborat: *P* =∏*ai* .

*i*=1

Hisoblash jarayoni yuqoridagiga o‘xshash bo‘ladi, faqat ko‘paytmaning boshlang‘ich qiymati R = 1 va joriy amal R = R \* a*i* bo‘ladi. Bu jarayonning so‘zlar orqali ifodalangan algoritmi quyidagicha:

kiritish (n, ai );

R = 1;

*i* = 1;

R = R \* ai ;

*i* = *i* + 1;

agar ( *i* <= n ) shart bajarilsa, u holda => (4) 7) muhrlash (R) .

9-*misol*. B={bi} massiv elementlari maksimal (eng katta) qiymatini aniqlash bilan bog‘liq masala ko‘riladi.

Mazkur masalaning matematik modeli quyidagi ko‘rinishga ega:

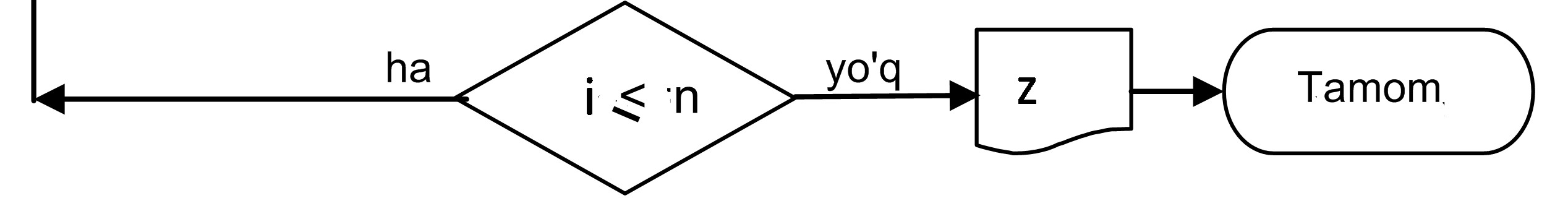
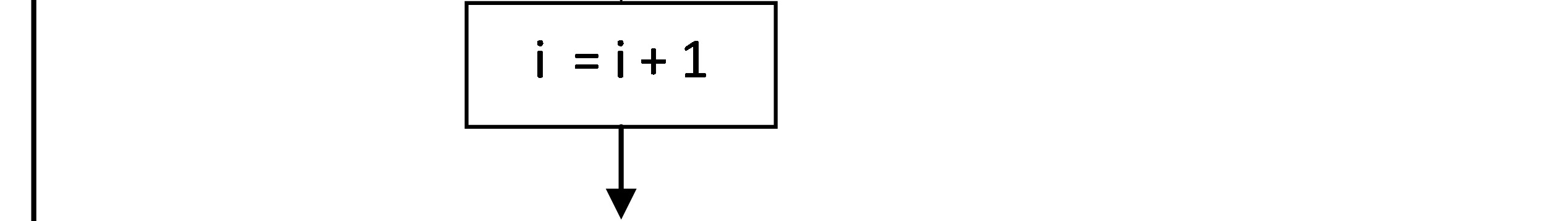
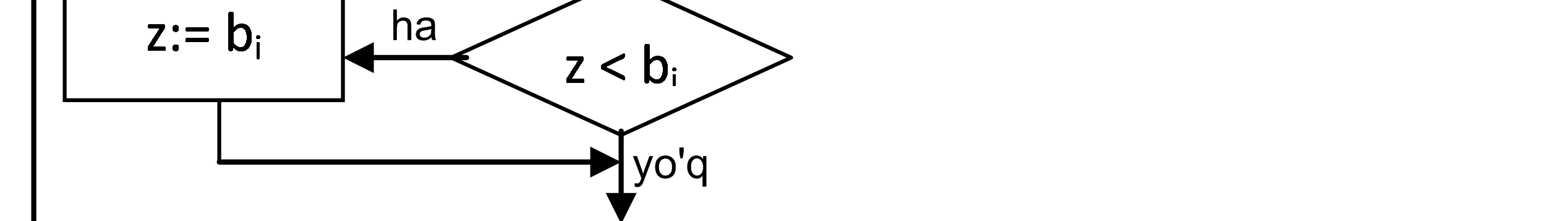
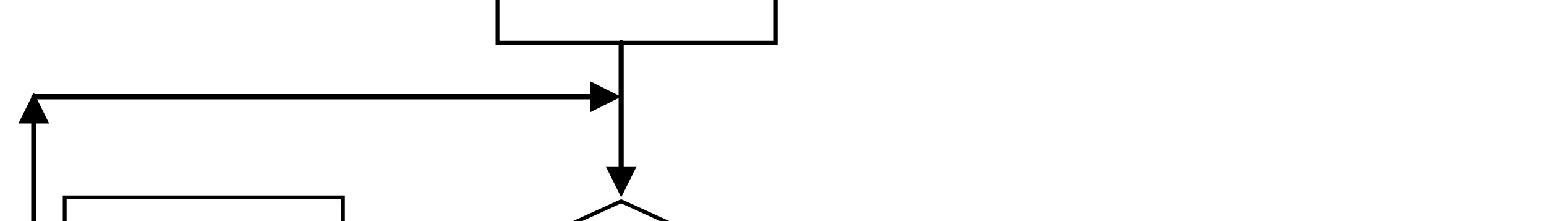
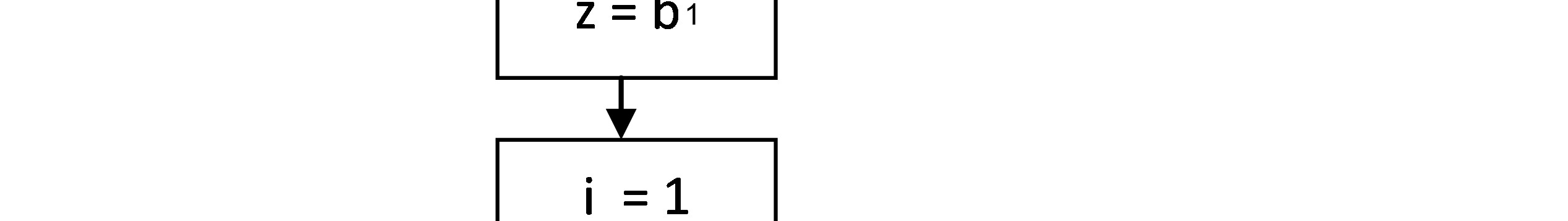
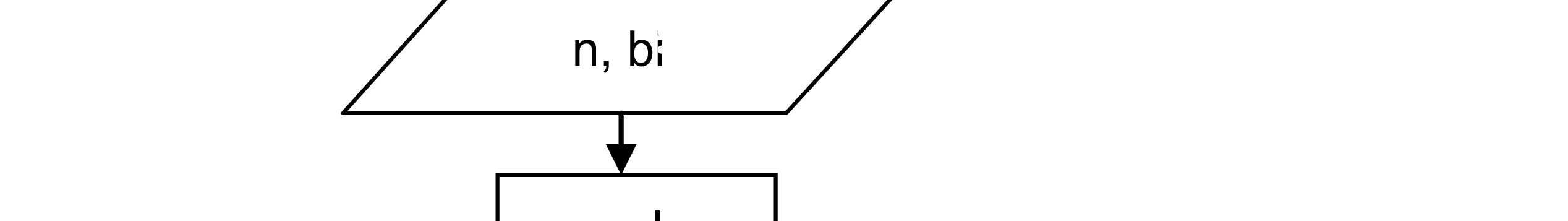
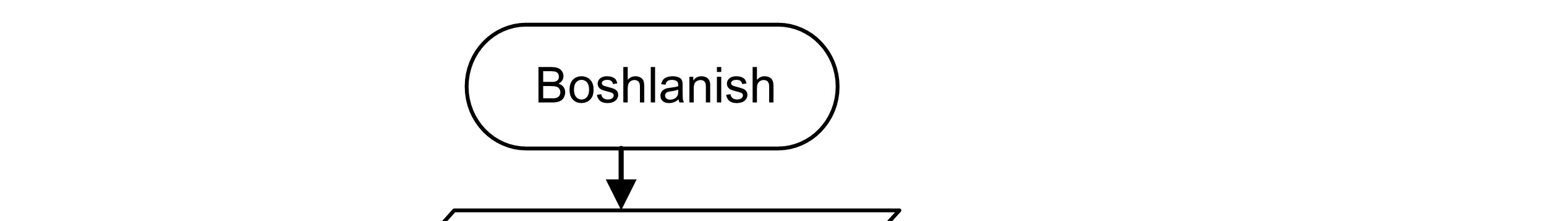
*z*= max*bi m*=8.

1≤ ≤*i m*

Maksimal elementni aniqlash uchun quyidagi tadbirni amalga oshirish zarur. Avval, massivning birinchi elementi maksimal qiymatga ega deb taxmin qilinadi. So‘ngra taxmin qilingan maksimal element navbatdagi elementlar bilan navbatmanavbat solishtiriladigan takrorlash jarayoni tashkil etiladi. Agar massivning navbatdagi elementi maksimal deb belgilangan elementdan katta bo‘lsa, u holda joriy element maksimal deb belgilanadi. Takrorlashning yakunida o‘zgaruvchining qiymati massivning maksimal elementiga mos keladi.

Massivning maksimal elementini aniqlash algoritmi blok-sxemasi ko‘rinishi

1.18-rasmda keltirilgan.



1.18-rasm. Hisoblash blok-sxemasi

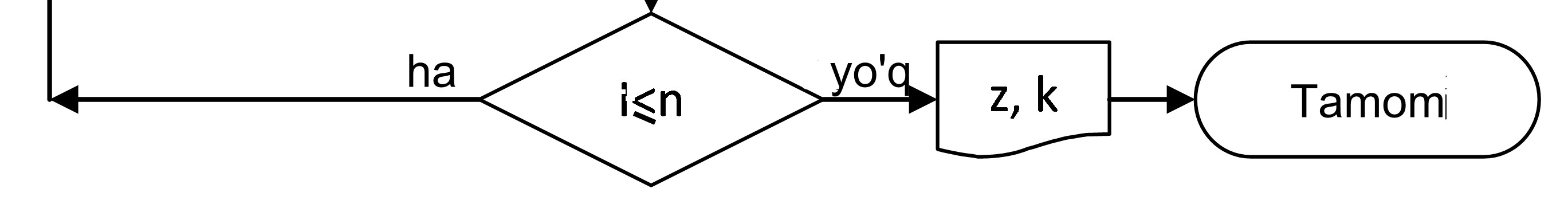
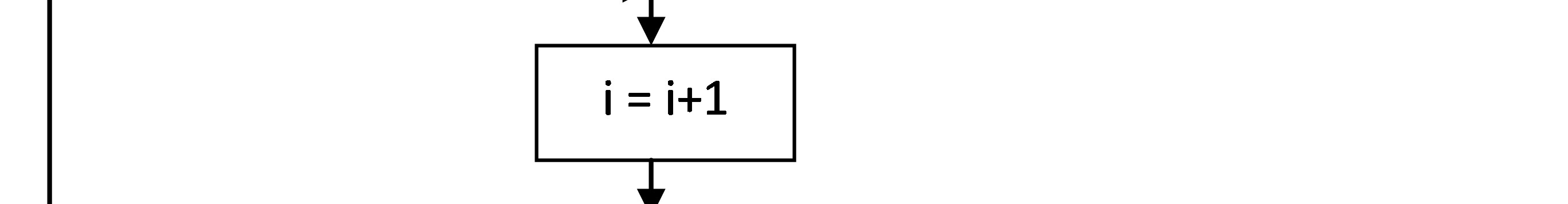
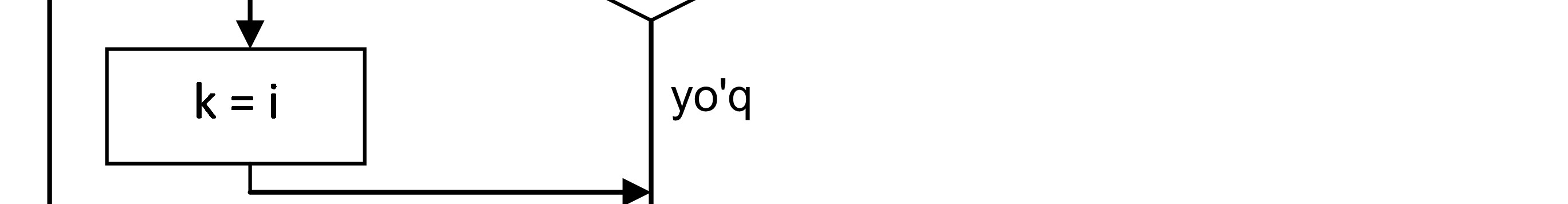
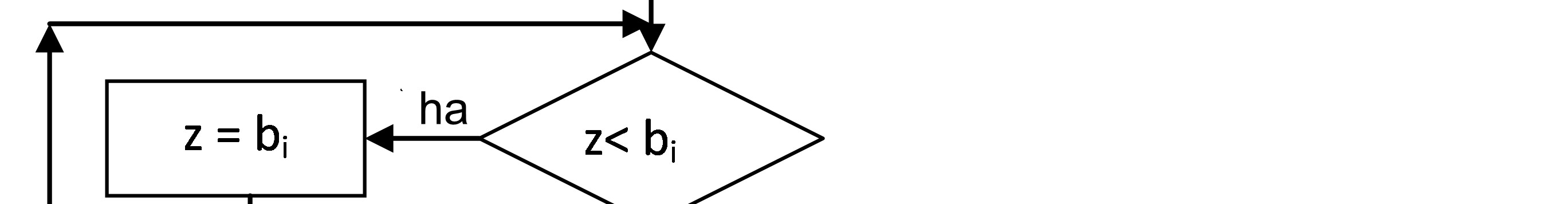
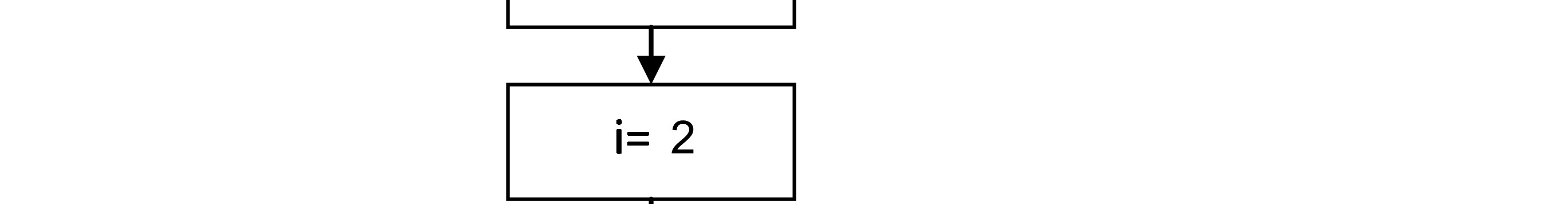
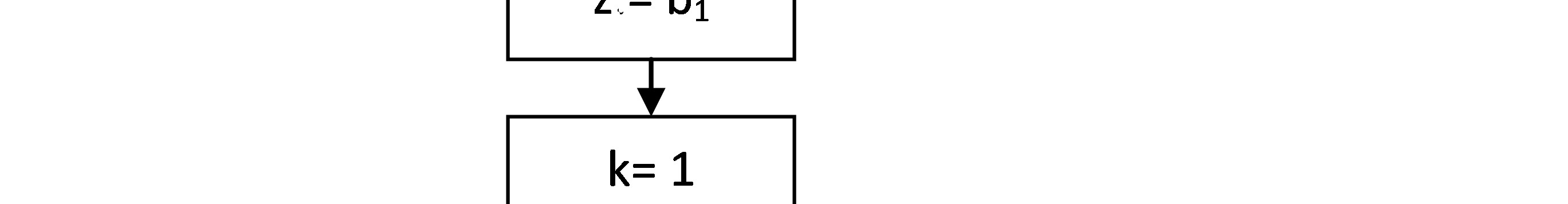
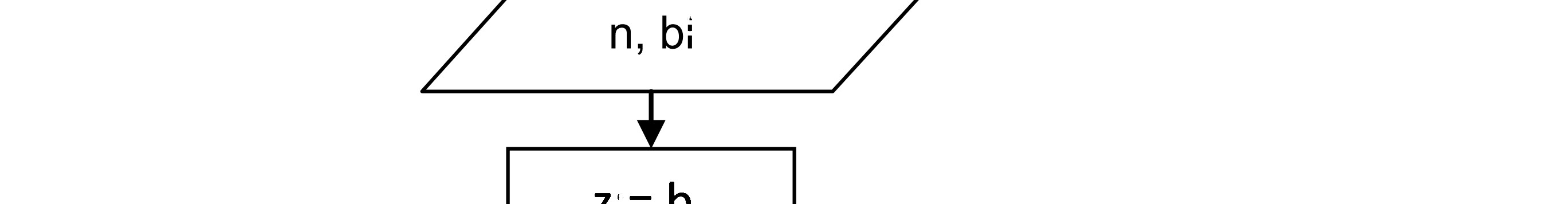
Minimal elementni aniqlash uchun shart ifodasida “**<”** (kichik) belgisini “**>”** (katta) belgiga o‘zgartirishning o‘zi kifoya.

*10-misol.* Massivning maksimal elementi indeksini, ya’ni u joylashgan o‘rnini aniqlash uchun yuqorida keltirilgan algoritmga boshlang‘ich va joriy elementining indeksini belgilaydigan o‘zgaruvchi qo‘shishning o‘zi kifoya:

*k* = 1 (birinchi element maksimal deb taxmin qilanadi);

k = *i* (agar joriy *i* – chi element taxmin qilingan maksimumdan katta bo‘lsa, u qiymati bo‘yicha barcha elementlardan eng kattasi bo‘ladi).

Qo‘shimchalarni hisobga olgan holda blok-sxema 1.19-rasmda keltirilgan.



1.19-rasm. Hisoblash blok-sxemasi

Algoritmning so‘zlar orqali ifodalangan usulidan foydalanib, amallar ketmaketligini keltiramiz:

kiritish (n, *b*i )

z = *b*1;

k = 1; 4) *i* = 2;

agar ( z < b *i* ) shart bajarilsa, u holda { z= bi; k=i; }

*i* = *i* + 1;

agar ( *i* ≤ n ) shart bajarilsa, u holda => (5) 8) muhrlash (S, k).

*11-misol*. Vektorni vektorga skalyar ko‘paytmasi: s = A\*V ni hisoblash masalasi ko‘riladi (vektorlar skalyar ko‘paytmasi).

Bu yerda: A={ a i }, B={ b i }, *i* =1,2, ..., n, c – skalyar.

Jarayonning matematik modeli ( hisoblash formulasi) :

n

*s* = ∑*ai* ∗*bi* = *a*1 ×*b*1 + *a*2 ×*b*2 + ⋅⋅⋅ + *an* ×*bn* ,

*i*=1

Bu munosabatni hisoblash - vektorlarning mos elementlari ko‘paytmalari yig‘ishdan iborat.

Algoritmning so‘zlar orqali ifodalangan usulidan foydalanib, amallar ketmaketligi keltiriladi:

kiritish (n, a *i*, b *i*)

S = 0;

*i* = 1;

S = S +a i \* b i ;

*i* = *i* + 1;

agar ( i <= n ) shart bajarilsa, u holda => (4); 7) muxrlash (S).

*12-misol*. A={ai} (i=1, 2, …, n) massiv elementlari qiymatlari yig‘indisidan eng katta elementi qiymatini ayrish jarayonini akslantiradigan algoritm yarating.

Masalaning matematik modeli quyidagidan iborat:

n

R = ai −max≤≤i n ai .

∑

1 *i*=1

Bu murakkab matematik model uchta nisbatan sodda munosabatlar ketmaketligi bilan almashtiriladi (dekompozitsiya amali):

n

1)*S* = ∑*ai* , 2)*P*= max*ai* , 3)*R* = *S* − *P*. *i*=1 1≤*i*≤*n*

Asosiy algoritmda amallar bajarilishi ketma-ketlikligi keltiriladi, ya’ni:

kiritish (n, m, a i );

S = 0;

*i* = 1;

S = S + a i ;

*i* = *i* + 1;

agar ( i <= n ) shart bajarilsa, u holda => (4);

P = a 1 ;

*i* = 2 ;

agar (P < a i ) shart bajarilsa, u holda P = a i ;

*i* = *i* + 1;

agar ( *i* <= n ) shart bajarilsa, u holda => (9) ;

R = S – P ;

muhrlash (R) .

Yuqoridagi keltirilgan masalani yechish algoritmini ixchamlashtirish mumkin:

kiritish (n, m, a i );

S = a 1 ;

P = a 1 ;

*i* = 2;

S = S + a i ;

agar (P < a i ) shart bajarilsa, u holda P = a i ; 7) *i* = *i* + 1;

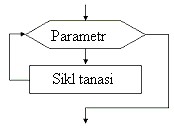
agar ( *i* <= n ) shart bajarilsa, u holda => (5) ;

R = S – P ;

muhrlash (R) .

Algoritmda yig‘indi va maksimal qiymat aniqlash jarayonida boshlang‘ich indeks qiymatini tenglashtiriladi (S = a 1 va P = a 1 ) va jarayon massivning 2 chi elementini qayta ishlashdan boshlandi. Ya’ni bir takrorlash jarayonida ikkita: massiv element qiymatlari yig‘indisini hisoblash va maksimal qiymatni aniqlash amalga oshiriladi.

Blok-sxemalarning takrorlanuvchi qismlarini quyidagi parametrli takrorlash strukturasi ko‘rinishida ham ifodalash mumkin (1.20-rasm).

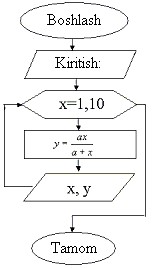


1.20-rasm. Parametrli takrorlash operatorining umumiy ko‘rinishi

13-*misol.* Parametrli takrorlash operatoriga masalasifatida berilgan *ax*

*x*=*1,2,3,.....10* qiymatlarda *y*= funksiyasining qiymatini hisoblash blok*a*+*x*

sxemasiga keltiriladi (1.21-rasm).



1.21-rasm. Parametrli takrorlash operatoriga doir blok-sxema