

erishishga olib kelmasligi mumkin. Bunga baʼzan algoritmning notoʻgʻri tuzilganligi yoki boshqa xatoliklar sabab boʻladi. Lekin salbiy *natija* ham natija deb qabul qilinadi.

3-misol. Tomonlarining uzunliklari a, b, c boʻlgan uchburchakning yuzi hisoblansin.

Quyida keltirilgan “Uchburchakning mavjud boʻlish shartini tekshirish” algoritmidan foydalanib, avval uchburchakning mavjud yoki mavjud emasligini aniqlaymiz. Agar uchburchak mavjud boʻlsa, uning yuzini hisoblaymiz, aks holda hisoblamaymiz. Bu ham natija sanaladi.

- 1) a, b, c qiymatlar aniqlansin;
- 2) agar $a \leq 0$, yoki $b \leq 0$, yoki $c \leq 0$ boʻlsa, uchburchak tomonlarining uzunligi manfiy son boʻla olmaydi, deb olinsin va 5-bosqichga oʻtilsin;
- 3) agar $((a+b \leq c)$ va $(a+c \leq b)$ va $(b+c \leq a))$ boʻlsa, uchburchak mavjud emas, deb olinsin va 5-bosqichga oʻtilsin;
- 4) yarim perimetr $P=(a+b+c)/2$ hisoblansin;
- 5) yuza: $S=\sqrt{(P*(P-a)*(P-b)*(P-c))}$ hisoblansin;
- 6) tugallansin.

Yuqoridagi barcha xossalar bajarilgandagina, koʻrsatma yoki buyruqlar ketma-ketligi algoritm boʻla oladi hamda u qandaydir (ijobiy yoki salbiy) natijaga ega boʻladi.

SAVOL VA TOPSHIRIQLAR



1. Qanday koʻrsatmalarni ijrochi bajara olmaydi?
2. Algoritmning asosiy xossalarini aytib bering.
3. Tushunarlilik xossasiga oid misol keltiring.
4. Koʻrsatmalar ijrochiga tushunarli boʻlishi uchun qanday tizimdan olinishi kerak?
5. Algoritmning diskretlilik xossasi mohiyatini tushuntirib bering.
6. Algoritmning natijaviylik xossasiga oid misollar keltiring.
7. Natijaviylik xossasi bajarilmaydigan koʻrsatmalar ketma-ketligiga misol keltiring.
8. Algoritmning ommaviylik xossasini misollar asosida tushuntiring.

UY VAZIFASI



1. x va y sonlarining oʻrta arifmetigini hisoblash algoritmini yozing.
2. Kalkulyator dasturi yordamida ixtiyoriy 3 ta sonning oʻrta geometrigini hisoblash algoritmiga oid kichik loyiha ishini tayyorlang (taqdimot koʻrinishida).

15–16-DARSLAR. ALGORITM TURLARI VA TASVIRLASH USULLARI

Avvalgi darsda koʻrilgan misollarda biz masalani yechish algoritmini soʻzlar va matematik formulalar orqali ifodalagan edik. Lekin algoritm boshqa koʻrinishlarda ham berilishi mumkin. Quyida biz algoritmni tasvirlashning eng keng tarqalgan usullari bilan tanishib chiqamiz.

BUNI BILASIZMI?



1. Algoritmni qanday koʻrinishlarda ifodalash mumkin?
2. Formulalardan iborat algoritm boʻlishi mumkinmi?
3. Grafiklardan iborat algoritm boʻlishi mumkinmi?

1. Algoritmning soʻzlar yordamida ifodalanishi. Bu usulda ijrochi uchun beriladigan har bir koʻrsatma tabiiy tildagi jumlar, soʻzlar orqali buyruq shaklida beriladi. Bunda algoritmning har bir buyrugʻi ijrochiga tushunarli soʻzlar orqali ifodalanadi.

Algoritmning soʻzlar yordamida ifodalanishiga oid misol keltirib oʻtamiz.

1-masala. Tomonlariga koʻra toʻgʻri toʻrtburchakning perimetri, diagonal va yuzasini hisoblash:

- 1) boshlansin;
- 2) tomonlar qiymati kiritilsin (a, b);
- 3) perimetr qiymati hisoblansin (P);
- 4) diagonal qiymati hisoblansin (D);
- 5) yuzasi hisoblansin (S);
- 6) perimetr, diagonal va yuza qiymatlari chop etilsin;
- 7) tugatilsin.

2. Algoritmning formulalar yordamida ifodalanishi. Bu usulda algoritmning har bir amali matematik formulalar yordamida ifodalanadi. Algoritm amallarini ifodalashda oddiy matematik yozuvlardan foydalanish mumkin. Bu usuldan matematika, fizika, kimyo kabi aniq fanlardagi formulalarni oʻrganishda foydalaniladi. Bu usulga baʼzan *analitik ifodalash* ham deyiladi.

Endi 1-masalaning formulalar yordamida ifodalanishini koʻrib chiqamiz:

- 1) boshlansin;
- 2) toʻrtburchak tomonlari a va b ning qiymatlari aniqlansin;
- 3) $P=2*a+2*b$;
- 4) $D=\sqrt{a^2+b^2}$;
- 5) $S=a*b$;
- 6) P, D va S qiymatlar chop etilsin;
- 7) tugatilsin.

3. Algoritmning jadval yordamida ifodalanishi. Algoritmning bu tarzda tasvirlanishidan ham koʻp foydalanamiz. Masalan, maktabda qoʻllanib kelinayotgan dars jadvali, Pifagor jadvali, kimyoviy elementlar jadvali va h. k. Funksiyalarning grafiklarini chizishda ham algoritmlar qiymatlarining jadvali koʻrinishlaridan foydalanamiz. Bunday jadvallardan foydalanish algoritmlari sodda boʻlganligi tufayli, ularni oʻzlashtirib olish oson.

Biror funksiyaning grafigini chizish uchun ham funksiyaning argument qiymatlariga mos qiymatlar jadvalini hosil qilamiz. Bu ham algoritmning jadval koʻrinishiga misol boʻla oladi. Masalan, $y=x^2+2$ algoritmi asosida harakat qilayotgan ijrochi oʻtadigan nuqtalarning baʼzilari koʻrsatilgan quyidagi jadval bilan matematikadan tanishsiz:

x	-4	-3	-2	-1	0	1	2	3	4
y	18	11	6	3	2	3	6	11	18

4. Algoritmning grafik shaklda ifodalanishi. Algoritmning bu koʻrinishdagi ifodasi sizga avvaldan tanish, chunki matematika kursida chizilgan grafiklarning koʻpchiligi algoritmning grafik usulda berilishiga misol boʻladi. Bundan tashqari, shahar yoki turar joy mavzelerida joylashgan uylar yoki inshootlarning joylashish sxemasi, biror uy hamda inshootlarni izlash

va harakatlanish bo'yicha berilgan karta-sxemalari, avtobuslarning yo'nalish sxemasi ham bunga misol bo'la oladi.

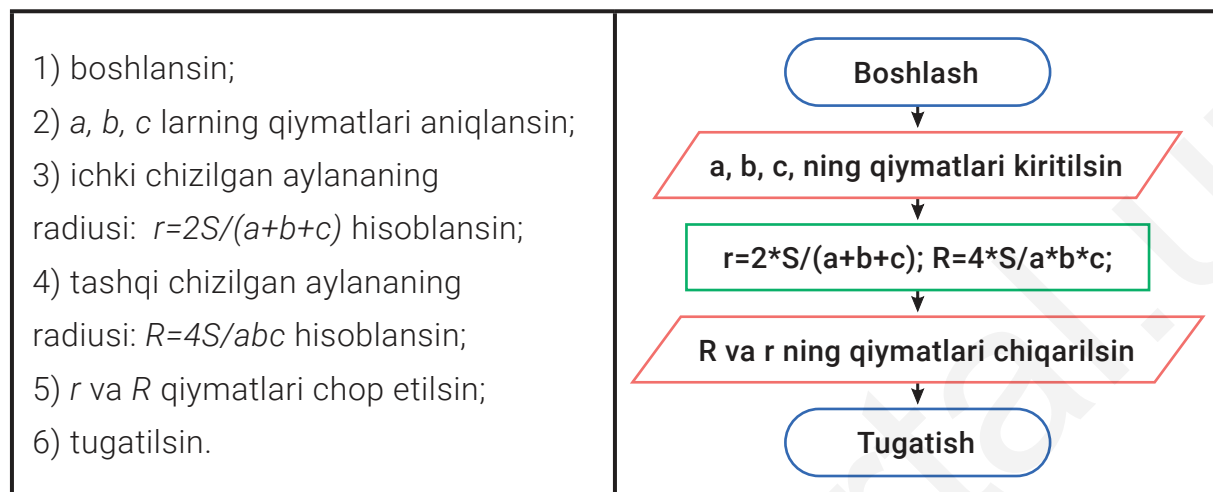
Algoritm lash asoslarini o'rganishning yana bir qulay grafik shakli blok-sxema usulidir. **Blok-sxemalar** yo'nalish chiziqlari orqali tutashtirilgan ma'lum buyruq yoki ko'rsatmani aks ettiruvchi maxsus geometrik shakl – bloklardan tashkil topadi.

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

Blokning nomi	Blokning ko'rinishi	Blokning vazifasi
Algoritmni boshlash / tugatish bloki		Algoritm blok-sxemasining boshlanishi va tugashida qo'llaniladi.
Kiritish/ chiqarish bloki		Dastlabki ma'lumotlar kiritishni tashkil qilish va olingan ma'lumotlarni chiqarish uchun xizmat qiladi.
Funksional blok (operator bloki)		Qiymat berish yoki tegishli ko'rsatmalarni bajarishga xizmat qiladi. To'rtburchak ichiga bajarilishi kerak bo'lgan buyruq yoziladi. Bir blokda bir nechta buyruqni yozish mumkin.
Muqobil blok		Shart tekshirish orqali algoritmning bajarilish yo'nalishi belgilanadi. Agar romb ichida yozilgan shart o'rinli bo'lsa, u holda boshqaruv "ha" tarmog'i, aks holda, "yo'q" tarmog'i bo'ylab uzatiladi.
Sikl bloki		Parametrli takrorlanish jarayonni tashkil qilishda qo'llaniladi. Siklning takrorlanishlar soni va qadami ma'lum bo'lishi kerak. Blokning ichida sikl parametrining boshlang'ich, yakuniy qiymati va uni o'zgartirish qadami ko'rsatiladi.
Qism dastur bloki		Oldindan yaratilgan yordamchi algoritmgacha murojaat qilish uchun ishlatiladi.
Xabarlarni chop etish bloki		Natijalarni bosib chiqarish uchun ishlatiladi.
Yo'nalish bloki		Blok-sxemadagi harakat yo'nalishini ko'rsatadi.
O'zlashtirish bloki	=	Qiymat berish ko'rsatmasi hisoblanadi.

2-masala. Uchburchak tomonlarining uzunligi bilan berilgan. Uchburchakga ichki va tashqi chizilgan aylanalar radiuslari va uzunliklari hisoblansin.

Ichki chizilgan aylana radiusi $r=2S/(a+b+c)$, tashqi chizilgan aylananing radiusi esa $R=4S/abc$ formulalar orqali hisoblanadi. Bu yerda S – uchburchakning yuzi, a, b, c – uchburchak tomonlarining uzunliklari.



Biz blok-sxemalar algoritmlarni ifodalashning qulay vositalaridan biri hamda ularning koʻrgazmalilik imkoniyati ancha katta ekanligini inobatga olgan holda, ulardan dasturlashni oʻrganish davomida foydalanib boramiz. Shu sababli blok-sxemalar bilan ishlashni hozirdan boshlab puxta oʻzlashtirib borishimiz zarur.

DIQQAT

5. Algoritmning dastur shaklida ifodalanishi.

Hozirgi kunda juda koʻp algoritmik tillar mavjud boʻlib, ularni *dasturlash* tillari deb ataymiz. *Algoritmik til* – algoritmlarni bir xil va aniq yozish uchun ishlatiladigan belgilashlar va qoidalar tizimi. Algoritmik til oddiy tilga yaqin boʻlib, u matematik belgilarni (yuqorida aytilganidek) oʻz ichiga oladi. Qoʻyilgan masalalarni yechish uchun tuzilgan algoritmlarni toʻgʻridan-toʻgʻri mashinaga kiritib, yechib boʻlmaydi, shu sababli yozilgan algoritmni birorta algoritmik tilga oʻtkazish zarur.

Har qanday algoritmik til oʻz qoʻllanilish sohasiga ega. Odatda, algoritmning kompyuter tushunadigan tilda yozilishi **dastur** deb ataladi. Kompyuter tushunadigan tilga **dasturlash tili** deyiladi. Jahonda minglab dasturlash tillari mavjud va ularning soni yanada ortib bormoqda. Hozirgi kunda **Pascal, Delphi, C, C++, Java, Phyton** dasturlash tillari keng tarqalgan va oʻrganish uchun qulay.

Yuqorida koʻrilgan algoritmlarni tasvirlash usullarining asosiy maqsadi qoʻyilgan masalani yechish uchun zarur amallar ketma-ketligining eng qulay holatini aniqlash va inson tomonidan dastur yozilishini yanada osonlashtirishdan iborat. Aslida dastur ham algoritmning boshqa bir koʻrinishi boʻlib, u insonning kompyuter bilan muloqotini qulayroq amalga oshirish uchun moʻljallangan.