



**O'ZBEKISTON RESPUBLIKASI OLIY VA O'RTA
MAXSUS TA'LIM VAZIRLIGI
ALISHER NAVOIY NOMIDAGI
SAMARQAND DAVLAT UNIVERSITETI
AXBOROTLASHTIRISH TEXNOLOGIYALARI KAFEDRASI
ALGORITMLAR NAZARIYASI FANIDAN**

REFERAT

Mavzu: Tanlash operatorlari

TALABA: Baratov J

TEKSHIRDI: QARSHIYEV H.

S A M A R Q A N D - 2012

Reja:

Kirish

1-bob. Tarmoqlanuvchi hisoblash algoritmlari turlari

1.1. Shartsiz o'tish operatori

1.2. Shartli o'tish operatori

1.3. Tanlash operatori

2-bob. Tanlash operatoriga doir masalala yechish algoritmlari

2.1.Tarmoqlanuvchi algoritmlarning blok-sxema ko'rinishi

2.2.Tanlash operatorlarigadoir masalalar

Xulosa

Foydalanilgan adabiyotlar

Kirish

Ko'pgina masalalarni echishda ba'zi bir jarayonlar ma'lum shart yoki shartlarning qo'yilishiga nisbatan bajariladi. Bunday jarayonlar tarmoqlanuvchi jarayonlar deb yuritiladi.

Tarmoqlanuvchi hisoblash jarayonlari oddiy va murakkab bo'lishi mumkin. Bu esa jarayondagi tarmoqlar soniga bog'liq. Ma'lum bir tarmoqlanuvchi jarayon tarkibida yana tarmoqlanishlar bo'lishi mumkin. Bunday tarmoqlanishlari bor bo'lgan hisoblash jarayonlari murakkab tarmoqlanuvchi hisoblash jarayonlari deb ataladi.

Bir ketma-ketlikdan ikkinchisiga o'tish maxsus strukturalar yordamida bajariladi. Kerakli harakat (xisoblashlar)ning bajarilishi malum shartlarga bog'liq bo'ladi, masalan, bajarilgan dastur natijalariga yoki kritilayotgan ma'lumotlarga.

Paskal tilida tarmoqlanuvchi jarayonlarni dasturlash uchun shartsiz, shartli o'tish va tanlash operatorlaridan foydalaniladi.

1-bob. Tarmoqlanuvchi hisoblash algoritmlari turlari

1.1. Shartsiz o'tish operatori

Dasturda ba'zi bir hollarda boshqaruvni to'g'ridan-to'g'ri biron-bir operatorga uzatishga, ya'ni dasturning bajarilish ketma-ketligini buzishga to'g'ri keladi. Bu jarayon shartsiz o'tish operatori yordamida bajariladi. Shartsiz o'tish operatorining umumiy ko'rinishi quyidagicha:

GOTO <nishon>;

Bu yerda <nishon > identifikator bo'lib, goto yo'riqnomasidan keyin bajariladigan yo'riqning oldida joylashadi va nishon dasturning nishonlarini tafsiflash bo'limida *label* kalit so'zi yordamida tafsiflangan bo'lishi kerak, bu bo'lim dasturda o'zgaruvchilarni tafsiflash bo'limidan oldin turadi. Dastur matnida nishon, bajarilishi kerak bo'lgan operatoridan oldin va birdaniga undan keyin ikki nuqta qo'yiladi.

Dasturlashtirishga bag'ishlangan adabiyotlarda ayrim hollarda goto yo'riqnomasidan foydalanmaslik tug'risidagi fikrlarni uchratish muumkin, go'yoki goto dasturning chalkashishiga olib kelishi mumkin. Bu fikrlarga ot'liq qo'shilish to'g'ri emas. Nishonlarning ko'p bo'lishi, xaqiqatdan ham chalkashtirishi mumkin, ammo ko'pincha nishondan foydalanish qulay, hamda dasturning ixcham bo'lishini taminlaydi.

Tanlash operatori quyidagicha:

goto<nishon>;

Bu yerda goto xizmatchi so'z (*[nishonga]o't*);<nishon>- belgi.

Belgi Paskal da dasturning bir qancha operatorilarini nomlovchi va unga murojaat qilish imkonini yaratuvchi ixtiyoriy identifikatordir.

Belgi bevosita operator oldiga qo'yiladi va ular ikki nuqta bilan ajratiladi. Operatorni bir vaqtda bir nechta belgilar bilan belgilash mumkin. Dasturda belgilash sharoiti tug'ilgach, u oldindan e'lon qilinishi kerak. Belgilarni e'lon qilinishi, label (*nishon*) xizmatchi so'zi orqali amalga oshiriladi.

Label

loop, bl,b2;

begin
gotobl;

loop:

bl:b2:

.....

gotob2;
end;

Goto operatorining vazifasi boshqarishni mos ravishda belgilangan operatorga uzatishdir.

Belgilardan foydalanishda quyidagi qoidalarga amal qilish lozim

- goto operatori ko'rsatayotgan belgi, albatta e'lon qilinishi shart va dastur tanasida albatta uchrashi kerak.
- dasturda e'lon qilingan belgi albatta uning tanasidagi operatorlarga boshqaruvni uzatishi lozim, dastur tanasi tashqarisiga boshqaruvni uzatish mumkin emas.

1.2. Shartli o'tish operatori

Dasturda boshqaruvni ma'lum shart asosida u yoki bu tarmoqqa uzatish shartli o'tish operatori yordamida amalga oshiriladi. Shartli o'tish operatori ikki xil ko'rinishda ishlatilishi mumkin: to'liq va qisqa.

Paskal tilidagi shart – bu ikki rost (true) yoki yolg'on (false) qiymatlaridan birini qabul qiladigan mantiqiy turdagi (boolean) ifodadir.

Shartli operator berilgan shartni tekshirish imkoniyatini beradi va olingan natijaga qarab u yoki bu harakat bajariladi. Demak shartli operator – xisoblash jarayonini tarmoqlantiruvchi jarayondir.

Shartli o'tish operatorining to'liq ko'rinishi:

Shartlar oddiy va murakkab bo'lishi mumkin. Agar mantiqiy ifodada bitta munosabat amali berilgan bo'lsa, «oddiy shart» ni ifodalaydi.

Kattaliklar orasidagi shartlar HAM, YOKI, EMAS (Paskal tilida AND, OR, NOT) mantiq amallari belgilari orqali bog'lanuvchi bir necha munosabatlardan iborat bo'lsa „murakkab shartlar" deb ataladi.

Shartli operatorining umumiy ko'rinishi

IF <shart>**THEN** <operator1>

ELSE <operator2>;

IF<shart>**THEN**<operator>;

Shart - bu boolean turdagi ifoda. Birinchi holatda, agar ifoda qiymati chin bo'lsa, <operator1>bajariladi, agar yolg'on bo'lsa <operator2>bajariladi. Ikkinchi holatda - agar ifoda natijasi True bo'lsa, <operator>bajariladi, agar False bo'lsa - IF operatoridan keyingi operator bajariladi. IF operatorlari *joylashtirilgan* bo'lishi mumkin .

Matematik yozilishi Algoritmik tilda yozilishi

1) $2 \leq X < 5$

2) $(X \geq 2) \text{ AND } (X < 5)$

AND amalining natijasi uning ikkala argumenti ham rost bo'lsa rost bo'ladi.

OR amalining natijasi rost bo'lishi uchun argumentlardan birining rost bo'lishi etarli.

NOT amalining natijasi argumentning inkor qiymatiga teng, yani argument rost bo'lsa - natija yolg'on, argument yolg'on bo'lsa - natija rost bo'ladi.

Masalan:

$(4 < 5) \text{ AND } (5 < 100)$ - mantiqiy ifoda TRUE (rost),

$(\sin(X) > 1) \text{ AND } (5 \text{ DIV } 2 = 0)$ ifoda FALSE (yolg'on) qiymatga teng.

Shuni ta'kidlab o'tish kerakki, agar mantiqiy ifodalar, biz yuqorida aytganimizdek, mantiqiy amallar yordamida (AND, OR, NOT) murakkab ko'rinishga ega bo'lsa, ular qavslarga olib yoziladi.

Shartli o'tish operatorining ishlatilishini misollarda ko'rib chiqamiz.

1) IF $x > 0$ THEN $y := \text{SQRT}(x)$ ELSE $z := \text{sqr}(x)$;

Operatorning bajarilishi natijasida $x > 0$ bo'lsa, u holda $y := \text{sqr}(x)$ operatori, aks holda $z := \text{sqr}(x)$ operatori bajariladi.

Ayrim algoritmlarda ba'zan shunday xol uchrashi mumkinki, bunda hisoblash jarayonida ayrim amallar ba'zi bir shartlar bajarilgandagina hisoblanadi, aks holda, hech qanday amal bajarilmaydi. Bu holda shartli o'tish operatorini qisqa ko'rinishda ifodalash mumkin.

IF <mantiqiy ifoda> THEN <operator>;

Misol:

IF $X < 1$ THEN $Y := \text{sqr}(X)$;

Shartli o'tishda operator o'rnida, o'z navbatida, yana shartli o'tish operatorining to'la va qisqa ko'rinishlari ishlatilishi mumkin. Masalan:

1) IF B1 THEN IF B2 THEN A;

Bu erda B1, B2 - mantiqiy ifoda, A - operator.

Bu operatorning bajarilishi natijasida B1 mantiqiy ifoda tekshiriladi, agar TRUE qiymat qabul qilsa, B2 mantiqiy ifoda tekshiriladi, u ham rost bo'lsa (TRUE), A operator bajariladi.

Agar B1 yoki B2 mantiqiy ifodalar yolg'on bo'lsa (FALSE), shartli o'tish operatoridan keyingi operator bajariladi.

Agar shartli o'tish operatorida THEN yoki ELSE dan keyin bir necha operator guruhi bajarilsa, ular tarkibiy operator ko'rinishida yozilishi kerak, ya'ni operatorlar qavsi - BEGIN va END lar orasida yoziladi.

1.3. Tanlash operatori

Tanlash operatori, selektor, parametrlar ro'yxati, tanlash o'zgarmaslari ro'yxati.

Juda ko'p tarmoqlanish jarayonlarida tarmoqlanish ikki yoki undan ortiq tarmoqqa ajraladi. Umuman olganda, buni bizga tanish shartli o'tish operatori yordamida amalga oshirish mumkin:

IF B1 THEN A1 ELSE

IF B2 THEN A2 ELSE

.....

.....

IF BK THEN AK ;

Lekin bu hollarda shartli o'tish operatorlarining yozilishi noqulay.

Ko'p hollarda dasturchi uchun shartli operatorning umumiydashgan ko'rinishi - tanlash operatorini ishlatish qulay. Tanlash to'plami uchun 'case' komandasi ishlatiladi.

CASE operatori tarmoqlanish jarayonini berilgan bir necha operatoridan birini tanlash yo'li bilan amalga oshiradi. Tanlash operatorida barcha operatorlar, shu jumladan bajarilishi uchun tanlangan operator ham aniq ravishda keltiriladi (berilgan operatorlar ketma-ketligi chegaralangan).

Tanlash operatori CASE mavjud variantlardan tanlash imkoniyatini beradi. U har biriga tanlash o'zgarmaslari ro'yxati (ro'yxat bitta o'zgarmasdan iborat

bo'lishi mumkin) tegishli *selektor* deb nomlangan ifodadan va *parametrlar ro'yxatidan* iborat.

Formati:

CASE <ifoda-selektor> OF

<ro'yxat1>: <operator1>;

<ro'yxat k2>: <operator2>;

...

<ro'yxat N>: <operatorN>

ELSE <operator>

END;

O'zgarmlar turi doim selektor turiga to'g'ri kelishi kerak. Selektor uchun real va string turlari man etilgan.

CASE operatori quyidagicha ishlaydi. Birinchi navbatda selektor-ifodaqiymati hisoblanadi, keyingi navbatda joriy selektor qiymatiga teng bo'lgan o'zgarmlar qatnashgan operator bajariladi. Agar hech qaysi o'zgarmlar selektorning joriy qiymatiga teng bo'lmasa ELSE so'zidan keyingi operator bajariladi. Agar ELSE so'zi bo'lmasa END so'zidan keyingi operator ishga tushadi, ya'ni CASE chegarasidan keyingi operator.

Selektor butun sonli (-32768..32767 diapazonida bo'lgan) butun, liter yoki foydalanuvchi turiga bog'liq bo'lishi kerak .

O'zgarmlar qiymatlar ro'yxati tasodifiy qiymat yoki diapazondan iborat, ular bir-biridan vergul yordamida ajratiladi. Diapazon chegaralari ikkita biri-biridan ".." belgisi yordamida ajratilgan o'zgarmlar sonlar yordamida yoziladi. O'zgarmlar turi selektor turiga to'g'ri kelishi kerak.

CASE so'zidan keyingi ifodaning qiymti hisoblanadi.

Olingan qiymat, ikki nuqtadan oldingi o'zgarmlar ro'yxatdagi o'zgarmlar bilan ketma- ket solishtiriladi. Bundan keyin quyidagi harakatlardan biri bajariladi:

- Agar ifoda qiymati ro'yxatdagi o'zgarmlarning biriga mos kelsa , u holda bu ro'yxatdagi mos kelgan ketma-ket komandalar bajariladi va operator bajarilishi tugaydi;

- Agar ifoda qiymati ro'yxatdagi o'zgarmaslar biriga mos kelmasa, u holda else so'zidan keying komandalar bajariladi va operator bajarilishdan to'xtaydi;

- Agar bu operatorida else bloki bo'lmasa u holda uning bajarilishi yakunlanadi.

Bajarilishi kerak bo'lgan operator yoki operatorlar ketma-ketligi operator selektorining qiymatiga ko'ra aniqlanadi. Operator selektori sifatida haqiqiy bo'lmagan, skalyar ko'rinishdagi har qanday ifoda yoki o'zgaruvchi ishlatilishi mumkin.

Operatorning ishlashida uning tarkibidagi har bir operator tanlash belgisi deb ataluvchi belgi bilan ta'minlanadi. Bu belgi operatorning bajarilishi uchun zarur bo'lgan selektorning maxsus qiymatini qabul qiladigan selektorning tavsifiga mos konstantadir. Operator bir necha mavjud qiymatlar bilan ishlashi uchun, unda tanlash belgilari ro'yxati keltirilishi kerak.

Tanlash operatoridagi belgili operatorlar oddiy belgiga ham ega bo'lishlari mumkin. Bu holda oldin tanlash belgilari, so'ngra oddiy belgilar yoziladi.

Shuni ham inobatga olish lozimki, tanlash operatoriga faqat CASE xizmatchi so'z orqali kirish mumkin, ya'ni tanlash operatoridan tashqaridagi o'tish operatori orqali bu operatorga murojaat qilish mumkin emas.

Tanlash operatorining bajarilishi uning tarkibidagi operatorlar ketma-ketligidagi bitta operatorning bajarilishiga olib keladi. Shuning uchun ularning biridan biriga GOTO operatori yordamida o'tish xato demakdir.

Shartli o'tish operatorining quyidagi

IF B THEN A1 ELSE A2

ko'rinishi tanlash operatorining quyidagi ko'rinishiga ekvivalentdir:

CASE B OF

TRUE: A1;

FALSE:A2;

END;

qisqa ko'rinishdagi shartli o'tish operatorining IF B THEN A ko'rinishi tanlash operatorining quyidagi ko'rinishga ekvivalentdir:

CASE B OF

TRUE: A;

FALSE

END;

Misol:

CASE T OF

'*', '/': R:=1;

'+', '-': R:=2

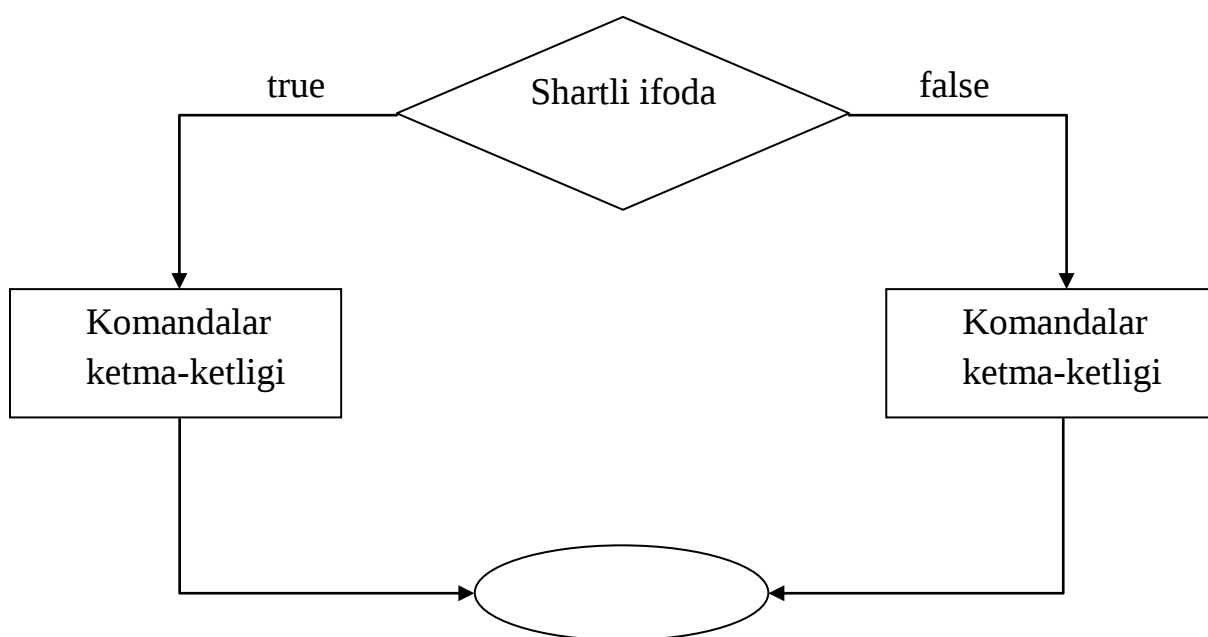
End;

Bu operatorning bajarilishi natijasida, agar T-belgili o'zga ruvchi "+" yoki "-" belgi qiymatlarni qabul qilsa, R o'zgaruvchi 2 qiymatni, agar T o'zgaruvchi "*" yoki "/" belgini qabul qilsa, R o'zgaruvchi 1 qiymatni qabul qiladi.

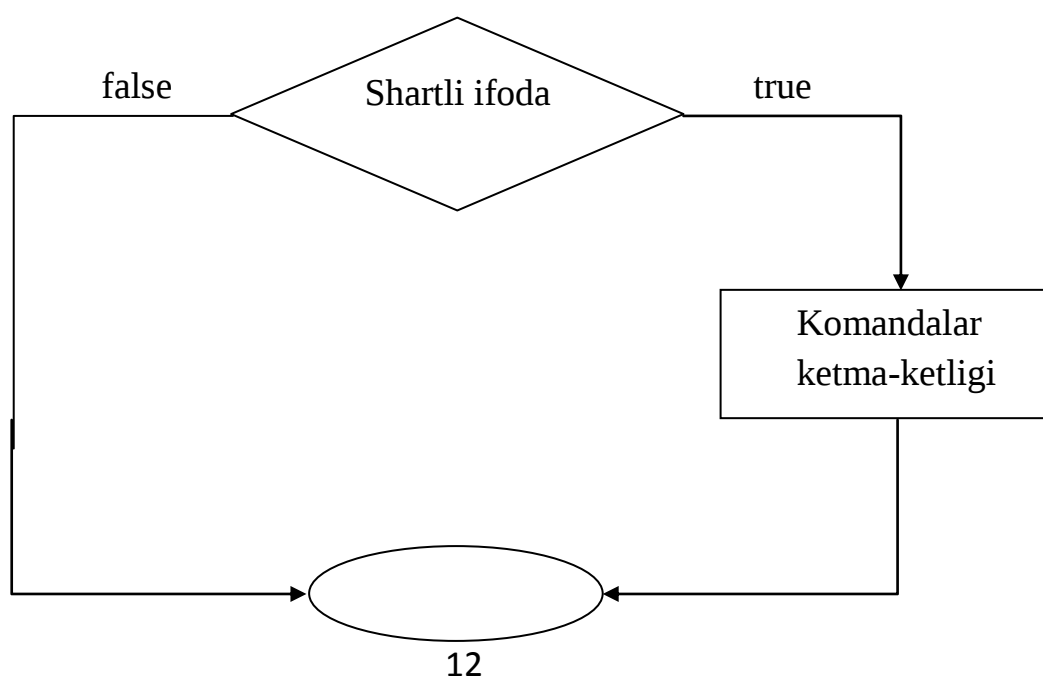
2-bob. Tanlash operatoriga doir masalala yechish algoritmlari

2.1. Dasturga blok-sxema yaratish.

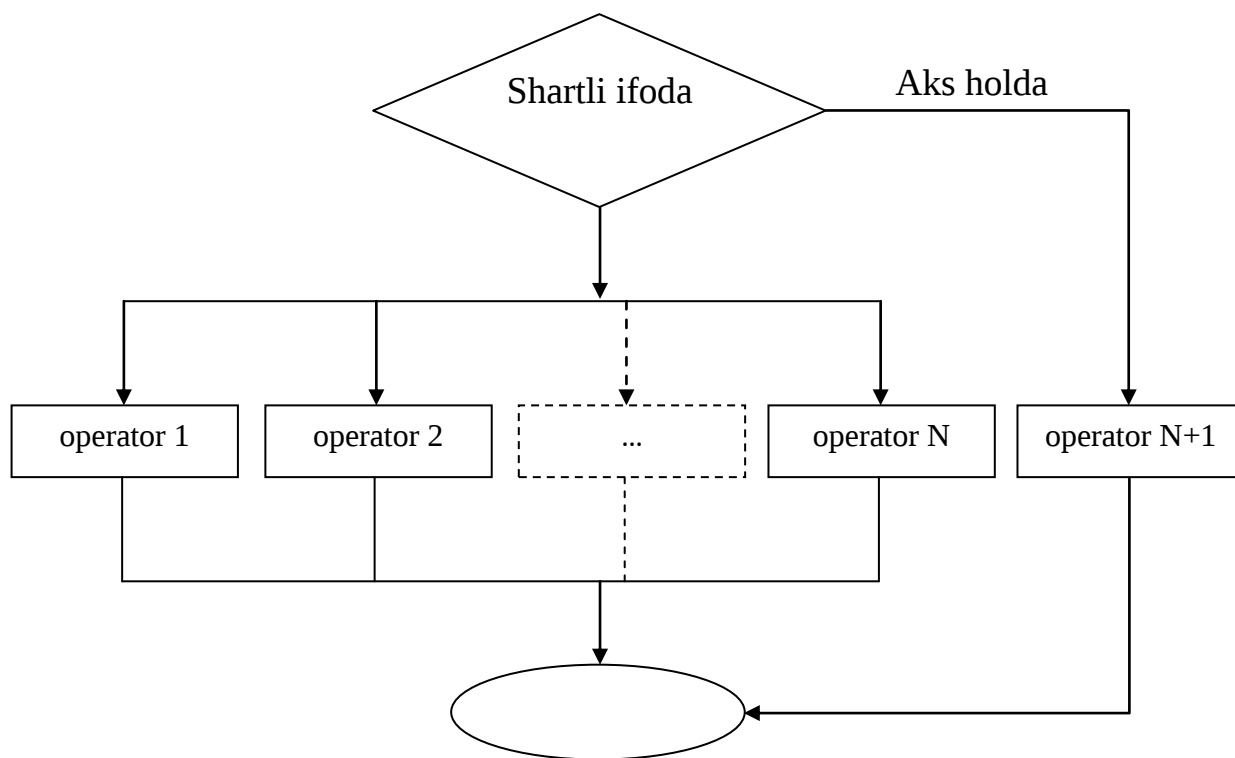
If ... then ... else ... komandasiga mos keluvchi algoritm blok sxema ko'rinishi quyidagicha:



If ... then ... komandasiga mos keluvchi algoritm blok sxema ko'rinishi quyidagicha:



Case operatoriga mos keluvchi algoritm blok sxema ko'rinishi quyidagicha:



2.2. Tanlash operatorlarigadoir masalalar

1-masala: Fasllar 1 dan 4 gacha bo'lgan raqamlarda ifodalangan, hozir qaysi fasl ekanini aniqlovchi dastur tuzing.

Program m2;

Var n:integer;

Begin

 Readln(n);

CASE n of

1:writeln('bahor');

2:writeln('yoz');

3:writeln('kuz');

4:writeln('qish') else

Writeln('bunday fasl yo''q');

End.

Xulosa

Ko'pgina masalalarni yechishning algoritmi ketma ketliklardan iborat bo'lmaydi, yani ularning bajarilishi ketma-ket amalga oshmaydi. Bir ketma ketlikdan ikkinchisiga o'tish maxsus strukturalar yordamida bajarilishi lozim bo'lgan hollarda *tarmoqlanuvchi hisoblash jarayonidan* foydalanish juda samarali bo'ladi. Juda ko'plab shartlar ichidan birini tanlashga to'g'ri kelganda ***tanlash*** operatorlaridan foydalanish juda qulay va samarali.

Foydalanilgan adabyotlar

1. **Gudenko D , Petrochenko D** “DASTURLASHTIRISHDAAN MASALALAR TO’PLAMI”. Tarjimonlar: Qarshiyev A.B, Murodov O’.M., Yusupov P.A. Samarqand-2010 y.
2. Ahatov A.R., Zaripova G.L. va boshq. Axborot texnologiyalari // Uslubiy qo’llanma. – Samarqand: SamDU nashri, 2008 yil .
3. Жуманов И.И. Мингбаев Н.С., Информатика.- Самарқанд: СамДУ нашри, 2002
4. Гуломов С.С. ва бошқалар. Ахборот тизимлари ва технологиялари. Тошкент, 2000 й.
5. Hamrayev .D, Sindorov .M “Algoritm nazaryasi” Toshkent -2004y.