



P. H A N K O

Programmeerimine
M A L G O L -
süsteemis

TALLINN 1969

71-30213
TALLINNA POLÜTEHNILINE INSTITUUT
Arvutusmatemaatika kateeder

PROGRAMMEERIMINE MALGOL-SÜSTEEMIS

Loengukonspekt

Koostanud P. H a n k o

Tallinn
1969

ТАЛЛИНСКИЙ ПОЛИТЕХНИЧЕСКИЙ ИНСТИТУТ
Кафедра вычислительной математики

ПРОГРАММИРОВАНИЕ В СИСТЕМЕ MAlGOL

Конспект лекций

На эстонском языке

Составитель П. Ханко

Tartu Ülikooli Raamatukogu

Vastutav toimetaja A. Korjus

Trükkimisele antud 12.III 69. Paber 50x84/16
Trükipg. 6,0. Tingpg. 5,58. Tiraaz 3000
MB-03011. TPI rotaprint, Tallinn,
Pikk jalg 14. Tell.127
Hind 15 kop.

SISSEJUHATUS.....	4
1. MALGOL-i põhimõisted	
§ 1. Sümbolid.....	6
§ 2. Arvud.....	7
§ 3. Identifikaatorid. Muutujad.....	9
§ 4. Elementaarfunktsioonid.....	11
§ 5. Aritmeetilised avaldised.....	12
§ 6. Võrdlused.....	15
2. Operaatorid	
§ 7. Omistamisoperaator.....	17
§ 8. Operaatorite kirjutamine MALGOL-programmi	18
§ 9. Liitoperaator.....	20
§ 10. Margitud operaator.....	20
§ 11. Suunamis- ja peatamisoperaator.....	21
§ 12. Tingimuslik operaator.....	22
§ 13. Täielik tingimuslik operaator.....	26
3. Tsükliid	
§ 14. Tsükli mõiste.....	30
§ 15. Tingimuslikud operaatorid tsükliites.....	33
§ 16. Tsüklioperaatorid.....	38
4. Massiivid	
§ 17. Indeksitega muutujad.....	46
§ 18. Massiivi kirjeldamise operaator.....	48
§ 19. Massiivide kasutamine programmides.....	49
§ 20. Lüliti.....	54
5. Boole'i algebra	
§ 21. Boole'i avaldised.....	56
§ 22. Boole'i avaldiste kasutamine programmides	59
6. Alamprogrammid ja protseduurid	
§ 23. Alamprogrammi mõiste.....	62
§ 24. Alamprogrammid MALGOL-is.....	63
§ 25. Protseduurid.....	67
§ 26. Funktsioonid.....	73
7. Standardised protseduurid	
§ 27. Standardse protseduuri mõiste.....	75
§ 28. Informatsiooni viimine arvutisse.....	75
§ 29. Informatsiooni väljatoomine arvutist.....	77
§ 30. Valismalu poole pöördumine.....	79
§ 31. Operatsioonid massiividega.....	80
§ 32. Integraali arvutamine.....	85
8. Transleerimine	
§ 33. Transleerimise käik.....	89
§ 34. Programmi silumine.....	90
§ 35. Programmide perforreerimine.....	91

SISSEJUHATUS

Tänapäeval on arvutusmasinate kasutamine seotud mitmesuguste algoritmiliste keelte loomisega. Algoritmiliste keelte loomise eesmärgiks on programmeerijate töö kergendamine. Programmi koostamisel arvuti keeles (s.o. käskude jadana) pidi programmeerija leidma ülesande lahendamise täpse algoritmi, koostama sellele algoritmile vastava loogilise skeemi ja programmi sümbolsetes aadressides, seejärel jaotama arvuti mälu ja kirjutama kogu programmi ümber tegelikkudes aadressides. Nüüd on programmeerimise teooria arenenud automaatse programmeerimiseni, mille puhul suurem osa sellest tööst jäetakse arvuti enda teha. See tähendab, et arvutil enesel lastakse konkreetse ülesande jaoks koostada tööprogramm ja seejärel arvutada selle programmi alusel.

Automaatse programmeerimise korral koostavad programmeerijad programmid kõikides tähtsamate programmeerimisvõtete jaoks, mida tuleb rakendada enamiku ülesannete programmeerimisel, ja viivad need programmid standardsete alamprogrammideni arvuti mälusse. Need arvuti mälus säilitatavad standardsed alamprogrammid moodustavad arvuti translaatori. Translaatori koosseisu kuuluvad näiteks järgmised alamprogrammid: alamprogramm, mis moodustab käskude jada etteantud avaldise väärtuse arvutamiseks; alamprogramm, mis organiseerib tsükleid; alamprogramm, mis modifitseerib käske; alamprogramm, mis jaotab arvuti mälu jne. Konkreetse ülesande programmeerimisel kasutatakse translaatori olemasolevaid osi, mis vähendab tunduvalt programmeerija tööd. Juhul kui arvutile on koostatud translaator, jääb programmeerija ülesandeks arvutusalgoritmi leidmine, sellele algoritmile vastava blokk-skeemi loomine ning algoritmi kirjeldamine sellises algoritmilises keeles, mis on arvutile arusaadav ja rakendab tööle translaatori.

Algoritmiliste keelte loomisega hakati tegelema viie-

kümnendatel aastatel. Viidi läbi rida rahvusvahelisi arvutusmatemaatikute nõupidamisi, millede tulemusena jõuti 1960.a. algoritmilise keele ALGOL-60 (Algoritmic Language) loomiseni. ALGOL-is kirjeldatud arvutusalgoritmi alusel saab töötada iga arvuti, millel on vastav translaator. Loomulikult on igal arvutil omad iseloomulikud jooned. Seetõttu on ka iga konkreetse arvuti translaatoril ja algoritmilisel keelel omad iseärasused. Põhiliselt on nad siiski ALGOL-i variandid. Arvuti MINSK-22 jaoks loodi translaator ja vastav algoritmiline keel 1965.a. Eesti NSV TA Küberneetika Instituudis. Seda algoritmilist keelt nimetatakse MALGOL-iks.

Arvuti MINSK-22 töötab MALGOL-programmi alusel. Programmeerija kirjeldab konkreetse ülesande algoritmi MALGOL-programmina ja laseb selle viia arvuti mälusse. Translaator tõlgib programmi arvuti keelde, s.o. arvuti MINSK-22 käskude jadaks. Kui tööprogramm on valmis, viiakse arvuti mälusse algandmed, lastakse arvutil läbi viia arvutused ja trükitakse tulemused.

1. p e a t ü k k

MALGOL-i PÕHIMÕISTED

§ 1. Sümbolid

MALGOL-programmi koostamisel kasutatakse rahvusvahelise telegraafikoodi sümboleid (tabel 1):

1. suuri ladina tähti;
2. numbreid;
3. spetsiaalmarke.

T a b e l 1

Tähed		Numbrid	Spetsiaalmärgid	
A	O	0	:	∞
B	P	1	(+
C	Q	2)	-
D	R	3	,	=
E	S	4	3	/
F	T	5	⌊	⌋
G	U	6	?	⌊
H	V	7	.	<
I	W	8	,	≡
J	X	9		
K	Y			
L	Z			
M				
N				

Tähti kasutatakse tähistuste ja tekstide koostamiseks. Programmeerija võib MANGOL-programmi kirjutamise juures kasutada ka väikesi tähti, pidades silmas, et perforeerimisel loetakse väikesed tähed identseteks vastavate suurte tähtedega.

Numbreid kasutatakse arvude, tähistuste ja tekstide koostamiseks.

Spetsiaalsetel märkidel ja nende kombinatsioonidel on MALGOL-programmis kindel tähendus, mida selgitame keele kirjelduse vastavates kohtades.

§ 2. Arvud

MALGOL-programmidesse võime kirjutada täisarve, kümnendmurde, arve kümne järgu äranäitamise ja kaheksandarve.

Täisarvu kujutame MALGOL-programmis numbrite jadana, mille ees võib olla märk + või -, näiteks

555
+ 681 075
- 9450789101.

Translaator teisendab ainult need täisarvud, mis kuuluvad diapasooni

$$0 \leq |x| \leq 999\,999\,999,$$

kahendsüsteemi täisarvudeks. Need täisarvud, mis langevad sellest diapasoonest välja, teisendatakse kahendsüsteemi liikuva koma arvudeks.

Kümnendmurdu kujutatakse MALGOL-is numbrite jadana, panes täis- ja murdosa vahele punkti "." (mitte koma!). Kümnendmurru ees võib olla märk + või -.

N ä i d e

Kümnendmurd MALGOL-is	Tõlgendus
6.75	6,75
-572.44	-572,44
0.00	0,00

Arvul kümne järgu äranäitamise on üldiselt kuju

$r \cdot 10^t$

kus r on täisarv või kümnendmurd ja t täisarv. Arv $r \cdot 10^t$ tähendab, et arv r on korrutatud kümne täisastmega, s.o. arvu $r \cdot 10^t$. Järguks t võib olla täisarv ilma märgita või märgiga + või -. Soovi korral võib järgu panna sulgudesse.

N ä i d e s:

Arv kümne järgu äranäitamisega MALGOL-is

Tõlgendus

7.48 $\times 10^8$	$7,43 \cdot 10^8$	ehk 743 000 000
-9.34 $\times 10+10$	$-9,34 \cdot 10^{10}$	ehk -934 000 00000
2 $\times (-4)$	$2 \cdot 10^{-4}$	ehk 0,0002
-1 $\times -7$	$-1 \cdot 10^{-7}$	ehk -0,000 0001

Arvude kirjutamise reeglid lubavad MALGOL-is üht ja sama arvu kirjutada mitmel eri viisil. Näiteks arvu 200 võime MALGOL-is kujutada kui 200, 200 $\cdot 0$, 2 $\times 10^2$, 20.00 $\times 10^1$, 0,0002 $\times 10^6$, 2000 $\times 10^{-1}$ jne.

Kõik kümnendmurrud ja arvud kümne järgu äranäitamisega teisendab translaator kahendsüsteemi liikuva koma arvudeks. Liikuva koma arvude diapasoon arvutis MINSK-22 on

$$10^{-19} \leq |x| \leq 10^{+19}.$$

Arvude kirjutamise juures MALGOL-programmidesse peab silmas pidama, et

- 1) kümnendmurdudes tuleb punkt panna numbrite jada keskele; numbrite jada ei tohi alata ega lõppeda punktiga;
- 2) arv ei tohi koosneda ainult kümne järgust;
- 3) kümne järguks ei tohi olla kümnendmurd.

MALGOL-programmi e i v õ i järelikul kirjutada arve, nagu

3,46
 .51
 675.
 1.4 $\times 10^2.5$
 10^{-7} .

Arv PI'. Arvu $\pi = 3,1415927$ asemel võib MALGOL-programmi kirjutada sõna PI'. Translaator asendab automaatselt sõna PI' vastava numbrilise väärtusega.

Kaheksandarvud. MALGOL-i programmi on sageli vaja kirjutada kaheksandarve. Kaheksandarvudena väljendatakse võtmete numbreid juhtimispuldil, sõnade järjekorranumbreid magnetlindil jne. Kaheksandarvu tunnuseks MALGOL-programmis on

sõna OCTAL', mis kirjutatakse kaheksandarvu ette. Translaator võtab kaheksandarvus arvesse 12 viimast kohta. Juhul kui kaheksandarvus on rohkem kui 12 kohta, jätab translaator esimesed numbrid arvestamata.

N ä i d e s:

OCTAL'12	kujutatakse arvutis kui	0000 0000 0012,
OCTAL'0210000	"	0000 0021 0000,
OCTAL'12345670170774	"	3456 7017 0774.

§ 3. Identifikaatorid. Muutujad

MALGOL-programmides peab programmeerija kõikidele programmi elementidele andma nimetuse. Programmi elementide nimetused valib programmeerija vabalt. Ainsaks kitsendavaks tingimuseks on, et nimetused peavad algama tähega ja koosnema ainult tähtedest ja numbritest. Sellised tähtede ja numbrite jadasid, mis algavad tähega, nimetatakse MALGOL-is identifikaatoriteks. Identifikaatoritega tähistatakse kõiki programmi elemente, s.t. muutujaid, funktsioone, lülitteid, massiive, indekseid, operaatorite märgendeid, alamprogramme ja protseduure. Programmi elementide nimetusteks on soovitav valida selliseid identifikaatoreid, mis peegeldavad nende sisu. Sellega muudame programmi ülevaatlikumaks.

N ä i d e s:

Indifikaator MALGOL-is	Tõlgendus
A	a
Y2	y_2
Z3	z^3
KIIRUS	kiirus
DIAGONAL	diagonaali pikkus
ALPHA	α
I3	kuupjuur
FACT	faktoriaal
MATRIX	maatriks
FA	$f(a)$

Programmeerija peab silmas pidama, et translaator loeb identseteks need identifikaatorid, milledeks koos esimest märki langevad ühte. Järelikult tuleb programmi eri elemendid tähistada nii, et nende nimetustes on kuue esimese sümboli hulgas erinevaid tähti või numbreid. Identifikaatori pikkus ei ole MALGOL-programmides piiratud.

Programmides eraldatakse identifikaatorid üksteisest spetsiaalmärkidega või nende kombinatsioonidega.

Identifikaatorite kirjutamise juures peab silmas pidama, et

1) identifikaator ei tohi alata numbriga;

2) identifikaator ei tohi sisaldada spetsiaalseid märke (ta peab koosnema tähtedest ja numbritest);

3) identifikaatoris peavad kõik sümbolid olema kirjutatud ühele reale, s.o. ta ei tohi sisaldada astmenäitajaid ja indekseid.

Seoses sellega ei sobi identifikaatoriteks näiteks 42X, 1Y2, T1.4, F(A), G^{-1} , B₇ jne.

Identifikaatoriteks ei või valida elementaarfunktsioonide nimetusi, standardsete protseduuride nimetusi ja sõnu PI', TRUE', FALSE', IF', THEN', ELSE', FOR', DO', STEP', UNTIL', WHILE', COMMENT', BEGIN', END', SUBROUTINE', PROCEDURE', SWITCH', ARRAY', KEY', GOTO', STOP', START', FINISH', OCTAL', sest nendel on MALGOL-programmides kindel tähendus. Kõikide ülalmainitud sõnade järel kirjutatakse MALGOL-programmis apostroof'. Neid sõnu nimetatakse MALGOL-i reserveeritud sõnadeks.

Iga MALGOL-programmi identifikaatoriga seab translaator vastavusse ühe mälupeasa. Seega võime MALGOL-programmi identifikaatoreid lugeda vastavate mälupeasade sümboolseteks aadressideks. Mälupeasade sisuks võivad olla kas arvud või käsud. Teatavasti saab arvuti iga mälupeasa sisu programmi täitmise käigus muuta. Seetõttu nimetatakse MALGOL-programmides identifikaatoritega tähistatud suurusi muutujateks. Muutujad erinevad üksteisest identifikaatorite poolest ning neil võib olla mitmesuguseid väärtusi. Väärtusi omistatakse muutujatele omistamisoperaatoriga.

§ 4. Elementaarfunktsioonid

MALGOL-i translaator sisaldab standardsete protseduuridena programme elementaarfunktsioonide väärtuste arvutamiseks. Kui MALGOL-programmi on kirjutatud elementaarfunktsiooni nimetus, millele järgneb sulgudes eeskiri ξ argumenti jooksva väärtuse arvutamiseks, siis translaator arvutab automaatselt välja vastava funktsiooni väärtuse. Eeskirjaks ξ võib olla mistahes aritmeetiline avaldis. Juhul kui eeskirjaks ξ on arv või muutuja, võib sulud ära jätta. Kõik elementaarfunktsioonide nimetused kuuluvad MALGOL-i reserveeritud sõnade hulka. Elementaarfunktsioonide nimetuste juurde pannakse apostroof.

Elementaarfunktsioonideks on MALGOL-is järgmised funktsioonid:

ABS'(ξ) - avaldise ξ absoluutväärtus.

SIGN'(ξ) - avaldise ξ märgi leidmine:

$$\text{sign } x = \begin{cases} + 1, & \text{kui } x > 0 \\ + 0, & \text{kui } x = 0 \\ - 1, & \text{kui } x < 0 \end{cases}$$

SQRT'(ξ) - ruutjuure leidmine avaldisest ξ .

SIN'(ξ) - avaldise ξ siinus; avaldise ξ jooksev väärtus tuleb leida radiaanides.

COS'(ξ) - avaldise ξ koosinus; avaldise ξ jooksev väärtus tuleb leida radiaanides.

TG'(ξ) - avaldise ξ tangens; avaldise ξ jooksev väärtus tuleb leida radiaanides.

ARCSIN'(ξ) - avaldise ξ arkussinuse peaväärtus radiaanides.

ARCTG'(ξ) - avaldise ξ arkustangensi peaväärtus radiaanides.

SH'(ξ) - avaldise ξ hüperboolne siinus.

LN'(ξ) - avaldise ξ naturaallogaritm.

EXP'(ξ) - avaldise ξ eksponentfunktsioon e^{ξ} .

- FIX' (ℓ) - avaldise ℓ väärtuse teisendamine liikuva koma režiimist fikseeritud koma režiimi.
- FLOAT' (ℓ) - avaldise ℓ väärtuse teisendamine fikseeritud koma režiimist liikuva koma režiimi.
- ENTIER' (ℓ) - avaldise ℓ väärtuse täisososa eraldamine.
- FRAC' (ℓ) - avaldise ℓ murdosa eraldamine.
- EXPONENT' (ℓ) - avaldise ℓ väärtuse kahendjärgu eraldamine.

§ 5. Aritmeetilised avaldised

Aritmeetilisi avaldise moodustatakse ühest või mitmest arvust, muutujast ja funktsioonist, neid ühendatakse aritmeetiliste tehete märkide ja ümarsulgude abil.

Arvud, muutujad ja funktsioonid on samuti aritmeetilised avaldised. Neid nimetatakse elementaarseteks aritmeetilisteks avaldisteks.

Aritmeetiliste tehete jaoks kasutatakse MALGOL-is järgmisi tehtemärke:

- + liitmine,
- lahutamine,
- ? korrutamine,
- / jagamine,
- ?? astendamine.

Aritmeetiline avaldis on eeskiri mingi numbrilise väärtuse arvutamiseks. Igal muutujal, mis esineb aritmeetilises avaldises, peab olema mingi numbriline väärtus. Muutujate numbrilised väärtused on kantud perfolindile või magnetlindile ning loetakse sealt vajaduse korral arvuti mälusse. Muutujatele saab väärtusi anda ka omistamisoperaatori abil.

Esitame näiteid aritmeetiliste avaldiste kirjutamise kohta MALGOL-programmis:

Aritmeetiline avaldis MALGOL-is	Tõlgendus
-17.41	-17,41
Z12	z_{12}
R?R - S?S	$r^2 - s^2$
-ARCTG'(X???) ? 0.710 (-7)	$-0,7 \cdot 10^{-7} \operatorname{arctg} x^3$
A / (2?(A - B?X)??2)	$\frac{a}{2(a - bx)^2}$
SIN'(ALPHA?T)/LN'X	$\frac{\sin \alpha t}{\ln x}$
A??3 ? 0.310 (-5) / ABS'(A-B)	$\frac{0.3 \cdot 10^{-5} \cdot a^3}{ a - b }$
(313??(-4) + 9)??(-1) / (-17.2103)	$\frac{1}{(313^{-4} + 9) \cdot (-17200)}$

Aritmeetiliste avaldiste väärtuste arvutamise juures peab arvuti kinni järgmisest tehete järjekorrast:

esimene: funktsioonide väärtuste arvutamine,
teine: ??,
kolmas: /,
neljas: ?,
viies: -,
kuues: +.

See tehete järjekord realiseeritakse arvutis sel teel, et arvuti võrdleb igat tehete aritmeetilises avaldises järgmise tehtega ja sooritab kahest võrreldavast tehtest selle, mis ülaltoodud pingereas asub kõrgemal. Kui võrreldavateks teheteks osutuvad kaks ühesugust tehete, teeb arvuti selle tehte, mis on kirjutatud vasemale.

Tehete järjekorda saab programmeerija reguleerida ümar-sulgude paigutamise aritmeetilisse avaldisse. Avaldis, mis on pandud sulgudesse, arvutatakse välja eraldi, ning tema väärtust kasutatakse edasistes arvutustes.

Aritmeetiliste avaldiste kirjutamise juures on kasulik meeles pidada järgmisi näpunäiteid:

1. Kahte aritmeetilise tehte märki ei kirjutata kõrvalt. Näiteks $A? - B$ ei ole õige kirjutusviis; tuleb kirjutada $-A?B$ ehk $A?(-B)$.

2. Astendaja pannakse sulgudesse. Sulud võib ära jätta ainult siis, kui astendajaks on arv ilma märgita või muutuja ilma märgita.

Näiteks avaldis $(\frac{x}{y})^{2i-1}$ tuleb kirjutada kujul $(X/Y)??$ ($2?I-1$). Juhul kui sulud jäetakse ära, s.o. kirjutatakse $(X/Y)??2?I-1$, interpreteerib arvuti seda avaldist hoopis teise avaldisena, nimelt avaldisena $(\frac{x}{y})^2 \cdot i - 1$.

3. Korrutistes ei tohi ära jätta korrutamise märki, nagu seda sageli tehakse algebras. Arvuti tõlgendab avaldist AB mitte korrutisena $A?B$, vaid muutujana, mille identifikaatoriks on AB .

4. Parema ülevaate saamiseks on soovitatav kasutada täiendavaid sulge. Näiteks on soovitatav kirjutada $(A/B)?C$, mitte $A/B?C$.

5. Kui astendajaks on väike positiivne arv, siis on soovitatav astendamise asemel kirjutada korrutamine. See on tingitud sellest, et astendamine toob kaasa standardse protseduuri poole pöördumise ja nõuab rohkem masinaaega kui korrutamine. Näiteks $X??3$ asemel on parem kirjutada $X?X?X$.

Esitame näiteid aritmeetiliste avaldiste õige ja väära kirjutusviisi kohta MALGOL-programmis.

Matemaatiline kir- jutusviis	Õige kirjutisviis MALGOL-is	Vaar kirjutisviis MALGOL-is
AB	A?B	AB
A(-B)	A?(-B) ehk -A?B	A?-B
-(A+B)	-(A+B) ehk -A-B	-A+B ehk -+A+B
A^{i+2}	A??(I+2)	A??I+2 ($=A^{i+2}$)
$A^{i-3} \cdot B$	A??(I-3) ? B	A??I-3?B ($=A^{i-3}B$)
$\frac{ab}{cd}$	A?B/(C?D) ehk A?B/C/D ehk A/C?B/D	A?B/C?D($=\frac{abd}{c}$)
$(\frac{a+b}{c})^{2,5}$	((A+B)/C)??2.5	(A+B)/C??2.5 ($=\frac{a+b}{c^{2,5}}$)
$(-a)^i$	(-A)??I	-A??I ($=-a^i$)

§ 6. Võrdlused

Arvutusalgoritmides tuleb peale aritmeetiliste tehete teatavasti sooritada ka kontrollimisoperatsioonid. MALGOL-programmides saab kontrollimisoperatsioonid esitada võrdluste kehtivuse kontrollimisena. Iga võrdlus moodustatakse kahest aritmeetilisest avaldisest, nende vahele pannakse mingi võrdlemise sümbol. Seega on võrdluse üldkuju järgmine:

$$\mathcal{E}_1 \mathcal{R} \mathcal{E}_2,$$

kus \mathcal{E}_1 ja \mathcal{E}_2 on aritmeetilised avaldised ja \mathcal{R} on mingi võrdlemise sümbol.

MALGOL-programmides on võimalik kasutada kuut eri võrdlemise sümbolit:

- (: väiksem
- (= väiksem-võrdne
- = võrdne
- =/ mittevõrdne
- :) suurem
- =) suurem-võrdne

Esitame võrdluste näiteid:

Võrdlus MALGOL-is	Tõlgendus
$X = 0$	$x \geq 0$
$ABS^*Y (: EPS$	$ y < \varepsilon$
$X^*X + Y^*Y = R^*R$	$x^2 + y^2 = r^2$
$3 - SQRT^*(X+Y) = 0$	$3 - \sqrt{x+y} = 0$
$X^*Y - 3 (: X^*X - 2 \wedge (-4)$	$xy - 3 < x^2 - 0,0002$

Võrdlus

$\mathcal{E}_1, R, \mathcal{E}_2$

kahe aritmeetilise avaldise \mathcal{E}_1 ja \mathcal{E}_2 vahel võib kehtida või mitte kehtida. Seoses sellega võib võrdlusel olla kaks tõeväärtust: TRUE' või FALSE' (õige või väär). MALGOL-is esitatakse võrdlusega tavaliselt mingi tingimus, mille kehtivust arvuti kontrollib. Arvuti omistab võrdlusele tõeväärtuse TRUE', kui võrdlus kehtib, ning tõeväärtuse FALSE', kui võrdlus ei kehti. Võrdluse $\mathcal{E}_1, R, \mathcal{E}_2$ tõeväärtuse arvutamine toimub järgmisel viisil:

- 1) arvutatakse välja aritmeetiliste avaldiste \mathcal{E}_1 ja \mathcal{E}_2 jooksvad väärtused;
- 2) kui avaldiste \mathcal{E}_1 ja \mathcal{E}_2 väärtused rahuldavad võrdlust, omistatakse võrdlusele tõeväärtus TRUE';
- 3) kui avaldiste \mathcal{E}_1 ja \mathcal{E}_2 väärtused võrdlust ei rahulda, omistatakse võrdlusele tõeväärtus FALSE'.

2. p e a t ü k k

OPERAATORID

§ 7. Omistamisoperaator

Arvutusprotsessis omandavad kasutatavad muutujad järjest uusi väärtusi. Väärtuste omistamine muutujatele toimub MALGOL-programmides kindla kujuga eeskirja alusel, mida nimetatakse omistamisoperaatoriks. Omistamisoperaatori põhiliseks osaks on mingi avaldis, mida vastava muutuja identifikaatorist eraldab omistamise sümbol " := ".

Omistamisoperaatori üldine kuju:

$$h := v,$$

milles v on avaldis, mille väärtus tuleb arvutada, ja h on muutuja, millele see väärtus omistatakse. Sümbolit $h := v$ loetakse "muutujale h omistada avaldise v väärtus".

Omistamisoperaator ei ole võrdus!

Omistamisoperaatori täitmise puhul leiab arvuti avaldise v jooksva väärtuse ja paigutab leitud väärtuse sellesse mälupeassa, mis on reserveeritud muutuja h väärtuste jaoks. Muutuja h endine väärtus asendub omistamisoperaatori täitmise puhul avaldise v jooksva väärtusega. Seega erineb omistamisoperaator oluliselt võrdusest, mis MALGOL-programmides on üheks võimalikuks võrdlusoperatsiooniks. Võrdleme selgituseks omistamisoperaatori $y:=3$ ja võrduse $Y=3$ tähendust MALGOL-is. Omistamisoperaator $y:=3$ tähendab, et arvuti paigutab väärtuse 3 sellesse mälupeassa, mis on reserveeritud muutuja y väärtuste jaoks. Muutuja y olemasolev väärtus asendub omistamisoperaatori $y:=3$ tegevuse tagajärjel väärtusega 3. Võrdluse $y=3$ puhul aga arvuti kontrollib, kas muutuja y jaoks reserveeritud mälupeas on arv 3. Juhul kui selles mälupeas on arv 3, omistatakse võrdusele $y = 3$ tõeväärtus TRUE', vastasel korral FALSE'.

Omistamisoperaatori näiteid:

1. $X := 3$ Muutuja X väärtuste jaoks reserveeritud mälupeassa paigutatakse väärtus 3.
2. $I := 4?K - 6?K1?K2$ Arvutatakse välja avaldise $4?K - 6?K1?K2$ jooksev väärtus ja suunatakse sellesse mälupeassa, mis on reserveeritud I väärtuste jaoks.
3. $K := K-1$ Arvutada $K-1$ väärtus, kasutades pesas K olevat väärtust ning tulemus suunata pesa K , s.o. pesa K sisu väheneb ühe võrra.

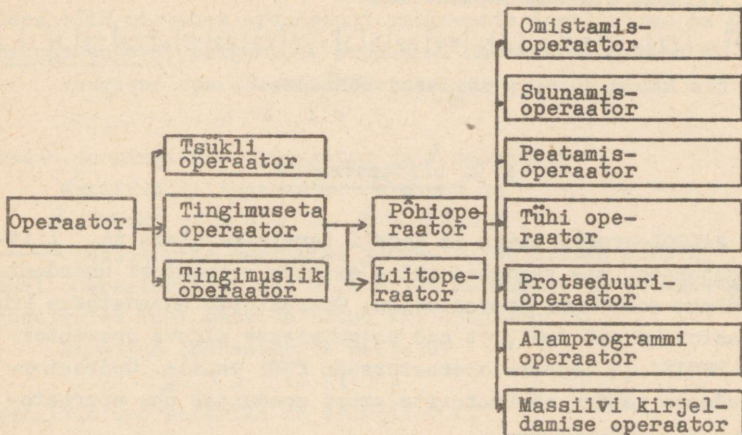
Omistamisoperaatori vasakul pool peab alati olema muutuja, mitte arv või avaldis. Omistamisoperaatoriga saab väärtuse omistada ainult ühele muutujale. Seetõttu e i t o h i MALGOL-programmidesse kirjutada omistamisoperaatoreid

```
6.9 := 0
-A := I??3 + B
X := Y := 0.
```

§ 8. Operaatorite kirjutamine MALGOL-programmi

MALGOL-programmina esitatud lahendusalgoritm kujutab endast operaatorite jada. Operaatorite abil kirjeldatakse ülesande lahenduskaiku. Operaatorite täitmise tulemuseks on nõutavate arvutuste tegemine. Operaatorite üheks põhiligiks on omistamisoperaator. Üldse on automaatse programmeerimise keeles MALGOL olemas 10 eri liiki operaatorit (skeem 1).

Programmi kirjutatakse operaatorid üksteise järel. Sealjuures peab programmeerija silmas pidama järgmist reeglit: programmisis tuleb operaatorid üksteisest eraldada sümboli 3 abil. Sümbol 3 on translaatorile tunnuseks selle kohta, et üks operaator on lõppenud ja alates järgmisest sümbolist algab uus operaator. Sümbol 3 jäetakse alati ära sõna ELSE' ees. Sümboli 3 võib ära jätta sõna END' ees. Kõikidel muudel juhtudel toob sümboli 3 ärajätmine operaatorite vahelt kaasa tõsiseid vigu.



Skeem 1.

Operaatorite jada üldkuju on järgmine:

$$\sim \mathcal{I}_1 \ni \mathcal{I}_2 \ni \sim \ni \mathcal{I}_n \ni \sim$$

kus $\mathcal{I}_1, \mathcal{I}_2, \sim, \mathcal{I}_n$ on mingisugused operaatorid.

MALGOL-programm kirjutatakse tavalisele paberile. Tähtis on, et operaatorid oleksid kirjutatud õiges järjekorras. On soovitatav, et programmeerija paigutaks operaatorid ridadesse sel viisil, et kogu programmist saadaks võimalikult hea ülevaade. Ühte ritta võib paigutada ühe operaatori, mitu operaatorit või osa ühest operaatorist. Ülemineku puhul ühelt realt teisele tuleb silmas pidada järgmisi reegleid:

1. Ka nendel juhtudel, kui rida lõpeb operaatoriga ja järgmine rida algab uue operaatoriga, tuleb operaatorid teineteisest eraldada sümboli \ni abil. Sümboli \ni ärajätmine ei ole lubatud.

2. Kui avaldis tuleb üle kanda ühelt realt teisele, ei tohi sümbolit, millega lõpeb eelmine rida, korrata järgmise rea algul.

3. Ühelt realt teisele ei tohi üle minna identifikaatori keskelt või sellise sümboli keskelt, mis koosneb mitmest märgist.

Näiteks omistamisoperaatorit

$$F := |XI| + (|X1|??|2| - |Y1|??|2|) / (|X2|??|2| + |Y2|??|2|) \ni$$

võib üle kanda joonega näidatud kohtades.

§ 9. Liitoperaator

MALGOL-programmides on sageli tarvis kasutada nn. liitoperaatoreid, mis saadakse mitme erineva operaatori ühendamisel üheks suuremaks operaatoriks. Operaatorid ühendatakse liitoperaatoriks sel teel, et nad paigutatakse algava operaatorsulu BEGIN' ja lõppeva operaatorsulu END' vahele. Operaatorsulgudesse pandud operaatorite grupp moodustab ühe operaatori.

Liitoperaatori üldkuju on järgmine:

$$\text{BEGIN}' \mathcal{J}_1 \ni \mathcal{J}_2 \ni \dots \ni \mathcal{J}_k \text{ END}' \ni .$$

Operaatoreid $\mathcal{J}_1, \mathcal{J}_2, \dots, \mathcal{J}_k$ nimetatakse selle liitoperaatori alamoperaatoreiks. Liitoperaatori alamoperaatoreiks võivad olla ka liitoperaatorid.

Lõppeva operaatorsulu END' eest võib ära jätta sümboli \ni .

N ä i d e: Esitame kolmnurga pindala arvutamise Heroni valemi alusel liitoperaatorina

```
BEGIN' P := (A + B + C)/2 \ni
      S := P?(P-A)?(P-B)?(P-C) \ni
      S := SQRT'S
END' \ni .
```

§ 10. Märgitud operaator

Programmides tekib sageli vajadus viidata mõnele operaatorile. See on võimalik, sest MALGOL-programmides võib iga operaatori varustada märgendiga. Operaatori märgendiks võib olla iga identifikaator, s.o. niisugune tähtede ja numbrite jada, mis algab tähega. Märgend kirjutatakse märgitava operaatori ette ja eraldatakse operaatorist kooloniga. Märgen-

diga operaatorit nimetatakse märgitud operaatoriks. Märgendada võib mistahes operaatori programmis (sealhulgas ka tühja operaatori, paigutades programmi näiteks kirjutise A: 3).

Märgitud operaatori üldkujuks on

$\mathcal{L} \cdot \mathcal{F} \ni$

kus \mathcal{L} on mingi identifikaator ja \mathcal{F} operaator.

Märgitud liitoperaatori üldkuju on

$\mathcal{L} : \text{BEGIN}' \mathcal{F}_1 \ni \mathcal{F}_2 \ni \dots \ni \mathcal{F}_n \text{ END}' \ni$.

Operaatorite märkimise juures tuleb silmas pidada, et ühel operaatoril ei saa olla mitut märgendit ja et sama märgendiga ei tohi märkida mitut eri operaatorit.

Märgitud operaatorite näiteid:

1. A: X := 0 \ni Y := PI' \ni .

Märgitud operaatoriks on siin X:=0 \ni .

2. A: BEGIN' X:=0 \ni Y:=PI' \ni END' \ni .

Märgitud operaatoriks on siin operaatorite grupp X:=0 \ni Y:=PI', mis on pandud operaatorsulgudesse.

3. HERON: BEGIN' P := (A+B+C)/2 \ni

S := P?(P-A)?(P-B)?(P-C) \ni

S := SQRT'S \ni

END' \ni

§ 11. Suunamis- ja peatamisoperaator

Ülesande lahendamisel täidetakse MALGOL-programmis olevaid operaatoreid loomulikus järjekorras, s.t. selles järjekorras, nagu nad on programmi kirjutatud. Operaatorite loomulikus järjekorras täitmine ei võimalda aga kirjeldada harunevaid ja tsüklilisi arvutusalgoritme. Seepärast on MALGOL-programmides ette nähtud võimalus muuta operaatorite täitmise loomulikku järjekorda. See toimub suunamisoperaatori, tingimuslike operaatorite ja tsüklioperaatorite abil.

Suunamisoperaator on üks lihtsamaid operaatoreid MALGOL-is. Tema üldkujuks on

GOTO' $\mathcal{L} \ni$,

kus $\&$ on mingi operaatori märgend programmis. Operaator GOTO' $\&$ näitab, et järgmisena tuleb täita operaator märgendiga $\&$

Suunamisoperaator muudab operaatorite täitmise loomulikku järjekorda, s.t. järjekorda, mis on kindlaks määratud operaatorite üksteisele järgnemisega programmis. Suunamisoperaator näitab, missugune operaator tuleb järgmisena täita.

Suunamisoperaatori näiteid:

GOTO' M3 \exists ,
GOTO' SUMMA \exists

Peatamisoperaator STOP' peatab arvuti, s.t. lõpetab programmi täitmise.

§ 12. Tingimuslik operaator

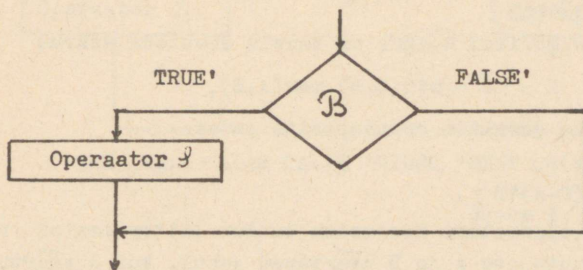
Suunamisoperaator muudab MALGOL-programmides operaatorite täitmise järjekorda, ilma et sealjuures mingi tingimuse kehtivust kontrollitaks. See on nn. tingimuseeta suunamine. MALGOL-programmidena tuleb aga sageli esitada selliseid lahendusalgoritme, milledes operaatorite täitmise järjekord sõltub algtingimustest või programmi täitmise vahepealsetest tulemustest. Seega tekib vajadus suunata operaatorite täitmist teatavatest tingimustest (tingimusega suunamine) olenevalt. Tingimusega suunatakse tingimuslikkude operaatorite abil. Tingimuslikud operaatorid võivad, olenevalt teatavatest tingimustest, vahele jätta või juurde võtta mõne operaatori täidetavate operaatorite jadas.

1. Operaator IF' \sim THEN'. Tingimuslik operaator IF' \sim THEN' on erikujuline liitoperaator, mis algab reserveeritud sõnaga IF'. Edasi kuulub selle operaatori koosseisu mingi võrdlus B ("tingimus"), sõna THEN' ning operaator $\&$. Tingimusliku operaatori IF' \sim THEN' üldkujuks on seega

IF' B \sim THEN' $\&$ \exists

Kui võrdluse B tõeväärtuseks on TRUE', siis täidetakse operaator f ja asutakse järgmise operaatori täitmisele; kui aga võrdluse B tõeväärtuseks on FALSE', jäetakse operaator f vahele ja asutakse kohe järgmise operaatori täitmisele.

Operaatori IF' $B \text{ THEN } f$ täitmist kujutab skeem 2.



Skeem 2.

N ä i t e i d:

1. Kirjutada operaatorite jada lineaarse võrrandisüsteemi

$$\begin{cases} a_1x + b_1y = c_1 \\ a_2x + b_2y = c_2 \end{cases}$$

lahendamiseks. Kui süsteemi determinant on null, peatada arvuti.

Lahendus:

$D := A1?B2 - A2?B1 \ni$
 IF' $D=0 \text{ THEN } \text{GOTO}'M \ni$
 $X := (C1?B2 - C2?B1)/D \ni$
 $Y := (A1?C2 - A2?C1)/D \ni$
 $M: \text{STOP}' \ni$

2. Kirjutada operaatorite jada järgmise arvutuse jaoks:

$$v = \begin{cases} \frac{kx + 0,5}{y - mx}, & \text{kui } y - mx < 0 \\ \frac{a}{x^2}, & \text{kui } y - mx = 0 \\ \frac{kx + 0,5}{y - mx} + \frac{a}{x^2}, & \text{kui } y - mx > 0 \end{cases}$$

Lahendus:

$Y := Y - M?X \ni$

IF' $Y=0$ THEN' BEGIN' $V := A/(X?X) \ni$ GOTO'M END' \ni

IF' $Y(=0)$ THEN' BEGIN' $V := (K?X + 0.5)/Y \ni$ GOTO'M END' \ni

$V := (K?X + 0.5)/Y + A/(X?X) \ni$

M: programm jätkub

3. Kui algoritmi käigus on tarvis arvutada suurus

$$X = C - \min(A, B) \cdot \max(A, B),$$

võib seda teha järgmise operaatorite jadaga:

IF' $A > B$ THEN' BEGIN' $P := A \ni B := A \ni A := P$ END' \ni

$X := (C - A)?B \ni$.

Tingimusliku operaatori koosseisu kuuluv liitoperaator vahetab siin muutujate A ja B väärtused juhul, kui A väärtus on suurem B väärtusest (loomulikult on see lubatav vaid siis, kui A ja B neid väärtusi programmis enam tarvis ei tule).

2. Operaator IF' WHEN THEN' WHEN ELSE'. Operaator IF' \mathcal{B} THEN' \mathcal{S} annab võimaluse täita operaator \mathcal{S} parajasti siis, kui tingimus \mathcal{B} on täidetud. Teiste sõnadega, ta annab võimaluse vahele jätta operaator \mathcal{S} juhul, kui tingimus \mathcal{B} ei ole täidetud. Operaator IF' WHEN THEN' WHEN ELSE' annab selle operaatoriga võrreldes rohkem: ta määrab ka teise operaatori, mis täidetakse parajasti siis, kui antud tingimus ei ole täidetud. Operaatori IF' WHEN THEN' WHEN ELSE' üldkuju on järgmine

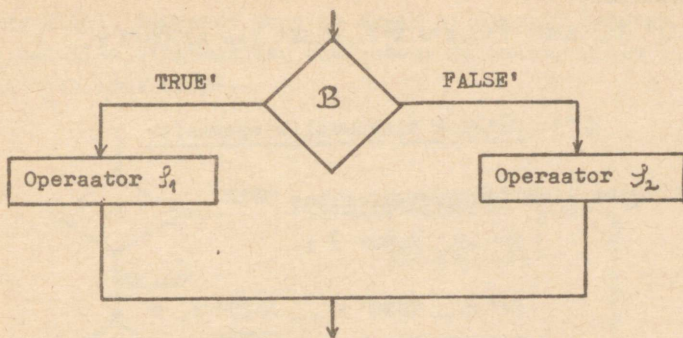
IF' \mathcal{B} THEN' \mathcal{S}_1 ELSE' $\mathcal{S}_2 \ni$

kus \mathcal{B} on võrdlus ja \mathcal{S}_1 ja \mathcal{S}_2 operaatorid. Selle operaatori tegevus on järgmine (skeem 3):

1) Kui võrdluse \mathcal{B} tõeväärtuseks on TRUE', siis täidetakse operaator \mathcal{S}_1 ja operaator \mathcal{S}_2 jäetakse vahele:

2) Kui võrdluse \mathcal{B} tõeväärtuseks on FALSE', siis täidetakse operaator \mathcal{S}_2 ja operaator \mathcal{S}_1 jäetakse vahele.

Seega sõltuvalt võrdluse \mathcal{B} tõeväärtusest täidetakse siin operaatoritest \mathcal{S}_1 ja \mathcal{S}_2 täpselt üks.



Skeem 3.

Sõna ELSE' ette ei kirjutata märki \ni .

N ä i t e i d:

1. Operaator

IF' A > B' THEN' X := 16.9' ELSE' X := 23.1 \ni

teeb arvutuse

$$x = \begin{cases} 16,9, & \text{kui } a > b \\ 23,1, & \text{kui } a \leq b \end{cases}$$

2. Operaator

IF' N \neq -1' THEN' I := Xⁿ⁺¹/(n+1)' ELSE' I := LN'X \ni

teeb arvutuse

$$I = \begin{cases} \frac{x^{n+1}}{n+1}, & \text{kui } n \neq -1 \\ \ln x, & \text{kui } n = -1. \end{cases}$$

M ä r k u s: Tingimuslikkudes operaatorites võib tingimuse kohal kasutada ka aritmeetilisi avaldiseid. Sel juhul arvuti leiab aritmeetilise avaldise väärtuse ning loeb negatiivse tulemuse puhul tingimuse tõevaartuseks TRUE'; mitte-negatiivse tulemuse puhul loeb ta tingimuse tõevaartuseks FALSE'.

N ä i d e.

Kirjutada operaator arvutuse

$$y = \begin{cases} a \sin x, & \text{kui } k < 0 \\ a \cos x, & \text{kui } k \geq 0 \end{cases}$$

jaoks.

Lahendus:

IF' K THEN' Y := A?SIN'X ELSE' Y := A?COS'X 3 .

§ 13. Täielik tingimuslik operaator

Tingimuslikkudes operaatorites

IF' B THEN' J 3

ja

IF' B THEN' J₁ ELSE' J₂ 3

võivad operaatoriteks J, J₁, J₂ olla mistahes operaatorid, sealhulgas ka tingimuslikud operaatorid. Tuleb ainult silmas pidada, et THEN' järel seisev tingimuslik operaator pannakse operaatorsulgudesse. Kui ELSE' järel seisev operaator on tingimuslik operaator, siis saame operaatori

IF' B₁ THEN' J₁ ELSE' IF' B₂ THEN' J₂ 3 J
või operaatori

IF' B₁ THEN' J₁ ELSE' IF' B₂ THEN' J₂ ELSE' J₃ 3 J

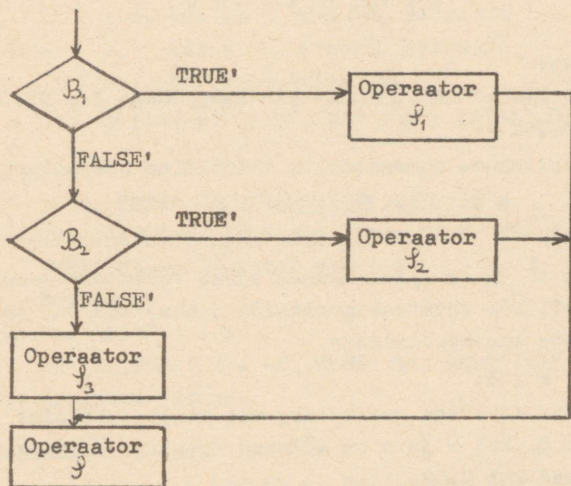
Viimasele ELSE'-le järgnev operaator võib samuti olla tingimuslik operaator. Sel viisil võib üksteisele järgneda kuitahes palju tingimuslikke operaatoreid. Tekkinud konstruktsioone nimetatakse täielikkudeks tingimuslikkudeks operaatoriteks. Täielike tingimuslike operaatoreid täidetakse järgmiste reeglite alusel:

1. Arvuti arvutab järjest võrdluste B₁, B₂, ... tõeväärtusi nii kaua, kuni ta leiab esimese tõeväärtust TRUE' omava võrdluse. Seejärel täidetakse sellele võrdlusele järgnev operaator; ülejäänud tingimuslikud operaatorid jäävad täitmata.

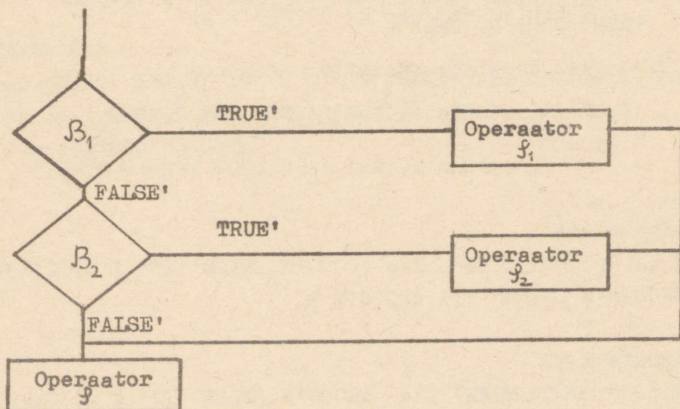
2. Kui kõikide võrdluste tõeväärtusteks on FALSE', siis täidetakse viimasele ELSE'-le järgnev operaator (juhul kui täielik tingimuslik operaator lõpeb sõnaga ELSE').

3. Kui kõikide võrdluste tõeväärtusteks on FALSE' ja täielik tingimuslik operaator ei lõpe sõnaga ELSE', siis antakse juhtimine üle sellele operaatorile, mis järgneb täielikule tingimuslikule operaatorile.

Sellise täieliku tingimusliku operaatori, mis lõpeb sõnaga ELSE', tegevust kujutab skeem 4. Skeem 5 kujutab sellise täieliku tingimusliku operaatori täitmist, mille lõpul ei ole sõna ELSE'.



Skeem 4.



Skeem 5.

N ä i d e s:

Ülesanne: Kirjutada operaator arvutuse

$$y = \begin{cases} a \sin x, & \text{kui } k < 0 \\ b \cos x, & \text{kui } k = 0 \\ c \tan x, & \text{kui } k > 0 \end{cases}$$

jaoks.

Lahendus:

```
IF' K < 0 THEN' Y := A * SIN' X ELSE' IF' K = 0 THEN' Y := B * COS' X ELSE' Y := C * TG' X
```

Tingimuslikkude operaatorite kasutamine / operaatorites

```
IF' B THEN' S
```

```
IF' B THEN' S1 ELSE' S2
```

operaatorite S , S_1 ja S_2 tähenduses annab võimaluse koostada programme selliste arvutusalgoritmide jaoks, mis on seotud liittingimuste kontrollimisega.

N ä i t e i d:

Ülesanne. Koostada operaator, mis annab juhtimise üle operaatorile Q, kui a ja b on mõlemad absoluutväärtuse poolest väiksemad kui 0,01.

Lahendus:

```
IF' ABS' A (< 0.01 THEN' BEGIN' IF' ABS' B (< 0.01 THEN' GOTO' Q END'
```

Ülesanne: Koostada operaator arvutuse

$$y = \begin{cases} \arcsin x, & \text{kui } 0,2000 < x < 0,4999 \\ \arctan x, & \text{kui } x \leq 0,2000 \text{ või } x \geq 0,4999 \end{cases}$$

jaoks.

Lahendus:

```
IF' X (< 0.2000 THEN' Y := ARCTG' X ELSE' IF' X > 0.4999 THEN' Y := ARCSIN' X ELSE' Y := ARCTG' X
```

ehk

```
Y := ARCSIN' X
```

```
IF' X (< 0.2000 THEN' Y := ARCTG' X ELSE' IF' X > 0.4999 THEN' Y := ARCTG' X
```

Ülesanne: Koostada operaatorid, mis arvutavad punkti $M(x, y)$ kauguse poolringjoonest raadiusega a . Punkti M kaugus koordinaatide alguspunktist on r .

$$K = \begin{cases} r-a, & \text{kui } y > 0, r-a \geq 0 \\ a-r, & \text{kui } y > 0, r-a < 0, y+r-a \geq 0 \\ y, & \text{kui } y > 0, r-a < 0, y+r-a < 0 \\ -y, & \text{kui } y \leq 0, |x|-a < 0, \\ \sqrt{(|x|-a)^2 + y^2}, & \text{kui } y \leq 0, |x|-a \geq 0 \end{cases}$$

Lahendus:

```
IF' Y:(=0 THEN' GOTO'T1 ELSE' GOTO'T2
T1: IF' R-A (=) 0 THEN' K:= R-A ELSE' IF' Y+R-A (=) 0
    THEN' K:= A-R ELSE' K:= Y
    GOTO'T3
T2: Z:= ABS'X - A
    IF' Z THEN' K:= -Y ELSE' K:= SQRT'(Z*Z + Y*Y)
T3: programm jätkub.
```

Ülesanne: Arvutada

$$f(x) = \begin{cases} \arccos x, & \text{kui } -1 \leq x \leq 0 \\ \sqrt{1-x} \cdot \frac{1+x}{2}, & \text{kui } 0 < x \leq 1 \\ 4 \arctan(2-x), & \text{kui } 1 < x \leq 2 \end{cases}$$

Lahendus:

```
IF' X(=0 THEN' BEGIN' IF' X (=) -1 THEN' F:= ARCCOS'X END'
ELSE' IF' X(=1 THEN' F:= PI' ? (1+X)/2 ELSE' IF' X(=2
THEN' F:= 4*ARCTG'(2-X)
.
```

3. p e a t ü k k

TSÜKLID

§ 14. Tsükli mõiste

Koostame algoritmi järgmise ülesande lahendamiseks. Lei-
da mürsu liikumise trajektoor, kui mürsk on lastud välja alg-
kiirusega V_0 ajamomendil t_0 .

Olgu teada algkiiruse komponendid x -telje ja y -telje suu-
nas V_{0x} ja V_{0y} ning algmoment $t_0 = 0$. Siis saame igal aja-
momendil t_1 leida mürsu koordinaadid x_1 ja y_1 , kasutades va-
lemeid

$$x_1 = V_{0x} t_1$$

$$y_1 = V_{0y} t_1 - \frac{1}{2} g t_1^2,$$

kus g on raskuskiirendus. Mürsu trajektoori määramiseks tu-
leb leida mürsu asend kindlate ajavahemikkude, näiteks iga
6 s. järel. Seega tuleb meil arvutuste jooksul kasutada kor-
duvalt antud valemid, suurendades t_1 iga kord 6 s. võrra. Ar-
vutused tuleb lõpetada siis, kui y_1 muutub negatiivseks. Prog-
ramm tuleb koostada nii, et arvuti trükiks välja t_1 , x_1 , y_1
väärtused.

Algoritm on järgmine (skeem 6):

1. Võtta $t_0 = 6$.

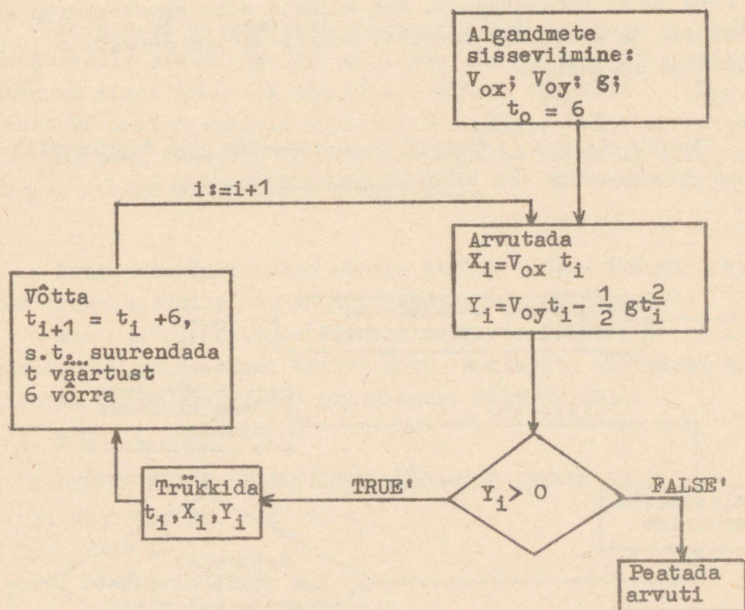
2. Arvutada valemite

$$x_i = v_{ox} t_i$$

$$y_i = v_{oy} t_i - \frac{1}{2} g t_i^2$$

alusel x_i ja y_i .

3. Kui $y_i > 0$, siis trükkida välja t_i , x_i , y_i väärtused, suurendada t_i väärtust 6 võrra ja täita uuesti p. 2; kui $y_i \leq 0$, siis lõpetada arvutus.



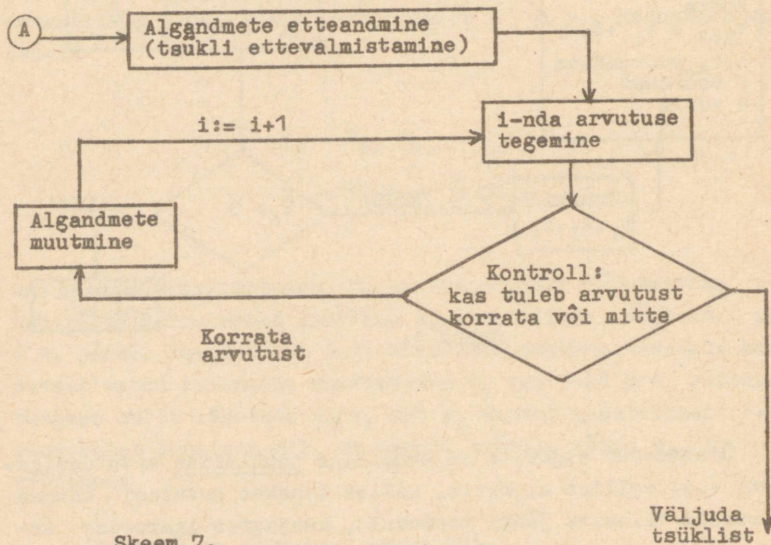
Skeem 6.

Ülaltoodud algoritm on tüüpiline tsükliline arvutusalgoritm, s.o. selline algoritm, milles tehakse arvutusi ühtede ja samade valemite järgi korduvalt, kusjuures igakordse ar-

vutamise puhul kasutatakse uusi lähteandmeid. Niisuguseid arvutusalgortmi korduvaid etappe nimetatakse tsükliteks.

Tsüklilisi arvutusalgortme kirjeldatakse tsükliliste MALGOL-programmidena. Tsüklilistes MALGOL-programmides täidetakse teatud operaatorite rühma korduvalt. Tsüklilistel programmidel on neli põhilist osa. Ta peab sisaldama 1) operaatoreid, mida täidetakse korduvalt; 2) operaatoreid, mis muudavad algandmeid (nn. tsükli parameetreid) enne järjekordset arvutust; 3) operaatoreid, mis määravad kindlaks tsükli kordamiste arvu ja 4) operaatoreid, mis annavad ette parameetrite algväärtused ja alustavad tsükli. Tsükli üldine struktuur on kujutatud skeemil 7.

MALGOL-is saab tsükleid organiseerida kas tingimuslike operaatorite või tsüklioperaatorite abil.



Skeem 7.

§ 15. Tingimuslikud operaatorid tsüklikes

Esitame mõned tsükliliste programmide näited, kasutades tsükli organiseerimiseks tingimuslikke operaatoreid.

Programmide vormistamise juures on kasutatud järgmisi MALGOL-i sümboleid. Programmi algus näidatakse translaatorile ära kirjeldusega START'A, kus A on esimesena täitmisele tuleva operaatori märgend. Kirjeldus START'A paigutatakse tavaliselt programmi lõppu.

Programmi on soovitatav alustada märkusega, mis annab programmile pealkirja. Märkus algab sõnaga COMMENT', millele järgneb mingi tekst ja sümbol \ni . Sõnale COMMENT' järgnev tekst ei tohi sisaldada sümbolit \ni . Märkusi võib kirjutada ka programmi keskele. Translaator märkust vastu ei võta. Märkus ei avalda programmile mingit mõju.

Programm lõpetatakse sõnaga FINISH'. Sõna FINISH' näitab translaatorile, et kogu informatsioon on lõppenud.

Muutujate väärtuste sisselugemine perfolindilt toimub standardse protseduuri READ1' abil. Muutujate väärtused trükitakse välja standardse protseduuri PRINT1' abil.

N ä i t e d.

1. COMMENT' MÜRSU TRAJEKTOORI MÄÄRAMINE (SKEEM 6) \ni

A: READ1' (VOX, VOY, G) \ni

T:=6 \ni

B: X:= VOX?T \ni

Y:= VOY?T - G?T?T/2 \ni

IF' Y:O THEN' BEGIN' PRINT1'(T,X,Y) \ni

T:=T+6 \ni GOTO' B_END' \ni

STOP' \ni START' A \ni FINISH' \ni

2. Koostada programm võrrandisüsteemi

$$\begin{aligned}2x^2 - xy - 5x + 1 &= 0 \\x - y^2 + 1,6 &= 0\end{aligned}$$

lahendite \bar{x} ja \bar{y} leidmiseks veaga $\varepsilon = 10^{-4}$, kui on teada, et punkt (\bar{x}, \bar{y}) kuulub piirkonda

$$R(3,5 \leq \bar{x} \leq 3,6; 2,2 \leq \bar{y} \leq 2,3).$$

Lahendite leidmiseks kasutada järgmist iteratsioonimeetodit (skeem 8):

1. Võtta $x_0 = 3,5$ ja $y_0 = 2,2$.

2. Arvutada

$$x_{i+1} = \sqrt{\frac{x_i(y_i+5)-1}{2}}$$

$$y_{i+1} = \sqrt{x_i + 1,6}.$$

3. Kontrollida tingimusi

$$|x_{i+1} - x_i| \leq \varepsilon, |y_{i+1} - y_i| \leq \varepsilon.$$

Kui mõlemad tingimused on täidetud, siis täita p.5; kui vähemalt üks nendest tingimustest ei ole täidetud, siis täita p.4.

4. Võtta $x_1 = x_{i+1}$ ja $y_1 = y_{i+1}$ ja täita p.2.

5. Lahenditeks võtta $\bar{x} = x_{i+1}$ ja $\bar{y} = y_{i+1}$.

Tähistused algoritmis	x_i	y_i	x_{i+1}, \bar{x}	y_{i+1}, \bar{y}
Tähistused programmis	X1	Y1	X2	Y2

Programm :

COMMENT' VÖRRANDISÜSTEEM ☺

M1: X1:= 3.5 ☺ Y1:= 2.2 ☺ EPS:= 1E-4 (-4) ☺

M2: X2:= SQRT' ((X1 ? (Y1 + 5) - 1)/2) ☺ Y2:= SQRT' (X1+1.6) ☺

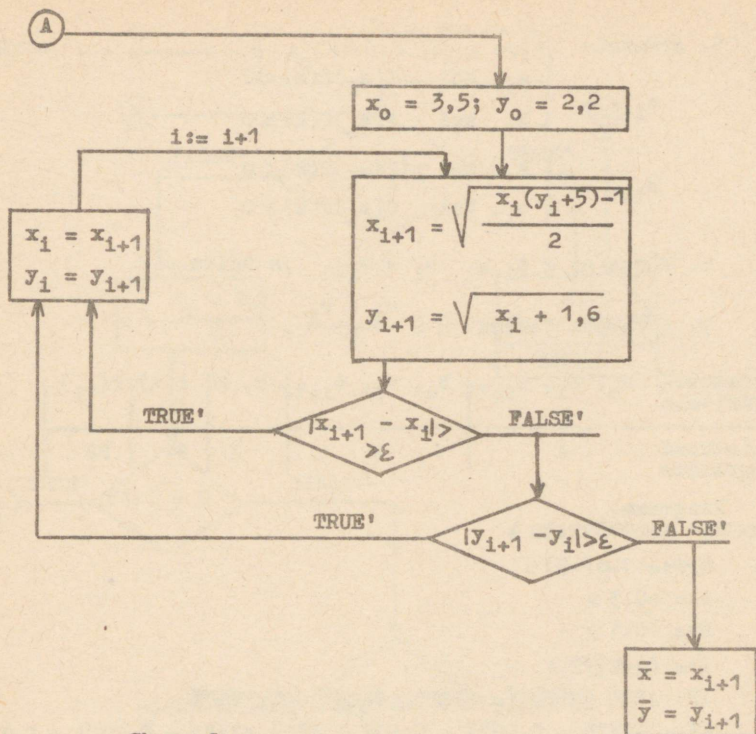
IF' ABS'(X2 - X1) :) EPS THEN'

BEGIN' X1:=X2 ☺ Y1:= Y2 ☺ GOTO'M2 END' ☺

IF' ABS'(Y2 - Y1) :) EPS THEN'

BEGIN' X1:= X2 ☺ Y1:= Y2 ☺ GOTO'M2 END' ☺

PRINT1'(X2,Y2) ☺ STOP' ☺ START'M1 ☺ FINISH' ☺



Skeem 8.

3. Koostada programm võrrandi

$$x^3 - 0,06x + 0,002 = 0$$

lahendi leidmiseks täpsusega $\varepsilon = 10^{-5}$, kui lahend asub vahemikus $[-0,1; -0,3]$.

Algoritm ülesande lahendamiseks (skeem 9):

1. Võtta $a_0 = -0,3$ ja $b_0 = -0,1$.

2. Kontrollida tingimust $|a_1 - b_1| \leq 2\varepsilon$. Kui see tingimus on täidetud, siis täita p.7, vastasel korral p.3.

3. Arvutada $k = \frac{a_1 + b_1}{2}$.

4. Kontrollida tingimust $f(k) = 0$. Kui see tingimus on täidetud, siis täita p.7, vastasel korral p.5.

5. Arvutada

$$a_{i+1} = \begin{cases} a_i, & \text{kui } f(a_i)f(k) < 0 \\ k, & \text{kui } f(a_i)f(k) > 0 \end{cases}$$

$$b_{i+1} = \begin{cases} k, & \text{kui } f(a_i)f(k) < 0 \\ b_i, & \text{kui } f(a_i)f(k) > 0 \end{cases}$$

6. Võtta $a_i = a_{i+1}$, $b_i = b_{i+1}$ ja täita p.2.

7. Arvutada lahend $\bar{x} = \frac{a_i + b_i}{2}$.

Tähistused algoritmis	a_0, a_i, a_{i+1}	b_0, b_i, b_{i+1}	k, \bar{x}	$f(x)$	$f(a_i)$
Tähistused programmis	A	B	X	F1	F2

Programm:

COMMENT' KUUPVÖRRAND 3

M1: EPS:= 10(-5) 3

A:= -0.3 3

B:= -0.1 3

M2: X:= (A+B)/2 3

IF' ABS' (A-B) (= 2?EPS_ THEN' GOTO'M3 3

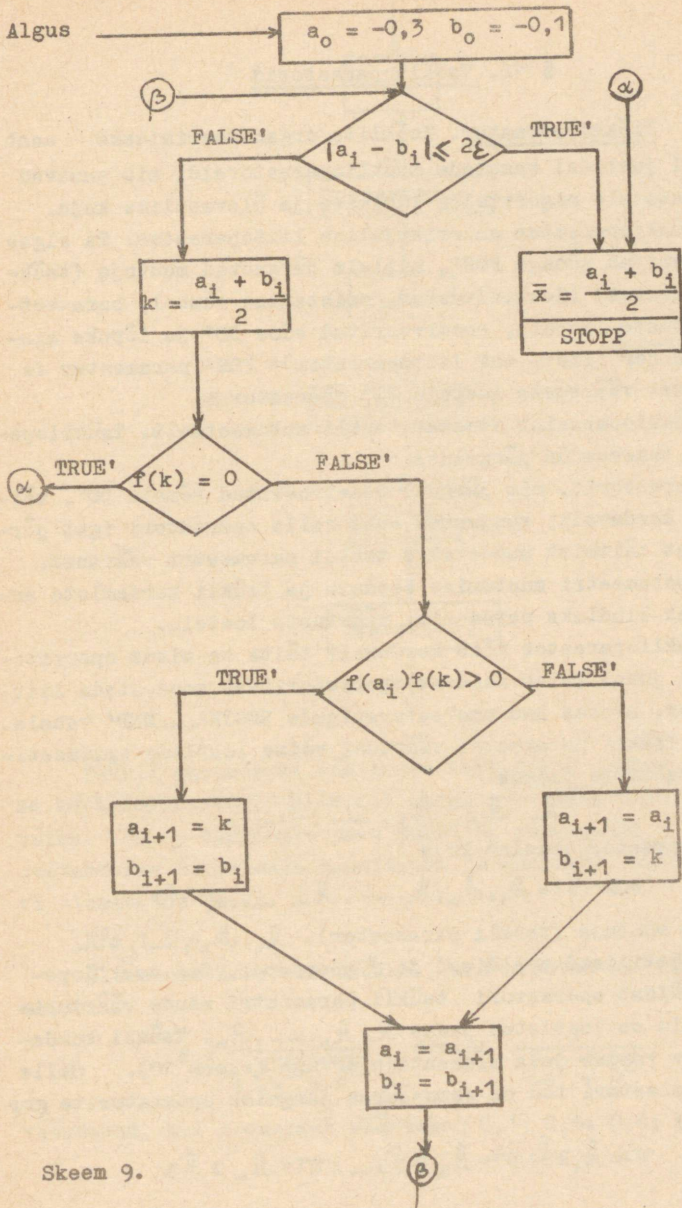
F1:= X?X?X - 0.06?X + 0.002 3 F2:= A?A?A - 0.06?A + 0.002 3

IF' F1= 0_ THEN' GOTO'M3_ ELSE' IF' F1?F2 :) 0_

THEN' BEGIN' A:= X 3 GOTO'M2_ END'

ELSE' BEGIN' B:= X 3 GOTO'M2_ END' 3

M3: PRINT1'(X) 3 STOP' 3 START'M1 3 FINISH' 3 .



Skeem 9.

§ 16. Tsüklioperaatorid

1. Tsüklioperaator. Tsükliite organiseerimiseks saab paljudel juhtudel kasutada tsüklioperaatoreid, mis annavad kirjeldatavale algoritmile lühikese ja ülevaatliku kuju.

Tsüklioperaator on erikujuline liitoperaator. Ta algab reserveeritud sõnaga FOR', millele järgnevad muutuja (tsükli parameetri) identifikaator, omistamise sümbol, parameetri väärtuste loetelu, reserveeritud sõna DO' ja lõpuks mingi operaator (liht- ehk liitoperaator): FOR' parameeter := parameetri väärtuste loetelu DO' operaator \exists .

Tsüklioperaator koostab tsükli automaatselt. Tsüklioperaatori tegevus on järgmine:

Operaatorit, mis järgneb reserveeritud sõnale DO', täidetakse korduvalt, kusjuures enne selle operaatori igat järjekordset täitmist muudetakse tsükli parameetri väärtust. Tsükli parameetri muutumise seaduse ja tsükli kordamiste arvu määrab kindlaks parameetri väärtuste loetelu.

Tsüklioperaator võib korduvalt täita ka mitut operaatorit. Sel juhul tuleb nendest operaatoritest moodustada liitoperaator, pannes nad operaatorsulgude BEGIN' END' vahele.

2. Tsükli parameetri väärtusi võime loetleda aritmeetiliste avaldiste jadana

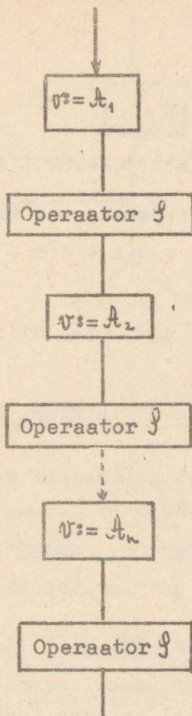
$$A_1, A_2, A_3, \dots, A_n.$$

Tsüklioperaator omandab kuju

$$\text{FOR } V := A_1, A_2, A_3, \dots, A_n \text{ DO } \mathcal{I} \exists,$$

kus V on muutuja (tsükli parameeter), A_1, A_2, \dots, A_n on aritmeetilised avaldised ja \mathcal{I} operaator. See tsüklioperaator täidab operaatori tsükli parameetri nende väärtuste jaoks, mis on loetletud jadas A_1, A_2, \dots, A_n . Tsükli kordamiste arv võrdub jada elementide arvuga (skeem 10). Selle tsüklioperaatori töö on samaväärne järgmise operaatorite grupi tööga:

$$V := A_1 \exists \mathcal{I} \exists V := A_2 \exists \mathcal{I} \exists \dots \exists V := A_n \exists \mathcal{I} \exists.$$



Skeem 10.

Tsükli parameetri väärtuste loetlemist jadana kasutatakse tsüklioperaatoris siis, kui parameetri väärtuste arv on väike. Suure väärtuste arvu puhul on parameetri väärtusi otstarbekas kirjeldada massiivina ja koostada tsükkel massiivi elementide jaoks.

N ä i t e d.

Ülesanne: Koostada tsükkel, mis arvutab ja trükitab funktsiooni

$$y = \frac{1}{a \sqrt{a^2 + b^2}} \arctan \frac{x}{\sqrt{a^2 + b^2}}$$

väärtused, kui x omandab väärtused 0,1; 0,6; 0,8; 1,3.

Lahendus:

```
Z:= SQRT'(A?A + B?B) ⅉ
FOR' X:= 0,1, 0.6, 0.8, 1.3 DO'
BEGIN' Y:= ARCTG'(X/Z)/(A?Z) ⅉ PRINT' (X.Y) END' ⅉ
```

Ülesanne: Koostada tsükkel, mis arvutab
 $y = a(z + b(z + c(z + d)))$.

Lahendus:

```
Y:= 0 ⅉ FOR' N:= D, C, B, A DO' Y:= N?(Z+Y) ⅉ
```

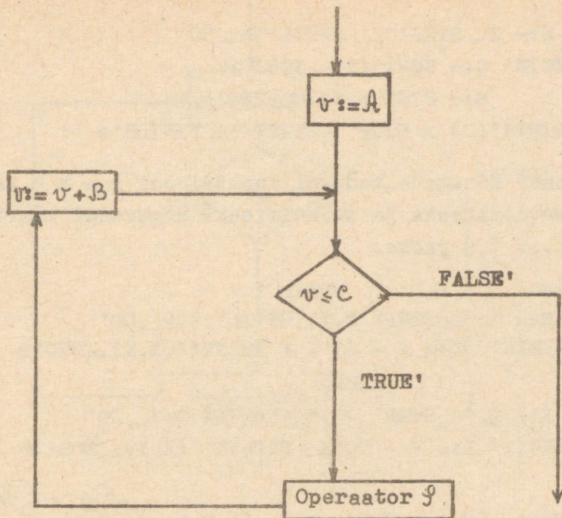
3. STEP' UNTIL' tüüpi loetelu

STEP' UNTIL' tüüpi parameetri väärtuste loetelu puhul omandab tsüklioperaator kuju

```
FOR' v:= A STEP' B UNTIL' C DO' f ⅉ,
```

kus v on muutuja, A , B ja C aritmeetilised avaldised ja f operaator.

Avaldiste A , B ja C väärtusi nimetatakse vastavalt tsükli parameetri algväärtuseks, sammuks ja lõppväärtuseks. Tsüklioperaatori täitmisel antakse tema parameetrile kõigepealt algväärtus ja täidetakse operaator f . Seejärel liidetakse parameetri väärtusele samm ja täidetakse operaator f uuesti. Nüüd liidetakse parameetrile jälle tema samm ja korraldatakse vaadeldud protsessi seni, kuni parameetri väärtus kas 1) positiivse sammu korral osutub lõppväärtusest suuremaks (skeem 11) või 2) negatiivse sammu korral lõppväärtusest väiksemaks. Tuleb arvestada, et kui näiteks positiivse sammu korral on algväärtus lõppväärtusest suurem, jääb sõnale DO' järgnev operaator f üldse täitmata.



Skeem 11.

Seda operaatorit loetakse järgmisel viisil: "parameetri väärtuste jaoks, mis muutuvad alates väärtusest A sammuga B kuni väärtuseni C , täita operaator J ".

STEP' ... UNTIL' tüüpi tsüklioperaatori abil on mugav kirjeldada niisuguseid tsükklilisi arvutusalgoritme, milles tsükli kordamiste arv on ette teada.

N ä i t e d.

Ülesanne. Koostada programm suuruse $\sin(10x)$ väärtuse arvutamiseks, kasutades valemit

$$\sin(ix) = \sin[(i-1)x] \cos x + \sin x \cos[(i-1)x],$$

$$\text{kui } \sin x = \frac{1}{14} \quad (0 \leq x \leq \frac{\pi}{20}).$$

Lahendus:

Tähistused ülesandes:	$\sin(ix)$	$\cos(ix)$	$\sin x$	$\cos x$
Tähistused programmis:	S	C	S1	C1

COMMENT' SIN'10X

A: S1:= 1/14 \ni C1:= SQRT'(1 - S1?S1) \ni S:= S1 \ni

```

FOR' N:= 2 STEP' 1 UNTIL' 10 DO'
  BEGIN' C:= SQRT'(1 - S?S) ⊃
    S:= S?C1 + S1?C END' ⊃
  PRINT1'(S) ⊃ STOP' ⊃ START'A ⊃ FINISH' ⊃

```

Ülesanne. Koostada tsükkel funktsiooni $y = x - \tan x$ väärtuste arvutamiseks ja trükkimiseks argumendi väärtuste 1,1; 1,2; ...; 1,9 jaoks.

Lahendus.

```

FOR' X:= 1.1 STEP' 0.1 UNTIL' 9.9 DO'
  BEGIN' Y:= X - TG'X ⊃ PRINT1'(X,Y) END' ⊃
    ehk
  FOR' X:= 9.9 STEP' -0.1 UNTIL' 1.1 DO'
    BEGIN' Y:= X - TG'X ⊃ PRINT1'(X,Y) END' ⊃

```

3. WHILE' tüüpi tsükli element. Paljudes arvutusalgoritmides tuleb teatud arvutuseeskirja tsüklikiliselt korrata seni, kuni on täidetud mingi etteantud tingimus. Sellistel juhtudel kasutatakse MALGOL-programmides tsüklioperaatorit, mille parameetri väärtuste loetelu algab aritmeetilise avaldisega A , millele järgnevad reserveeritud sõna WHILE' ja võrdlus B . Tsüklioperaator omandab kuju:

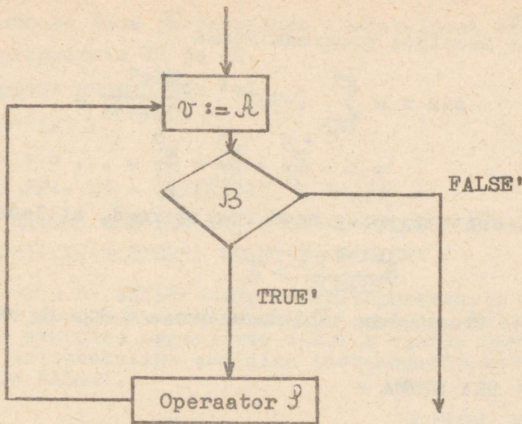
```
FOR' v:= A WHILE' B DO' S ⊃.
```

Siin kujutab omistamisoperaator $v:=A$ parameetri muutumise eeskirja. WHILE'-tüüpi tsüklioperaatori täitmine toimub järgmisel viisil (skeem 12): Tsükli parameetrile omistatakse avaldise A jooksev väärtus. Kui võrdluse B tõeväärtuseks on TRUE', siis täidetakse operaator S ja alustatakse kogu protsessi uuesti. Niipea kui osutub, et võrdluse B tõeväärtuseks on FALSE', jäetakse operaator S täitmata ja väljutakse tsüklist.

Operaatorit

```
FOR' v:= A WHILE' B DO' S ⊃
```

loetakse: "parameetri v jaoks, mis muutub seaduse $v:=A$ alusel, täita operaator S ."



N ä i t e d.

Ülesanne: Koostada programm mürsu trajektoori määramiseks (skeem 6).

Lahendus:

```

COMMENT' MÜRSU TRAJEKTOOR ⊃
M: READ1'(VOX, VOY, G) ⊃
  T:= 0 ⊃ Y:= 0 ⊃
  FOR' T:= T+6 ⊃ WHILE' Y:)0 ⊃ DO'
    BEGIN' X:= VOX?T ⊃
      Y:= VOX?T - G?T?T/2 ⊃
      PRINT1'(T,X,Y) ⊃ END' ⊃
  STOP' ⊃ START' M ⊃ FINISH' ⊃
  
```

Ülesanne: Koostada programm funktsiooni $y = ae^{-bx}$ väärtuste arvutamiseks ja trükkimiseks argumendi väärtuste $x = 0,1; 0,2; 0,3; \dots$ jaoks. Arvutused lõpetada siis, kui e^{-bx} väärtused osutuvad väiksemaks kui 0,0001.

Lahendus:

```

COMMENT' FUNKTSIOONI VÄÄRTUSED ⊃
V: X:= 0 ⊃
  FOR' X:= X + 0.1 ⊃ WHILE' EXP'(-B?X) =) 0.0001 ⊃ DO'
    BEGIN' Y:= A?EXP'(-B?X) ⊃ PRINT'(X,Y) ⊃ END' ⊃
  STOP' ⊃ START' V ⊃ FINISH' ⊃
  
```

Ülesanne: Koostada programm summa

$$\begin{aligned}\sin x &= \sum_{i=0}^{\infty} (-1)^{i+1} \frac{x^{2i+1}}{(2i+1)!} = \\ &= x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \dots\end{aligned}$$

arvutamiseks. Võtta arvesse need rea liikmed, millede puhul

$$\frac{x^{2i+1}}{(2i+1)!} > \varepsilon$$

Lahendus: Programmis tähistame summa S-iga ja rea elemendid M-iga.

```
COMMENT' REA SUMMA ⊃
```

```
A: READ1'(X, EPS) ⊃
```

```
M:= X ⊃ S:= 0 ⊃ i:= 0 ⊃
```

```
FOR' i:= i+2 ⊃ WHILE' M:)EPS ⊃ DO'
```

```
    BEGIN' S:= S+M ⊃
```

```
          M:= (-1)?M?X?X / (i?(i + 1)) ⊃ END' ⊃
```

```
PRINT1'(S) ⊃ STOP' ⊃ START'A ⊃ FINISH' ⊃
```

ehk

```
A: READ1'(X, EPS) ⊃
```

```
M:=X ⊃ S:=X ⊃ i:= 0 ⊃
```

```
FOR' i:= i+1 ⊃ WHILE' M:)EPS ⊃ DO'
```

```
    BEGIN' M:= (-1)?M?X?X / (2?i?(2?i + 1)) ⊃ S:= S+M ⊃ END' ⊃
```

```
PRINT1'(S) ⊃ STOP' ⊃ START'A ⊃ FINISH' ⊃
```

Ülesanne: Koostada programm $\sqrt[3]{A}$ arvutamiseks, rakendades korduvalt iteratsioonivalemit

$$X_{i+1} = X_i + \frac{1}{3} \left(\frac{A}{X_i^2} - X_i \right).$$

Arvutused lõpetada siis, kui

$$|X_{i+1} - X_i| \leq 10^{-7}.$$

Algväärtuseks võtta A.

Lahendus: Kahe järjestikuse iteratsiooniväärtused tähistame programmis X1 ja X2.

COMMENT' KUUPJUUR ☉

B: READ1'(A) ☉

X1:= A ☉

FOR' X2:= X1 + (A/(X1*X1) - X1)/3 ☐

WHILE' ABS'(X2-X1) > 10(-7) ☐ DO' X1:= X2 ☉

PRINT1'(X2) ☉ STOP' ☉ START'B ☉ FINISH' ☉

M ä r k u s: WHILE'-tüüpi tsüklioperaatoris võib tingimust & väljendada ka aritmeetiline avaldis. Sel juhul aritmeetilise avaldise negatiivne väärtus vastab tõeväärtusele TRUE' ja aritmeetilise avaldise mittenegatiivne väärtus tõeväärtusele FALSE'.

5. Parameetri väärtuste loetelu. Parameetri väärtuste loetelu võib koosneda mitmest eri tüüpi loetelu elemendist, mis üksteisest eraldatakse komadega. Näiteks võib tsüklioperaatoreid esitada kujul:

1. FOR' N:= 1, 2, 6 ☐ STEP' 2 ☐ UNTIL' 14, 19, 20 ☐ DO' ☉ ☉.

Selles operaatoris omandab tsükli parameeter järjest väärtusi 1, 2, 6, 8, 10, 12, 14, 19, 20.

2. FOR' A:= 3, 5, 9, A+3 ☐ WHILE' A(<25) ☐ DO' ☉ ☉.

Siin omandab parameeter A väärtused 3, 5, 9, 12, 15, 18, 21, 24.

4. p e a t ü k k

MASSIIVID

§ 17. Indeksitega muutujad

Matemaatilistes ülesannetes on sageli otstarbekas tähistada terve rida muutujaid ühe ja sama sobivate indeksitega varustatud tähega. Näiteks tähistatakse vektori \vec{X} komponente x_1, x_2, \dots, x_n ; maatriksi A elemente

$$a_{11}, a_{12}, a_{13}$$

$$a_{21}, a_{22}, a_{23}$$

jne. MALGOL-programmide korral nimetatakse niisuguseid ühtse tähistusega muutujate komplekse massiivideks. Massiivi koosseisu kuuluvaid muutujaid nimetatakse massiivi elementideks.

Massiivi elemente tähistatakse MALGOL-is indeksitega muutujate abil. Et MALGOL-is tuleb kõik sümbolid kirjutada ühte ritta, siis pole tavaline indeksite kirjutusviis siin võimalik. Selle asemel kirjutatakse indeksid massiivi identifikaatori järel indekssulgudesse $.()$. ja eraldatakse üksteisest komadega. Vektori \vec{X} komponente tähistatakse MALGOL-is seega

$$X.(1)., X.(2)., \dots, X.(n).,$$

ja maatriksi A elemente

$$A.(1,1)., A.(1,2)., A.(1,3).,$$

$$A.(2,1)., A.(2,2)., A.(2,3).$$

jne. Indeksiteks võivad MALGOL-is olla arvud, muutujad, funktsioonid, indeksitega muutujad ja aritmeetilised avaldised.

Kirjutusviis MALGOL-is	Tõlgendus
Y.(7,2).	$\gamma_{7,2}$
Q.(I,K).	$q_{i,k}$
MY.(7,m+n+1,p).	$\mu_{7,m+n+1,p}$
X.(i.(1).).	x_{i_1}
Y.(ABS*N).	$\gamma_{ n }$

Indeksitega muutujat tähistatakse MALGOL-is massiivi identifikaatoriga, millele järgneb indeksisulgudes indeksite loetelu. Indeksite loetelu koosneb aritmeetilistest avaldisest, mis kirjutatakse üksteise järele ja eraldatakse komadega. Igat aritmeetilist avaldist selles loetelus nimetatakse indeksiks.

Indeksitega muutuja üldine kuju MALGOL-is on seega järgmine:

$$M.(v_1, v_2, \dots, v_n).$$

kus M on massiivi identifikaator ja v_1, v_2, \dots, v_n aritmeetilised avaldised.

Indeksite loetelusse kuuluvate aritmeetiliste avaldiste jooksvad väärtused on indeksite väärtusteks. Indeksitel saab olla ainult täisarvulisi väärtusi. Kui indeksit kujutava avaldise jooksvaks väärtuseks on reaalarv, siis ümardatakse see automaatselt lähimaks täisarvuks. Näiteks A.(3.1, 5.8, -5.5) kujutab elementi A.(3,6,-5). Indeksitel võib olla ka negatiivseid väärtusi.

Programmi täitmise ajal võivad indeksite väärtused muududa. Olenevalt indeksite konkreetsetest väärtustest võib indeksitega muutuja kujutada kord üht, kord teist massiivi elementi. Programmeerija peab silmas pidama, et indeksite muutmise jääks massiivi kirjeldusega kindlaksmääratud piiridesse. MALGOL-programmides tohib kasutada ainult neid indeksitega muutujaid, millede indeksite väärtused kuuluvad massiivi kirjeldusega etteantud piiridesse.

MALGOL-is on nõue, et enne massiivi elementide kasutamist programmis peab massiiv olema defineeritud massiivikirjelduse abil. Massiivikirjelduse ülesandeks on anda translaatorile informatsiooni massiivi kohta: siduda massiiviga üks kindel identifikaator, näidata ära massiivi dimensioon (s.o. indeksite arv) ja iga indeksi muutumise piirid.

Edasise käsitluse hõlbustamiseks toome sisse paar mõistet. Tõkete paar koosneb kahest aritmeetilisest avaldisest, mis on teineteisest eraldatud kooloniga. Tõkete paaride loetelu koosneb ühest või mitmest tõkete paarist, mis on üksteisest eraldatud komadega.

Massiivikirjeldus algab reserveeritud sõnaga ARRAY', millele järgnevad massiivi identifikaator ja indeksisulgudes tõkete paaride loetelu. Massiivikirjeldus lõpeb märgiga \exists . n-dimensionaalse massiivi kirjelduse üldine kuju:

$$\text{ARRAY}' M.(a_1:b_1, a_2:b_2, \dots, a_n:b_n). \exists$$

Iga tõkete paar määrab kindlaks vastavas positsioonis seisva indeksi muutumise piirid. Programmis on lubatud kasutada ainult neid indeksitega muutujaid, millede indeksite väärtused kuuluvad massiivi kirjeldusega määratud piiridesse. Üheski tõkete paaris ei tohi alumine tõke ületada ülemist ($a_i \leq b_i$). Indeksil võib olla täisarvulisi väärtusi alumisest tõkkest alates kuni ülemise tõkkeeni sammuga 1. Indeksid võivad olla ka negatiivsed.

Tõkete paaride arv massiivikirjelduses määrab kindlaks massiivi dimensiooni, s.o. näitab ära, mitu indeksit on vastaval indeksitega muutujal.

Massiivikirjeldusest saadud informatsiooni alusel määrab translaator kindlaks elementide arvu massiivis ja vabastab arvuti mälus koha selle massiivi elementide jaoks. Seega on massiivikirjeldus operaator, mis seob kirjeldatava massiiviga ühe kindla identifikaatori ja reserveerib arvuti mälus vajaliku arvu pesi selle massiivi elementide salvestamiseks. Näiteks massiivikirjeldus

ARRAY' A1.(0:2, 1:5). 3

ütleb, et identifikaatoriga A1 on programmis tähistatud kahedimensionaalne massiiv, mille esimesel indeksil saab olla väärtusi 0,1,2 ning teisel väärtusi 1,2,3,4,5 (seega kuulub sellesse massiivi kokku $3 \cdot 5 = 15$ muutujat). Translaator reserveerib arvuti mälus 15 pesa selle massiivi elementide salvestamiseks. Sellele massiivikirjeldusele järgnevates operaatorites võime kasutada muutujaid $A1.(i,j)$, ($i=0,1,2$, $j = 1,2,3,4,5$).

Massiivikirjeldusega saab defineerida ka üle ühe massiivi, kusjuures igale massiivi identifikaatorile peab järgnema rajapaaride loetelu indekssulgudes. Massiivid eraldatakse üksteisest komadega. Massiivikirjeldus lõpeb sümboliga 3. Näiteks massiivikirjeldus

ARRAY' A.(1:10, 6:11), B.(1:100), C.(1:7, 0:5, 0:4). 3

defineerib kahedimensionaalse massiivi A. (kokku 60 muutujat), ühedimensionaalse massiivi B. (kokku 100 muutujat) ja kolmedimensionaalse massiivi C. (kokku 210 muutujat).

Massiivikirjelduste näited.

ARRAY' X.(1:200). 3

ARRAY' Y.(1:M, 0 : 2?N). 3

ARRAY' X.(20:30), Y.(2 : k-1), 0:m, 1:100), Z.(N-1 : N?N). 3

ARRAY' A.(-30:6), B.(-N : N?N, -60 : -4). 3

Väärate massiivikirjelduste näited.

ARRAY' X.(30;20). 3

ARRAY! A.(-1 : -10, 10:20). 3

ARRAY' M,N,U.(1:200). 3

} ülemine tõke ei tohi olla väiksem alumisest

igale massiivi identifikaatorile peab järgnema rajapaaride loetelu.

§ 19. Massiivide kasutamine programmides

Programmide koostamise juures peab rangelt silmas pidama järgmist reeglit:

Massiivikirjeldus peab olema programmi kirjutatud enne kôiki neid operaatoreid, mis kasutavad selle massiivi elemente kujutavaid indeksitega muutujaid.

Kui programmis vihjatakse massiivile, siis tuleb massiivi identifikaatori järele panna punkt.

Massiivi elementide sisselugemine perfolindilt toimub standardse protseduuri READARRAY' abil. Massiivi elementide trükkimine seadeldise БПМ-20 abil toimub standardse protseduuriga PRINTARRAY'.

Programmidesse on sageli kasulik sisse tuua mittevajalike massiivide kustutamine. Sellega vabastatakse mälupead uute andmete jaoks. Massiivide kustutamine mälust toimub standardse protseduuri DELETE' abil. Massiivi A. kustutamisega lõpeb massiivi A. kirjelduse toime, mälupead vabanevad ning identifikaatorit A. võib kasutada uues tähenduses.

Kui mittekustutatud massiivi kirjeldatakse teist korda, jääb kehtima esimene kirjeldus.

Näide massiivide kirjeldamise ja kustutamise kohta.

ARRAY' A.(1:100). ☺

READAR'(A.) ☺

FOR' i:=1, STEP'1, UNTIL'N, DO'

A.(i). := A.(i). ? A.(i). ? A.(i). ☺

PRINTAR'(A.) ☺

DELETE'(A.) ☺

ARRAY' A.(1:10,1:10). ☺

READAR'(A.) ☺

Jne.

Programmide koostamise puhul MALGOL-is on massiividel suur tähtsus. Indeksitega muutujate kasutamine annab võimaluse koostada operaatoreid ühe või mitme massiivi üldliikme kohta ja seejärel, muutes indeksite väärtusi, viia läbi arvutusi paljude elementidega. Väga sageli kasutatakse indeksite väärtuste muutmiseks tsüklioperaatoreid. Tsüklioperaatorite kasutamine massiivide jaoks on MALGOL-programmeerimise põhilisemaid võtteid.

Ülesanne. Arvutada kahekümne arvu ruutude summa.

Selle ülesande lahendamiseks võiksime need arvud tähistada eri identifikaatoritega ja seejärel kirjutada pika omistamisoperaatori ruutude summa arvutamiseks. Selline algoritmi kirjeldamisviis on väga tülikas. Lihtsam on perforeerida kakskümmend arvu massiivina, tähistada nad sümboliga X_i ($i = 1, 2, \dots, 20$) ja programmeerida arvutamine valemi

$$S = \sum_{i=1}^{20} X_i^2$$

alusel. Programm on järgmine:

```
COMMENT' RUUTUDE SUMMA  3
```

```
A:  ARRAY' X.(1:20). 3
```

```
    READAR'(X.) 3 S:=0 3
```

```
    FOR' i:=1 STEP' 1 UNTIL' 20 DO'
```

```
        S:= S + X.(i). * X.(i). 3
```

```
    PRINT1'(S) 3 STOP' 3 START' A 3 FINISH' 3
```

Ülesanne. Koostada programm vektorite $\vec{a} = (a_1, a_2, \dots, a_n)$ ja $\vec{b} = (b_1, b_2, \dots, b_n)$ skalaarkorrutise arvutamiseks. Vektorite komponendid lugeda sisse perfolindilt. Skalaarkorrutis arvutada valemi

$$\vec{a} \cdot \vec{b} = \sum_{i=1}^n a_i b_i$$

alusel.

Lahendus.

```
COMMENT' SKALAARKORRUTIS 3
```

```
H:  READ1'(N) 3
```

```
    ARRAY' A.(1:N)., B.(1:N). 3
```

```
    READAR'(A.,B.) 3
```

```
    S:=0 3
```

```
    FOR' i:= 1 STEP' 1 UNTIL' N DO'
```

```
        S:= S + A.(i). * B.(i). 3
```

```
    PRINT1'(S) 3 STOP' 3 START' H 3 FINISH' 3
```

Ülesanne: Koostada programm, mis arvude a_1, a_2, \dots, a_n jadas leiab üles suurima arvu ja trükitab selle välja. Arv n ja jada elemendid lugeda sisse perfolindilt.

Lahendus.

```
COMMENT' SUURIM ELEMENT ⊃
B: READ1'(N) ⊃ ARRAY' A.(1:N). ⊃
  READAR'(A.) ⊃
  MAX:= A.(1). ⊃
  FOR' i:=2 ⊃ STEP' 1 ⊃ UNTIL' N ⊃ DO'
    IF' A.(i). ⊃ MAX ⊃ THEN' MAX:= A.(i). ⊃
    PRINT1'(MAX) ⊃ STOP' ⊃ START' B ⊃ FINISH' ⊃
```

Ülesanne. Koostada programm polünoomi

$$y = a_0 x^n + a_1 x^{n-1} + a_2 x^{n-2} + \dots + a_{n-1} x + a_n$$

väärtuse arvutamiseks, juhul kui $x = x_0$. Polünoomi väärtuse arvutamiseks kasutada Horneri skeemi

$$y = (\dots(((a_0 x + a_1)x + a_2)x + a_3)x + \dots + a_{n-1})x + a_n.$$

Lahendus.

```
COMMENT' POLÜNOOMI VÄÄRTUS ⊃
H: READ1'(N, X0) ⊃
  ARRAY' A.(0:N). ⊃
  READAR'(A.) ⊃
  Y:= A.(0). ⊃
  FOR' i:= 1 ⊃ STEP' 1 ⊃ UNTIL' N ⊃ DO' Y:= Y?X0 + A.(i). ⊃
  PRINT1'(Y) ⊃ STOP' ⊃ START' H ⊃ FINISH' ⊃
```

Ülesanne. Koostada programm polünoomi

$$y = a_0 x^n + a_1 x^{n-1} + a_2 x^{n-2} + \dots + a_{n-1} x + a_n$$

väärtuste arvutamiseks, kasutades Horneri skeemi, kui x omandab väärtused x_0, x_1, \dots, x_k .

Lahendus.

COMMENT' POLÜNOOMI VÄÄRTUSED ☺

```
H: READ1' (N,K) ☺
ARRAY' A.(1:N)., X.(1:K)., Y.(1:K). ☺
READAR'(A.,X.) ☺
FOR' i:=1┘STEP' 1┘UNTIL' K┘DO'
    BEGIN' Y.(i). := A.(0). ☺
        FOR' j:=1┘STEP' 1┘UNTIL' N┘DO'
            Y.(i). := Y.(i). ? X.(i). + A.(j). ┘
        END' ☺
PRINTAR'(Y.) ☺ STOP'☺ START' H☺ FINISH'☺
```

Ülesanne. Koostada programm ristkülikmaatriksite A ja B korrutise arvutamiseks.

$$\begin{pmatrix} a_{11} & a_{12} & \dots & a_{1q} \\ a_{21} & a_{22} & \dots & a_{2q} \\ \dots & \dots & \dots & \dots \\ a_{m1} & a_{m2} & \dots & a_{mq} \end{pmatrix} \begin{pmatrix} b_{11} & b_{12} & \dots & b_{1n} \\ b_{21} & b_{22} & \dots & b_{2n} \\ \dots & \dots & \dots & \dots \\ b_{q1} & b_{q2} & \dots & b_{qn} \end{pmatrix} = \begin{pmatrix} c_{11} & c_{12} & \dots & c_{1n} \\ c_{21} & c_{22} & \dots & c_{2n} \\ \dots & \dots & \dots & \dots \\ c_{m1} & c_{m2} & \dots & c_{mn} \end{pmatrix}$$

Maatriksi C elemendid arvutatakse valemi

$$C_{ij} = \sum_{k=1}^q a_{ik} b_{kj} \quad \begin{matrix} (i = 1, 2, \dots, m \\ j = 1, 2, \dots, n) \end{matrix}$$

alusel.

Lahendus.

COMMENT' MAATRIKSITE KORRUTAMINE ☺

```
H: READ1' (M,Q,N) ☺
ARRAY' A.(1:M, 1:Q)., B.(1:Q, 1:N)., C.(1:M, 1:N). ☺
READAR'(A.,B.) ☺
FOR' i:=1┘STEP' 1┘UNTIL' M┘DO'
    FOR' j:=1┘STEP' 1┘UNTIL' N┘DO'
        BEGIN' C.(i,j). := 0 ☺
            FOR' k:=1┘STEP' 1┘UNTIL' Q┘DO'
                C.(i,j). := C.(i,j). + A(i,k). ? B.(k,j). ┘
            END' ☺
PRINTAR'(C.) ☺ STOP'☺ START'H☺ FINISH'☺
```

§ 20. Lüliti

Lüliti ehk üldistatud suunamisoperaator valib mitme operaatori hulgast välja ühe operaatori, olenevalt mingi aritmeetilise avaldise väärtusest. Lüliti koosneb kahest osast: lüliti poole pöördumise operaatorist ja lüliti kirjeldusest.

Lüliti poole pöördumise operaator algab reserveeritud sõnaga GOTO^{*}, millele järgnevad mingi identifikaator (lüliti nimetus) ja indekssulgudes aritmeetiline avaldis:

$$\text{GOTO}^* P.(v). \exists,$$

P on identifikaator ja v aritmeetiline avaldis.

Lüliti kirjeldus algab reserveeritud sõnaga SWITCH^{*} millele järgnevad lüliti nimetus, omistamise sümbol ja operaatorite märgendite loetelu:

$$\text{SWITCH}^* P := L_1, L_2, \dots, L_n \exists$$

L_1, L_2, \dots, L_n on operaatorite märgendid.

Lüliti poole pöördumise operaator arvutab välja aritmeetilise avaldise v jooksva väärtuse ja ümardab selle täisarvuks i . Vastavalt i väärtusele määrab lüliti kirjeldus järgmisena täitmisele tuleva operaatori, silmas pidades järgmisi reegleid:

1) kui $i = 0$, siis täidetakse järgmisena operaator, mis programmis järgneb vahenditult lüliti poole pöördumise operaatorile:

2) kui $0 < |i| < n$, siis täidetakse operaator, mille märgend seisab lüliti kirjelduses $|i|$ -ndal kohal, s.t. operaator $L_{|i|}$;

3) kui $|i| > n$, siis jääb järgmisena täitmisele tulev operaator määramata.

N ä i t e d.

Ülesanne. Antud on operaatorite jada, milles operaatorid on varustatud märgenditega L_1, L_2, L_3, L_4, L_5 . Pärast operaatorite L_1, L_2, L_4 täitmist täita operaator märgendiga M . Peale operaatori M täitmist täita vastavalt operaatorid L_2, L_3, L_5 .

Lahendus.

```
SWITCH' P := L2, L3, L5 >
L1: ~ ~ ~ > i:=1 > GOTO' M >
L2: ~ ~ ~ > i:=2 > GOTO' M >
L3: ~ ~ ~ >
L4: ~ ~ ~ > i:=3 > GOTO' M >
L5: ~ ~ ~ > STOP' >
M: ~ ~ ~ > GOTO' P.(i). >
```

Ülesanne: Kirjutada operaatorid järgmise arvutuse jaoks:

kui $n = -2$, siis $y = 6x - c \sin x$

kui $n = -1$, siis $y = 3ax^2 + c \cos x$

kui $n = 0$, siis $y = ax^3 + bx + c \sin x$

kui $n = 1$, siis $y = \frac{ax^4}{4} + \frac{bx^2}{2} + c \cos x$.

Lahendus.

```
GOTO' T.(N+3). >
```

```
SWITCH' P := R,S,T,V >
```

```
R: Y:= 6?X - C?SIN' X > GOTO' U >
```

```
S: Y:= 3?A?X?X + C?COS' X > GOTO' U >
```

```
T: Y:= A?X?X?X + B?X + C?SIN' X > GOTO' U >
```

```
V: Y:= A?X??4/4 + B?X?X/2 + C?COS' X >
```

```
U: programm jätkub.
```

§ 21. Boole'i avaldised

Eespool selgitasime, et kontrollimisoperatsioonid esitatakse MALGOL-programmides võrdluste kehtivuse kontrollimisena. Võrdlust kirjeldasime kui teatavat oletust kahe aritmeetilise avaldise kohta, mis võib kas kehtida või mitte. Vastavalt sellele võib võrdlusel olla kaks tõeväärtust: TRUE' või FALSE' (õige või väär). Seoses sellega võime võrdlused lugeda elementaarseteks Boole'i avaldisteks. Elementaarseteks Boole'i avaldisteks loetakse peale võrdluste veel

1) tõeväärtusi TRUE' ja FALSE';

2) Boole'i muutujaid, s.o. muutujaid, millede väärtusteks on ainult tõeväärtused;

3) Boole'i funktsioone, s.o. funktsioone, mille väärtusteks on ainult tõeväärtused.

Ühendades elementaarseid Boole'i avaldise loogiliste tehete märkide ja ümarsulgude abil, moodustame keerulisemaid Boole'i avaldise. MALGOL-programmides kasutatakse Boole'i avaldise mitmest lihtsast tingimusest koosneva liitmingimuse kirjeldamiseks.

Loogilisteks teheteks nimetatakse järgmisi operatsioone

- . eitamine
- + . loogiline liitmine
- ? . loogiline korrutamine
- / . implikatsioon
- = . ekvivalents.

Eitamine -. (loetakse "mitte") on tehe, mida rakendatakse ainult ühele elementaarsele Boole'i avaldisele. Olgu mingi elementaarne Boole'i avaldis. Kui a on tõene, siis $\neg a$ on väär; kui a on väär, siis $\neg a$ on tõene.

Loogiline liitmine +. (loetakse "või") on tehe, mida rakendatakse vähemalt kahele, üldiselt aga mistahes lõplikule arvule elementaarsetele Boole'i avaldistele. Olgu a ja b elementaarsed Boole'i avaldised. Loogiline summa $a + b$ loetakse tõseks, kui vähemalt üks avaldistest a ja b on tõene; vastasel korral loetakse loogiline summa vääraks.

Loogiline korrutis ?. (loetakse "ja") on samuti tehe, mida rakendatakse kahele, üldiselt aga mistahes lõplikule arvule elementaarsetele Boole'i avaldistele. Loogiline korrutis $a ? b$ loetakse tõseks vaid siis, kui mõlemad avaldised a ja b on tõesed; vastasel korral loetakse loogiline korrutis vääraks.

Implikatsioon /. (loetakse "järeldeb") on tehe, mida rakendatakse kahele elementaarsele Boole'i avaldisele. Implikatsiooni a / b tulemus loetakse vääraks vaid siis, kui a on tõene ja b on väär; kõigil teistel juhtudel loetakse implikatsiooni tulemus tõseks.

Ekvivalents =. (loetakse "on ekvivalentne") on tehe, mida rakendatakse kahele elementaarsele Boole'i avaldisele. Ekvivalents $a = b$ loetakse tõseks siis, kui avaldiste a ja b tõeväärtused on ühesugused; vääraks loetakse ekvivalents siis, kui avaldiste a ja b tõeväärtused on erinevad.

Loogiliste tehete definitsioonid võtame kokku tabelis 2.

T a b e l 2

a	TRUE'	TRUE'	FALSE'	FALSE'
b	TRUE'	FALSE'	TRUE'	FALSE'
$\neg a$	FALSE'	FALSE'	TRUE'	TRUE'
$a + b$	TRUE'	TRUE'	TRUE'	FALSE'
$a ? b$	TRUE'	FALSE'	FALSE'	FALSE'
a / b	TRUE'	FALSE'	TRUE'	TRUE'
$a = b$	TRUE'	FALSE'	FALSE'	TRUE'

Boole'i avaldise tõeväärtuse arvutamise juures peab arvuti kinni järgmisest tehete järjekorrast:

esimene: aritmeetiliste avaldiste väärtuste arvutamine;

teine: võrdluste tõeväärtuste arvutamine;
kolmas: -.
neljas: ?.
viies: +.
kuues: /.
seitsmes: =.

See tehete järjekord realiseeritakse arvutis sel teel, et arvuti võrdleb igat tehet Boole'i avaldises järgmise teheteiga ning teeb kahest võrreldavast tehetest selle, mis ülaltoodud pingereas asub kõrgemal. Kui võrreldavateks teheteks on kaks ühesugust tehet, teeb arvuti selle, mis on kirjutatud vasemale.

Tehete järjekorda saab programmeerija reguleerida ümar-sulgude kasutamisega Boole'i avaldises. Kui avaldis on pandud sulgudesse, arvutatakse ta tõeväärtus välja eraldi ning tulemust kasutatakse edasistes arvutustes.

Boole'i avaldiste näiteid.

Boole'i avaldis MALGOL-is	Tõlgendus
TRUE'	tõene
-(A=0)	$a = 0$
M=N +. M(: 2	Kehtib kas tingimus $m = n$ või $m < 2$;
SQRT'X - Y??3 =/ 0	$x - y^3 \neq 0$
-(ABS'(X-Y):)EPS)	Tingimus $ x-y > \varepsilon$ ei kehti;
X=)0 ? . X(=1	$0 \leq x \leq 1$
(X=)A ? . X(=B) ? . (Y=)C ? . Y(=0)	$\begin{cases} a \leq x \leq b \\ c \leq y \leq b \end{cases}$
X:;)1 / . X:;)0	Tingimusest $x > 1$ järeldeb, et $x > 0$.
-(ABS'(X-Y) :) EPS) =. ABS'(X-Y) (= EPS	Tingimus „ $ x - y > \varepsilon$ ei kehti”, on ekvivalentne tingimusega $ x - y \leq \varepsilon$

Iga Boole'i avaldis on eeskiri tõeväärtuse arvutamiseks. Boole'i avaldise tõeväärtuse arvutamise juures tuleb silmas pidada loogiliste tehete definitsioone ja loogiliste tehete järjekorda.

N ä i t e i d.

1. Olgu $F = \text{FALSE}'$, $R = 2$, $S = 6$, $T = 0$. Siis Boole'i avaldis

$$-.F +. R+S = T$$

omandab tõeväärtuse TRUE' :

$$\begin{array}{r} \underbrace{-.F}_{\text{TRUE}'} +. \quad R+S = T \\ \qquad \qquad \qquad \underbrace{6 = 0}_{\text{FALSE}'} \\ \hline \text{TRUE}' \end{array}$$

2. Olgu $E = \text{TRUE}'$, $F = \text{FALSE}'$, $R = 2$, $T = 0$. Siis Boole'i avaldis

$$E +. F /. R=T$$

omandab tõeväärtuse FALSE' :

$$\begin{array}{r} \underbrace{E +. F}_{\text{TRUE}'} /. \quad \underbrace{R=T}_{\text{FALSE}'} \\ \hline \text{FALSE}' \end{array}$$

§ 22. Boole'i avaldiste kasutamine programmides

Omistamisoperaatori abil saab Boole'i muutujatele anda tõeväärtusi. Sealjuures peab omistamisoperaatori

$$h := v$$

paremaks pooleks olema Boole'i avaldis v .

N ä i t e i d:

1. $A := \text{FALSE}' \ni$
2. $B := X > Y \ni$. Muutuja B omandab tõeväärtuse TRUE' , kui $x > y$, ja FALSE' , kui $x \leq y$.
3. $C := X = 1 \text{ ? } Y < 0 \ni$. Muutuja C omandab väärtuse TRUE' , kui $x = 1$ ja $y < 0$; muudel juhtudel omandab C väärtuse FALSE' .

4. Olgu täidetud operaatorid

E:=TRUE' 3 F:=FALSE' 3 R:=2 3 T:=0 3 .

Siis operaator

P:= E=. F / . R=T 3

omistab muutujale P tõeväärtuse TRUE':

$$P := E = . F / . \underbrace{R = T}_{\text{FALSE'}} \underbrace{}_{\text{TRUE'}} \underbrace{}_{\text{TRUE'}}$$

Tingimuslikkudes operaatorites ja WHILE'-tüüpi tsükli-
operaatorites kirjeldatakse Boole'i avaldiste abil liit-
gimusi.

N ä i t e i d .

Ülesanne. Anda juhtimine üle operaatorile Q, kui a
ja b on mõlemad absoluutväärtuse poolest väiksemad kui 0,01.

Lahendus

IF' ABS'A (: 0.01 ? . ABS'B (: 0.01 _ THEN' GOTO' Q 3

Ülesanne. Kirjutada operaator, mis arvutab

$$y = \begin{cases} \arcsin x, & \text{kui } 0,2000 < x < 0,4999 \\ \arctan x, & \text{kui } x \leq 0,2000 \text{ või } x \geq 0,4999. \end{cases}$$

Lahendus.

IF' X(=0.2000 +. X=)0.4999 _ THEN' Y:= ARCTG'X _ ELSE' Y:= ARCSIN'X 3
ehk

IF' X:)0.2000 ? . X(:0.4999 _ THEN' Y:=ARCSIN'X _ ELSE' Y:=ARCTG'X 3

3. Ülesanne. Kirjutada operaator, mis arvutab

$$f(x) = \begin{cases} \arccos x, & \text{kui } -1 \leq x \leq 0 \\ \frac{1}{2} \frac{1+x}{2}, & \text{kui } 0 < x \leq 1 \\ 4 \arctan (2-x), & \text{kui } 1 < x \leq 2 \end{cases}$$

Lahendus

IF' X=-1 ? . X(=0 _ THEN' F:=ARCCOS'X _ ELSE'

IF' X:)0 ? . X(=1 _ THEN' F:=PI'?(1+X)/2 _

ELSE' IF' X:)1 ? . X(=2 _ THEN' F:=4*ARCTG'(2-X) 3

4. Ülesanne. On antud ruutmaatriks $A = (a_{ij})$. Asendada selle maatriksi peadiagonaali elemendid suurustega

$$h_{ii} = \sqrt{\frac{a_{ik} \cdot a_{il}}{a_{kl}}},$$

kui a_{ik} ja a_{il} on i -nda rea kaks suurimat elementi, mis ei kuulu peadiagonaali (kõik maatriksi elemendid on positiivsed).

Lahendus.

COMMENT' PEADIAGONAALI ELEMENTIDE ASENDAMINE 3

H: ARRAY' A.(1:N, 1:N). 3

READAR'(A.) 3

FOR' i:=1_ STEP' 1_ UNTIL' N_ DO'

BEGIN' N1:=0 3

FOR' j:=1_ STEP' 1_ UNTIL' N_ DO'

IF' j=/i ? . A.(i,j). =) N1_ THEN'

BEGIN' N1:= A.(i,j). 3 k:=j_ END' 3

N2:=0 3

FOR' j:=1_ STEP' 1_ UNTIL' N_ DO'

IF' j=/i ? . j=/k ? . A.(i,j). =) N2_

THEN' BEGIN' N2:= A.(i,j). 3 l:=j_ END' 3

A.(i,i). := SQRT'(N1?N2/A.(k,l).)_

END' 3

PRINTAR'(A.) 3 STOP' 3 START' H 3 FINISH' 3

6. p e a t ü k k

ALAMPROGRAMMID JA PROTSEDUURID

§ 23. Alamprogrammi mõiste

Alamprogrammideks nimetatakse niisuguseid programmilõike, millede poole saab programmi mistahes kohalt ajutiselt pöörduda ja seejärel jätkata arvutusi katkenud kohalt. Alamprogrammidenä vormistatakse arvutamine niisuguste valemite ja arvutuseeskirjade järgi, mida programmeeritavas arvutusalgoritmis tuleb kasutada korduvalt. Põhiprogrammis toimub vajaduse korral pöördumine alamprogrammi poole. Pärast alamprogrammi täitmist jätkub põhiprogrammi täitmine sellest kohast, kust toimus juhtimise üleandmine alamprogrammile. Selline alamprogrammi kasutamine võib toimuda põhiprogrammi erinevatest kohtadest. Alamprogrammi poole pöördumine toimub MALGOL-programmis spetsiaalse operaatori abil. MALGOL-i translaator annab pärast alamprogrammi täitmist automaatselt juhtimise üle sellele operaatorile põhiprogrammis, mis järgneb alamprogrammi poole pöördumisele operaatorile. Iga kord, kui pöördutakse alamprogrammi poole, tuleb alamprogrammile ette anda algandmete jooksivad väärtused. Kui näiteks alamprogramm arvutab mõne funktsiooni väärtuse, tuleb alamprogrammile ette anda argumentide jooksivad väärtused põhiprogrammi vastavas kohas.

Alamprogrammi kasutamist illustreerib skeem 13. Selles skeemis ühendaja A näitab, et vastavas kohas toimub pöördumine põhiprogrammi poole. Ühendaja B on selles skeemis "muutuva" ühendaja: ta näitab, et alamprogramm annab juhtimise üle põhiprogrammi kohale B_1 või B_2 olenevalt sellest, kas alamprogrammi kasutatakse esimest või teist korda.

Mingi arvutuseeskirja esitamine iseseisva alamprogrammina võib osutuda kasulikuks isegi siis, kui seda vaadeldavas põhiprogrammis kasutatakse ühekorvalt. Nimelt saab ulatusli-

kuma ülesande lahendamise peaaegu alati taandada terve rea lihtsamate ülesannete lahendamisele. Üsna sageli on nende lihtsamate ülesannete lahendusalgorithmid juba varem programmeeritud ja neid säilitatakse arvutuskeskuse programmide kogus. MALGOL-programmeerimise puhul säilitatakse selliseid valmis programmilõike põhiprogrammist sõltumatute alamprogrammidenä ehk protseduuridena. Protseduure võib lülitada mistahes koostatava põhiprogrammi koosseisu. Loomulikult tagab selline valmis programmilõikude sisselülitamine koostatavasse programmi suure töö kokkuhoiu.

Asjaolu, et üksikuid programmiosi on võimalik põhiprogrammi lülitada alamprogrammidenä, on tähtis veel ühelt seisukohalt. Nimelt lubab see ulatuslikuma ülesande lahendamiseks vajaliku programmi koostamisele rakendada mitut programmeerijat - erisuguseid alamprogramme võivad koostada eri isikud.

§ 24. Alamprogrammid MALGOL-is

Reserveeritud sõnaga SUBROUTINE^{*} algav kirjeldus defineerib alamprogrammina operaatori \mathcal{J} . Selle kirjelduse üldkuju on

$$\text{SUBROUTINE} \mathcal{A} \ni \mathcal{J} \ni ,$$

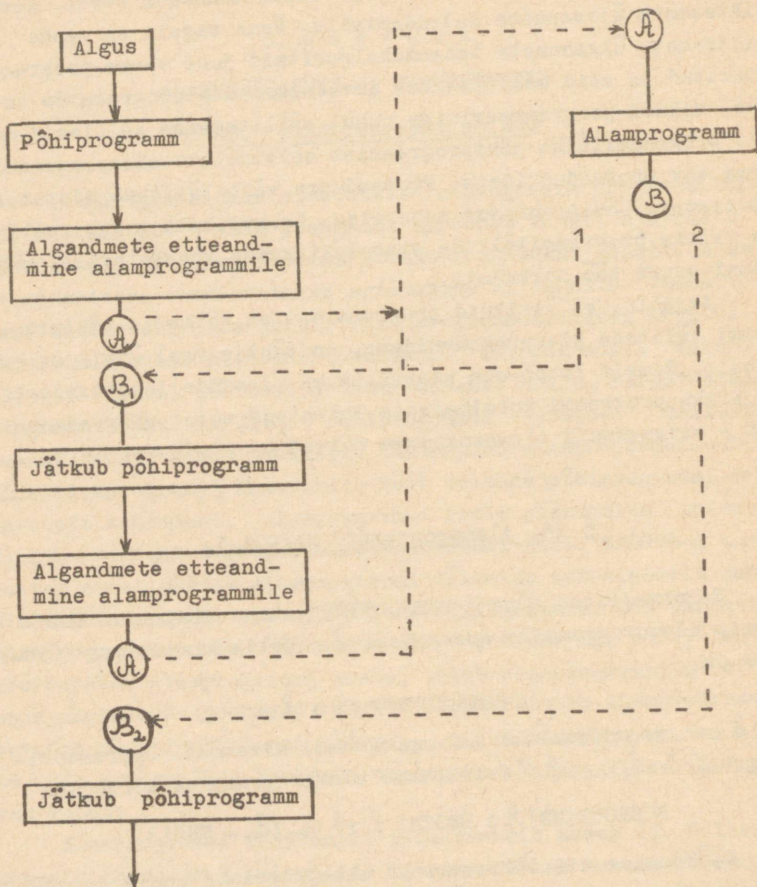
kus \mathcal{A} on identifikaator (alamprogrammi nimetus). Operaatorite grupi $\mathcal{J}_1 \ni \mathcal{J}_2 \ni \dots \ni \mathcal{J}_n$ defineerib alamprogrammina kirjeldus

$$\text{SUBROUTINE} \mathcal{A} \ni \text{BEGIN} \mathcal{J}_1 \ni \mathcal{J}_2 \ni \dots \ni \mathcal{J}_n \text{ END} \ni$$

Pöördumine nii defineeritud alamprogrammi poole toimub operaatoriga

$$\mathcal{A} \ni ,$$

s.t. alamprogrammi nimetusega. Seega on MALGOL-is alamprogrammi nimetus alamprogrammi poole pöördumise operaatoriks. Tuleb silmas pidada, et alamprogrammi nimetus ei ole märgend. Järelikult ei või tema ette kirjutada suunamisoperaatorit või lülitit. Suunamisoperaator ja lülitit, mis vihjavad alamprogrammi nimetusele, jäävad täitmata.



Skeem 13.

Alamprogrammi poole pöördumise operaatorit

⌈ 3

võime kasutada põhiprogrammi mistahes kohas. Alamprogrammi poole pöördumise operaator annab tulemusena juhtimise üle alamprogrammi kirjelduses esinevale operaatorile

BEGIN' ⌈₁ ⌈₂ ... ⌈_n — END' 3.

Selle liitoperaatori täitmist alustatakse tema esimesest operaatorist ⌈₁, s.o. operaatorist, mis vahetult järgneb esimesele operaatorsulule BEGIN', ja jätkatakse tavaliste reeglitega määratud järjekorras seni, kuni on täidetud viimane operaator ⌈_n, s.o. operaator, millele vahetult järgneb viimane operaatorsulg END'. Selle operaatori täitmine lõpetab alamprogrammi töö ja annab juhtimise üle põhiprogrammi sellele operaatorile, mis vahetult järgneb alamprogrammi poole pöördumise operaatorile. Kui alamprogrammi täitmine lõpeb alamprogrammi keskel, tuleb viimaseks operaatoriks kirjutada märgitud tühi operaator ja anda juhtimine üle sellele tühjale operaatorile. Liitoperaator omandab siis järgmise kuju

BEGIN' ⌈₁ ⌈ ... ⌈_i ⌈ GOTO' ⌈_j ⌈ ... ⌈_k ⌈ END' 3,

kus ⌈ on tühja operaatori märgend ja ⌈_i on operaator, millega lõpeb alamprogrammi täitmine.

Alamprogrammide kasutamisel tuleb silmas pidada, et SUBROUTINE'-ga kirjeldatud alamprogramm on põhiprogrammist sõltuv. See tähendab, et kõik identifikaatorid, mida kasutatakse põhiprogrammis, säilitavad oma tähenduse SUBROUTINE'-ga kirjeldatud alamprogrammis.

Programmeerimisel on kirjeldust SUBROUTINE' kohane kasutada ühe ja sama algoritmi erinevate osade esitamisel alamprogrammidenä. Nende kasutamine võimaldab ökonoomsemalt jaotada arvuti mälu ja lihtsustab MALGOL-programmide koostamist. Kirjeldusega SUBROUTINE' defineeritud alamprogrammid ei sobi aga kasutamiseks erinevates põhiprogrammides. Põhjuseks on alamprogrammide sõltuvus põhiprogrammist: alamprogrammide tähistused peavad alati olema kooskõlas vastava põhiprogrammi tähistustega.

N ä i t e i d.

Ülesanne. Koostada programm suuruse

$$C = \frac{n!}{a!(n-a)!}$$

arvutamiseks, kasutades alamprogrammi faktoriaali arvutamiseks.

Lahendus.

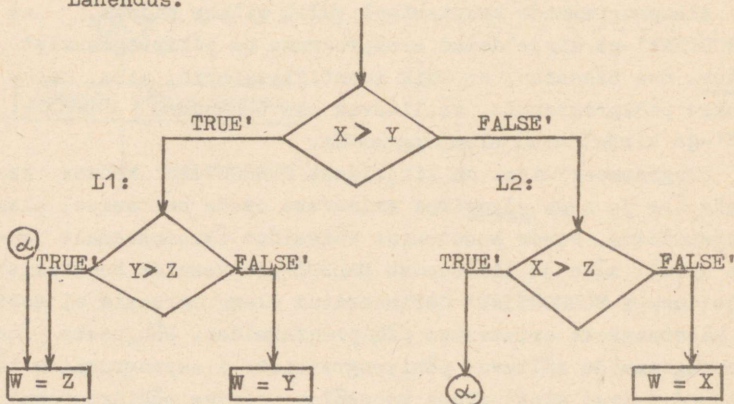
```
SUBROUTINE' FACTORIAL 3
BEGIN' Y:=1 3
  FOR' Z:=2 _STEP' 1 _UNTIL' X _DO' Y:= Y*Z _
END' 3
COMMENT' PÕHIPROGRAMM 3
H:  READ'1 (N,A) 3
    X:=N 3 FACTORIAL 3 C:=Y 3
    X:=A 3 FACTORIAL 3 C:= C/Y 3
    X:= N-A 3 FACTORIAL 3 C:= C/Y 3
    PRINT' (C) 3 STOP' 3 START' H 3 FINISH' 3
```

Ülesanne. Koostada programm suuruse

$$M = \max \{ \min(A, B, C), \min(D, E, F) \}$$

leidmiseks, vormistades alamprogrammina kolmest arvust vähima leidmise (skeem 14).

Lahendus.



Skeem 14

SUBROUTINE' MIN ⊃

```
BEGIN' IF' X:)Y ⊃ THEN' GOTO' L1 ⊃ ELSE' GOTO' L2 ⊃  
L1: IF' Y:)Z ⊃ THEN' GOTO' L3 ⊃ ELSE' GOTO' L4 ⊃  
L2: IF' X:)Z ⊃ THEN' GOTO' L3 ⊃ ELSE' GOTO' L5 ⊃  
L3: W:=Z ⊃ GOTO' L6 ⊃  
L4: W:=Y ⊃ GOTO' L6 ⊃  
L5: W:=X ⊃  
L6: ⊃ END' ⊃
```

COMMENT' PÕHIPROGRAMM ⊃

```
H: READ1'(A,B,C,D,E,F) ⊃  
X:=A ⊃ Y:=B ⊃ Z:=C ⊃ MIN ⊃ U:=W ⊃  
X:=D ⊃ Y:=E ⊃ Z:=F ⊃ MIN ⊃  
IF' U:)W ⊃ THEN' M:=U ⊃ ELSE' M:=W ⊃  
PRINT1'(M) ⊃ STOP' ⊃ START' H ⊃ FINISH' ⊃
```

§ 25. Protseduurid

1. Protseduurikirjeldus. Kinnist ja põhiprogrammist täie likult sõltumatut alamprogrammi nimetatakse protseduuriks. Sõltumatus on protseduuri tähtsaim omadus. Tänu sellele omadusele võib protseduure kasutada erinevates põhiprogrammides.

MALGOL-is defineeritakse protseduur protseduurikirjelduse abil. Protseduurikirjeldus algab reserveeritud sõnaga PROCEDURE', millele järgnevad kirjeldatava protseduuri identifikaator (protseduuri nimetus), protseduuri formaalsete parameetrite loetelu ümarsulgudes, märk ⊃ ja selle kirjeldusega protseduurina esitatav operaator:

$$\text{PROCEDURE' } P(t_1, t_2, \dots, t_k) \supset \mathcal{F} \supset .$$

Protseduurina esitatav operaatorite grupp pannakse operaatorsulgudesse. Protseduurikirjelduse üldine kuju on siis järgmine:

$$\text{PROCEDURE' } P(t_1, t_2, \dots, t_k) \supset \\ \text{BEGIN' } \mathcal{F}_1 \supset \mathcal{F}_2 \supset \dots \supset \mathcal{F}_n \supset \text{END' } \supset ,$$

kus P on protseduuri nimetus, t_1, t_2, \dots, t_k formaalsete parameetrite loetelu ja $\mathcal{F}, \mathcal{F}_1, \mathcal{F}_2, \dots, \mathcal{F}_n$ operaatorid.

PROCEDURE' $\mathcal{P}(t_1, t_2, \dots, t_n) \ni$

nimetatakse protseduuri peaks. Protseduuri pea annab nimetu-
se protseduurile ja tema formaalsetele parameetritele. Prot-
seduuri formaalseteks parameetriteks on identifikaatorid.
Protseduuri formaalsete parameetrite hulka peavad kuuluma
kõik need identifikaatorid, millede vahendusel protseduur te-
ma täitmise ajaks seostatakse põhiprogrammiga. Osa parameet-
reid on sealjuures vajalikud selleks, et põhiprogramm saaks
protseduurile dikteerida arvutusprotsessi sooritamiseks va-
jalikke lähteandmeid. Neid parameetreid nimetatakse sisend-
parameetriteks. Teine osa parameetreid, mida nimetatakse väl-
jundparameetriteks, annavad põhiprogrammile teada selle
arvutusprotsessi tulemused.

Protseduurikirjelduse teist osa

BEGIN' $\mathcal{P}_1 \ni \mathcal{P}_2 \ni \dots \ni \mathcal{P}_n _ \text{END}' \ni$

nimetatakse protseduuri kehaks. Protseduuri keha on kinnine
ja põhiprogrammist sõltumatu programmiosa. See tähendab, et
kõik identifikaatorid ja operaatorite märgendid kehtivad ai-
nult selle protseduuri ulatuses. Protseduuri koostamise juu-
res ei tule arvestada neid tähistusi, mida kasutatakse põhi-
programmis.

Protseduuri koosseisu kuuluvate operaatorite poole ei
saa pöörduda väljastpoolt. Need suunamisoperaatorid ja lüli-
tid, mis pöörduvad protseduuri koosseisu kuuluvate operaato-
rite poole väljastpoolt, jäävad täitmata.

Kõiki massiive, mida kasutatakse protseduuri kehas ja
millede nimetused ei kuulu protseduuri formaalsete parameet-
rite hulka, tuleb protseduuri kehas kirjeldada. Need massii-
vikirjeldused kehtivad ainult antud protseduuri piirides. Kui
selliste massiivide rajad sõltuvad formaalsetest parameetri-
test, siis tuleb need massiivid protseduuri lõpus kustutada
standardse protseduuri DELETE' abil. Kustutamine on vajalik
selleks, et protseduuri korduva kasutamise puhul võivad nen-
de massiivide rajad muutuda. Kustutamata massiivi korduva
kirjeldamise korral jääb aga kehtima esimene kirjeldus (vt.
§ 20).


```

PROCEDURE' RV(A,B,C,T,X1,X2) ⊃
BEGIN' D:=B2-4*A*C ⊃
    IF' D=)0 THEN' BEGIN' T:=0 ⊃
        X1:= (-B + SQRT'D)/(2*A) ⊃
        X2:= (-B - SQRT'D)/(2*A) ⊃
    END' ELSE' BEGIN' T:=1 ⊃
        X1:= -B/(2*A) ⊃
        X2:= SQRT'(-D)/(2*A) ⊃ END'
END' ⊃

```

Protseduur, mis leiab absoluutväärtuse poolest suurima elemendi matriksis A. Sisendparameetriteks on siin matriksi nimetus A ja matriksi ridade ja veergude arvud M ja N. Väljundparameetriteks on absoluutväärtuselt suurima elemendi väärtus Y ja selle elemendi indeksid P ja Q.

```

PROCEDURE' ABSMAX (A,M,N,Y,P,Q) ⊃
BEGIN' Y:= A.(1.1). ⊃
    FOR' i:=1 STEP' 1 UNTIL' M DO'
    FOR' j:=1 STEP' 1 UNTIL' N DO'
        IF' ABS'(A.(i,j).) > ABS'Y THEN'
            BEGIN' Y:= A.(i,j). ⊃ P:=i ⊃ Q:=j ⊃ END'
    END' ⊃

```

2. Protseduuri poole pöördumine. Pöördumine protseduuri-rina defineeritud alamprogrammi poole toimub protseduurioperaatoriga

$\mathcal{P}(p_1, p_2, \dots, p_k) \ni$,
 kus \mathcal{P} on täitmisele tuleva protseduuri identifikaator ja p_1, p_2, \dots, p_k tema tegelike parameetrite loend. Protseduuri tegelikeks parameetriteks võivad olla muutujate identifikaatorid, arvud, avaldised, massiivide identifikaatorid ja protseduuride identifikaatorid. Massiivi nimetuse tunnuseks on punkt ja protseduuri nimetuse tunnuseks ümarsulud (), mis on kirjutatud vastava identifikaatori järele. Protseduuri tegelike parameetrite loend protseduurioperaatoris peab sisaldama täpselt niisama palju parameetreid kui formaalsete parameetrite loend protseduurikirjelduses.

Protseduuri formaalset ja tegelikku parameetrit nimeta-

tatakse vastavateks, kui nad parameetrite loendites t_1, t_2, \dots, t_n ja p_1, p_2, \dots, p_n paiknevad samadel kohtadel.

Õeldakse, et protseduuri tegelikud parameetrid on kooskõlas vastavate formaalsete parameetritega, kui protseduuri kehas esinevate formaalsete parameetrite asendamine vastavate tegelike parameetritega annab süntaktiliselt korrektse operaatori. Protseduuri parameetrite kooskõlastamine, s.t. jälgimine, et peale formaalsete parameetrite asendamist vastavate tegelike parameetritega saame korrektse protseduuri keha, on täielikult programmeerija ülesanne.

Protseduurioperaatori täitmine toimub järgmisel viisil:

1. Translaator asendab formaalsed parameetrid protseduuri kehas protseduurioperaatoris loetletud vastavate tegelike parameetrite jooksvate väärtustega. Kui sealjuures tegelikuks parameetriks on avaldis, siis vastavale formaalsele parameetrile omistatakse selle avaldise väärtus.

2. Toimub protseduuri \mathcal{P} kehas toodud operaatori

BEGIN' $\mathcal{P}_1 \mathcal{P}_2 \dots \mathcal{P}_n$ END' \mathcal{E}

täitmine. MALGOL-i translaator on koostatud nii, et pärast viimase operaatori \mathcal{P}_n täitmist, millele järgneb operaatorisulg END', suunatakse juhtimine protseduurioperaatorile $\mathcal{P}(p_1, p_2, \dots, p_n)$ vahetult järgnevale operaatorile põhi-programmis.

Protseduurioperaatori näiteid.

Ruutvõrrandi

$$16,9t^2 + (R-s)t + Q + 12 = 0$$

lahendamine toimub operaatoriga

$$RV(16,9, R-S, Q+12, E, T1, T2) \mathcal{E},$$

kusjuures $E=0$, kui $T1$ ja $T2$ on ruutvõrrandi reaalsed lahendid, ja $E=1$, kui $T1$ ja $T2$ on vastavalt võrrandi lahendite reaal- ja imaginaarosa.

Maatriksi $C.(1:30, 1:20)$. absoluutväärtuse poolest suurima elemendi leidmiseks kasutame protseduurioperaatorit

$$ABS\text{MAX}(C., 30, 20, X, Y, Z) \mathcal{E}.$$

Leitud elemendi ja tema indekse väärtused omistatakse vastavalt muutujatele X, Y, Z .

Ülesanne. Koostada programm summa

$$S = \sum_{i=1}^{500} \frac{x_i \sqrt[4]{x_i}}{1+2x_i^2}$$

arvutamiseks, kasutades protseduuri $y = \sqrt[n]{A}$ arvutamiseks iteratsioonimeetodit

$$y_0 = A$$

$$y_{i+1} = y_i + \frac{1}{n} \left[\frac{A}{y_i^{n-1}} - y_i \right]$$

$$|y_{i+1} - y_i| \leq \epsilon$$

ja protseduuri A^n arvutamiseks. Nõutavaks täpsuseks on 10^{-7} .

Lahendus.

PROCEDURE' ASTE (A,N,Y) ⓓ

BEGIN' Y:=1 ⓓ

FOR' i:=1 STEP' 1 UNTIL' N DO' Y:=Y*A

END' ⓓ

PROCEDURE' JUUR (A,N,EPS,Y) ⓓ

BEGIN' Y:=A ⓓ

B: ASTE (Y, N-1, Z) ⓓ

Y1:= Y + (A/Z - Y)/N ⓓ

IF' ABS(Y1 - Y) > EPS THEN'

BEGIN' Y:=Y1 ⓓ GOTO' B END' ⓓ

Y:=Y1

END' ⓓ

COMMENT' PÕHIPROGRAMM-SUMMA ⓓ

H: ARRAY' X.(1:500). ⓓ

READAR'(X.) ⓓ

EPS:=1E-7 ⓓ S:=0 ⓓ

FOR' i:=1 STEP' 1 UNTIL' 500 DO'

BEGIN' JUUR(X.(i), 4, EPS, Y) ⓓ

S:= S + X.(i) * Y / (1 + 2 * X.(i) * X.(i))

END' ⓓ

PRINT1'(S) ⓓ STOP' ⓓ START' H ⓓ FINISH' ⓓ

Kui protseduuri keha lõpeb omistamisoperaatoriga

$$k := v_i$$

siis jätab translaator avaldise v väärtuse peale protseduurist väljumist summaatorisse. See asjaolu lubab MALGOL-programmides kasutada protseduurioperaatoreid funktsioonidena. Funktsiooni väärtuseks võetakse selle avaldise väärtus, mille arvuti toob vahenditult enne protseduurist väljumist summaatorisse. Funktsioonina kasutatava protseduuri kirjelduses ei ole vaja näidata väljundparameetrit.

N ä i t e i d.

Ülesanne. Kirjutada operaator avaldise

$u = v - a + a [a + b + (c - d)\cos(a + b)] - v \cos a$
arvutamiseks, kasutades protseduurikirjeldust

PROCEDURE'F(X,Y,Z) \ni Z:= X + Y?COS'X \ni

Lahendus.

$U := V + A?F(A+B, C-D, T) - F(A,V,T) \ni$

See omistamisoperaator tegutseb järgmiselt. Protseduurioperaator $F(A+B, C-D, T)$ arvutab avaldise

$$a + b + (c - d)\cos(a - b)$$

väärtuse ja paigutab selle muutuja T jaoks määratud mälupeassa. Et see väärtus jääb pärast protseduuri kehast väljumist summaatorisse, siis tehakse järjest tehted

$$A ? F(A+B, C-D, T)$$

$$V + A?F(A+B, C-D, T)$$

ning tulemus paigutatakse mingisse tööpeassa. Protseduurioperaator $F(A,V,T)$ arvutab avaldise $a + v \cos a$ väärtuse ja paigutab selle muutuja T jaoks määratud mälupeassa. Et see väärtus jääb pärast protseduuri kehast väljumist summaatorisse, siis sooritatakse tehe

$$V + A ? F(A+B, C-D, T) - F(A,V,T)$$

ning tulemus paigutatakse muutuja V jaoks määratud mälupeassa.

Ülesanne. Koostada programm suuruse

$$M = \max \{ \min(A, B, C), \min(D, E, F) \}$$

leidmiseks, kasutades funktsiooni $F = \min(X, Y, Z)$ (skeem 14).

Lahendus.

PROCEDURE' MIN(X,Y,Z) ⊃

BEGIN' IF' X:)Y ⊃ THEN' GOTO' L1 ⊃ ELSE' GOTO' L2 ⊃

L1: IF' Y:)Z ⊃ THEN' GOTO' L3 ⊃ ELSE' GOTO' L4 ⊃

L2: IF' X:)Z ⊃ THEN' GOTO' L3 ⊃ ELSE' GOTO' L5 ⊃

L3: W:=Z ⊃ GOTO' L6 ⊃

L4: W:=Y ⊃ GOTO' L6 ⊃

L5: W:=X ⊃

L6: W:=W ⊃ END' ⊃

COMMENT' PÕHIPROGRAMM ⊃

H: READ1'(A,B,C,D,E,F) ⊃

IF' MIN(A,B,C) :) MIN(D,E,F) ⊃ THEN'

M:= MIN(A,B,C) ⊃ ELSE' M:= MIN(D,E,F) ⊃

PRINT1'(M) ⊃ STOP' ⊃ START' H ⊃ FINISH' ⊃

Ülesanne. Koostada programm suuruse

$$W = \frac{2x^3 - 5x^2 + 6x - 4}{2y^3 - 5y^2 + 6y - 4} - 9(2z^3 - 5z^2 + 6z - 4)$$

arvutamiseks.

Lahendus.

PROCEDURE' F(X) ⊃ Y := 2?X?X?X - 5?X?X + 6?X - 4 ⊃

COMMENT' PÕHIPROGRAMM ⊃

A: READ1'(X,Y,Z) ⊃

W:= F(X)/F(Y) - 9?F(Z) ⊃

PRINT1'(W) ⊃ STOP' ⊃ START' A ⊃ FINISH' ⊃

STANDARDSEID PROTSEDUURID

§ 27. Standardse protseduuri mõiste

MALGOL-programmides on teatud protseduure lubatud kasutada, ilma et neid oleks enne kirjeldatud vastavate protseduuri kirjeldustega. Selliseid protseduure nimetatakse standardseteks. Standardsete protseduuride programmid on olemas arvuti mälus. Et standardset protseduuri tööle rakendada, tuleb programmi kirjutada vastav protseduurioperaator. Standardsete protseduuride nimetused kuuluvad MALGOL-i reserveeritud sõnade hulka ja nende järel kirjutatakse apostroof. MALGOL-i standardsete protseduuride hulka kuuluvad

- 1) programmid elementaarfunktsioonide väärtuste arvutamiseks (§4);
- 2) programmid informatsiooni viimiseks arvutisse;
- 3) programmid informatsiooni väljatrükkimiseks arvutist;
- 4) programmid välismälu poole pöördumiseks;
- 5) programmid mõnede operatsioonide läbiviimiseks mas-siividega;
- 6) programm määratud integraali ligikaudse väärtuse arvutamiseks.

§ 28. Informatsiooni viimine arvutisse

Informatsiooni sisseviimiseks kasutatakse MALGOL-is standardseid protseduure READ. Standardseid protseduurid READ loevad informatsiooni perfolindilt arvuti mälusse, teisendavad informatsiooni kahendsüsteemi ja kontrollivad sisseviimise õigsust. Kõikidel nendel juhtudel, kui informatsiooni sisseviimisel esineb häireid (informatsioon ei mahu mälusse, osa informatsioonist on jäänud perfolindile kandmata jne.), arvuti peatub ja seadeldis БПМ-20 trükitab välja peatumise põhjuse (vt.tabel 3).

Lihtmuutujate väärtuste lindilt lugemine toimub standardsete protseduuridega

READ1' (x,y,z, ~, t) ja READ1OCTAL'(x,y,z, ~, t).

Lihtmuutujate väärtused peavad olema perforeeritud ühe massiivina sisseviimismärkide vahele. READ1' (x,y,z, ~, t) on ette nähtud kümnendarvude ja READ1OCTAL' (x,y,z, ~, t) kaheksandarvude viimiseks perfolindilt arvuti mälusse. Pärast arvude sisseviimist toimub nende teisendamine kahendsüsteemi arvudeks.

Massiivide sisseviimiseks perfolindilt arvuti mälusse kasutatakse standardseid protseduure

READARRAY' (A.,B.,C., ~, F.) ja READAOCTAL'(A.,B.,C., ~, F).

Iga massiiv peab olema perforeeritud oma sisseviimismärkide vahele. READARRAY' (A.,B.,C., ~, F.) on ette nähtud kümnendarvude ja READAOCTAL' (A.,B.,C., ~, F.) kaheksandarvude sisseviimiseks arvuti mälusse. Pärast sisseviimist teisendatakse massiivide elemendid kahendsüsteemi arvudeks.

Suuremate kümnendarvudest koosnevate massiivide sisseviimiseks kasutatakse standardset protseduuri

READACONTR' (A.).

Selle protseduuri puhul jaotatakse sisseviidav massiiv osadeks ja arvutatakse iga osa jaoks kontrollsumma. Massiivi A perforeerimisel kantakse lindile iga elementide grupi järel leitud summa ja probell. Kogu massiiv A. perforeeritakse sisseviimismärkide vahele. Massiivi A. kirjeldamise puhul tuleb arvestada, et iga grupi juurde kuulub summa ja probell. Seega, kui massiiv koosneb m elemendist, mis jaotatakse n gruppiks, tuleb massiivi pikkuseks märkida massiivikirjelduses $p = m + 2n$. Standardne protseduur READACONTR' (A) viib massiivi A. arvuti mälusse ja kontrollib sisseviimise õigsust gruppide kaupa. Juhul kui mõne grupi puhul kontrollsummad ei lange ühte, trükib seadeldis EIM-20 välja järgmised andmed:

- 1) grupi number kaheksandarvuna;
- 2) grupi elemendid ja kontrollsumma kümnendarvudena;
- 3) elementide arv grupis kümnendarvuna;
- 4) arvuti poolt leitud kontrollsumma kümnendarvuna.

Kui kontrollsummad langevad ühte, siis trükitab seadeldis БИМ-20 välja grupi pikkuse ja arvuti poolt leitud grupi kontrollsumma kümnendarvudena.

Kontrollitud grupi elemendid teisendatakse kahendsüsteemi. Pärast kontrollimist surutakse massiiv kokku, s.o. jäetakse ära summad ja probellid. Massiivi A. lõplik pikkus (ilma summade ja probellideta) jääb summaatorisse. Omistamisoperaator

Z:=FLOAT'READAC'(A.)

omistab selle massiivi A. pikkuse muutujale Z.

Tekstide viimine perfolindilt arvuti mälusse toimub standardse protseduuriga

READTEXT'(B.,a,n).

Siin B. on sisseviidava massiivi nimetus, n-massiivi pikkus ja a - selle pesa aadress, millest alates toimub massiivi paigutamine arvuti mälusse.

Standardset protseduuri

MEASURE'(M.,k)

kasutatakse sel juhul, kui sisseviidavate massiivide pikkused ei ole teada. Protseduur MEASURE' loeb massiivid perfolindilt arvuti mälusse, leiab nende pikkused ja omistab saadud väärtused massiivi M elementidele. Parameetriga k näidatakse, kas kontrollitavad massiivid koosnevad arvudest (k=0) või tekstist (k=1).

§ 29. Informatsiooni väljatoomine arvutist

Standardsed protseduurid PRINT trükitavad arvuti mälust informatsiooni välja, rakendades tööle seadeldise БИМ-20. Protseduur

PRINT1'(x,y,z,~,t)

trükitab välja muutujate x, y, z, ~, t väärtused, teisendades nad enne kümnendsüsteemi. Protseduur

PRINTOCTAL'(x,y,z,~,t)

trükitab muutujate x, y, z, \dots, t väärtused välja kaheksandsüsteemi arvudena.

Massiivide väljatrükkimine toimub protseduuridega PRINTARRAY' (A., B., ..., F.) ja PRINTBOOLEAN' (A., B., ..., F.). Sealjuures jäetakse iga massiivi vahele pikem vahe. Vahedega näidatakse ära ka indeksite muutumine. Näiteks kahedimensionaalse massiivi elemendid trükitakse välja ridade kaupa, kusjuures iga rea järel jäetakse vahe. Protseduur PRINTARRAY' trükitab massiivide elemendid välja kümnendsüsteemi arvudena ja protseduur PRINTBOOLEAN' kaheksandsüsteemi arvudena.

Translaatori poolt koostatud programmi silumise kerendamiseks on ette nähtud protseduur PRINTLABEL'(A.).

Standardseid protseduure

TEXT10'(t, m, A, B, ..., C) ja TEXT8'(A, B, ..., C)

kasutatakse informatsiooni väljatoomiseks teletaibiga, s.t. perforatsiooniperforaatori nr.2 abil ja trükkimiseks. Parameetriteks A, B, ..., C võivad olla muutujate identifikaatorid ja tekstid. Tekstiks võib olla igasugune põhisümbolite jada, mis ei sisalda apostroofe ja on suletud kahe apostroofi vahele. Näiteks

'_ < = OTBET'

'_ X ='

on tekstid. Kui tekst algab sümboliga < = (s.t. uelt realt), siis tuleb tema ette kirjutada probell.

Parameetrid t ja m määravad trükitavate arvude formaadi: t numbrikohta arvu täisosas ja m numbrikohta murdosas. Iga arvu ette trükitakse probell ja arvu märk. Seega on iga trükitava arvu pikkus $t+m+3$. Täisarvudes trükitakse punkti ja murdosa numbrikohtade asemel välja probellid. Kui väljatrükitava arvu täisosas on rohkem kui t kohta, siis jagatakse arv kümne astmega ja trükitakse välja jagatis nõutud formaadis ning tema järele - kümne aste. Kui kõik parameetrid on tekstid, siis võib t ja m ära jätta.

Näide: Olgu arvuti mälus järgmised arvud

Y +23.05012

X1 -3.00278

X2 +25

Protseduur

TEXT10'(2,4,' $\perp \leq Y = \perp$ ', Y,' $\perp \leq P(X) = \perp$ ',X1,X2) \ni

trükkib välja

Y = +23.0501

P(X) = -03.0027 +25

Tekstide väljatrükkimisel tuleb silmas pidada, et üks rida saab koosneda maksimaalselt 63 sümbolist (kaasa arvatud probellid).

Protseduur TEXT10' trükkib muutujate väärtused välja kümnendsüsteemi arvudena. TEXT8' trükkib arvud välja kaheksandsüsteemi arvudena.

Protseduur

TEXT'(B.,a,n,k)

perforeerib perforaatoril nr.2 tekstist koosneva massiivi B., alates pesast a. Väljatrükitav massiiv võtab mälus enda alla n pesa. Kui $k=1$, siis tekst perforeeritakse ladina tähtedega; kui $k=0$, perforeeritakse tekst vene tähtedega.

Protseduur

PUNCHARRAY'(A.,B., \sim ,F.)

perforeerib massiivide A.,B., \sim ,F. elemendid kaheksandsüsteemis, rakendades tööle perforaatori nr.1. Iga massiiv perforeeritakse sisseviimismärkide vahele. Üksikute massiivide vahele jäetakse tühi lindiosa.

§ 30. Välismälu poole pöördumine

Arvutiga lahendatavad ülesanded on sageli seotud väga suure andmete hulgaga, mis mitmekordselt ületab operatiivmälu mahtuvuse. Niisugustel juhtudel kasutatakse välismälu. Välismälu mahutab piiramatul hulgal andmeid, kuigi töötab aeglasemalt operatiivmälust. Välismälus säilitatakse vajaduse korral ka arvutuste tulemusi ja programme.

Arvuti Minsk-22 välismäluks on 80-100 m pikkused magnetlindid. Iga magnetlint jaguneb 64 tsooniks. Igasse tsoonini on võimalik kirjutada 2048 37-kohalist kaheksandarvu.

Välismälu poole pöördumiseks on MALGOL'is olemas kaks standardset protseduuri

WRITETAPE' (L.,A.,S) ja READTAPE' (L.,A.,S), milledes A. on massiivi nimetus, S on selle muutuja identifikaator, millele on omistatud kontrollsumma väärtus, L on kas kaheksandarv või sellise muutuja identifikaator, millele on omistatud väärtus kaheksandarvuna. Protseduur WRITETAPE' (L.,A.,S) kirjutab massiivi A. elemendid magnetlindile. L näitab mitmendast pesast alates tuleb massiivi A. elemendid kanda magnetlindile.

Protseduur READTAPE' (L.,A.,S) loeb magnetlindilt operatiivmälusse massiivi A. elemendid, mis asuvad lindil alates pesast L.

Protseduuride WRITETAPE' (READTAPE') puhul toimub massiivi A. kontrollsumma S võrdlemine magnetlindile (vastavalt operatiivmällu) loetud massiivi kontrollsummaga. Juhul kui kontrollsummad, ei lange ühte, arvuti peatub ja trükitab välja peatumise tunnuse (vt.tabel 3).

§ 31. Operatsioonid massiividega

MALGOL-is on terve rida programme operatsioonideks massiividega vormistatud standardsete protseduuridena. Pöördumine nende protseduuride poole toimub vastavate protseduurioperaatoritega. Paljude standardsete protseduuride puhul peavad massiivid rahuldama teatavaid tingimusi. Kui massiivid neid tingimusi ei täida, siis arvuti peatub ja seadeldis EIIM-20 trükitab välja peatumise tunnuse (vt.tabel 3).

DELETE'(A.) ⊃

Massiivi A. kustutamine arvuti mälust.

TRANS'(A.,B.) ⊃

Massiivi A. elementide väärtused omistatakse massiivi B. elementidele. Massiivide A. ja B. pikkused peavad olema võrdsed.

$$\text{ADD}^*(A., B., C.) \ni$$

Liidetakse massiivide A. ja B. vastavad elemendid ja summadest moodustatakse massiiv C.. Massiivid A., B. ja C. peavad olema ühepikkused.

$$\text{MULT}^*(A., B., C.) \ni$$

Maatriksite A. ja B. korrutamine, tulemuseks on maatriks, mis paigutatakse maatriksi C. jaoks reserveeritud kohale. Protseduur on teostatav, kui

- 1) maatriksite A. ja C. ridade arvud on võrdsed;
- 2) maatriksi A. veergude arv võrdub maatriksi B. ridade arvuga;
- 3) maatriksite B. ja C. veergude arvud on võrdsed.

$$\text{INVERS}^*(A.) \ni$$

Maatriksi A. pöördmaatriksi A^{-1} leidmine Jeršovi meetodil. Pärast protseduuri asendub maatriks A. arvuti mälus oma pöördmaatriksiga.

Protseduur on teostatav, kui arvuti mälusse mahub üks ühedimensionaalne massiiv pikkusega n (n on maatriksi A. järk), mis kirjeldatakse protseduuri kehas.

$$\text{GAUSS}^*(A.) \ni$$

Lineaarse algebralise võrrandisüsteemi

$$\sum_{j=1}^n a_{ij} x_j = b_i \quad (i = 1, 2, \dots, n)$$

lahendamine Gaussi meetodiga. Massiiviks A. on siin süsteemi laiendatud maatriks

$$\begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} & b_1 \\ a_{21} & a_{22} & \dots & a_{2n} & b_2 \\ \dots & \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nn} & b_n \end{pmatrix}$$

EIGENV'(A., B., V., N) \ni

Sümmeetrilise N-järku ruutmaatriksi A. omaväärtuste ja omavektorite leidmine. Eeldatakse, et massiiv A. on esitatud kolmnurkmaatriksina, s.t. kirjeldatud ühedimensionaalse maatriksina, mille pikkuseks on $\frac{N(N+1)}{2}$.

Maatriksi A. omaväärtused salvestatakse pärast selle protseduuri täitmist ühedimensionaalse massiivina B. arvuti mälusse. Maatriksi A. omavektorite komponendid salvestatakse arvuti mälusse kui kahedimensionaalse massiivi V. veerud. Pärast protseduuri maatriks A. ei säili arvuti mälus.

COMBINATION'(A., j, m, t) \ni

Täisarvude tihendatud salvestamine arvuti mälusse.

Ühte mälupessa paigutatakse mitu täisarvu. Vajaduse korral ühendatakse mitu 37-kohalist mälupesa üheks pikaks mälupesaks. Pikas mälupesas kasutatakse ühtset kahendkohtade numeratsiooni, mis algab esimese mälupesa esimese kahendkohaga ja lõpeb viimase mälupesa viimase kahendkohaga. Kui pikaks mälupesaks on ühendatud n 37-kohalist mälupesa, siis kahendkohtade järjekorranumbriteks on

$$0, 1, 2, \dots, 37n-1.$$

Pikk mälupesa kirjeldatakse MALGOL-programmides ühedimensionaalse massiivina, mille pikkus määrab ühendatavate mälupesade arvu. Näiteks kirjelduse A.(1:10). või A.(0:9). puhul ühendatakse 10 mälupesa, kusjuures, kahendkohad nummerdatakse

$$0, 1, 2, 3, \dots, 369.$$

Protseduuri COMBINATION' kasutatakse nii täisarvude viimiseks massiivi A. kui ka täisarvude tootmiseks massiivist A. summaatorisse.

Protseduurioperaator COMBINATION'(A., j, m, t) \ni paigutab muutuja t väärtuse tihendatult massiivi A. koosseisu. Protseduurioperaatoris on parameetritel järgmine tähendus:

A. - tihendatud massiivi nimetus;

j - selle kahendkoha number massiivis A., millest algab täisarvu salvestamine;

m - täisarvu salvestamiseks määratud vahendkohtade arv (sealjuures jääb üks koht märgi ja m-1 kohta täisosa jaoks),
 $m \leq 28$;

t - selle muutuja identifikaator, mille täisosa salvestatakse tihendatult arvuti mälusse (murdosa jäetakse ära).

Protseduurioperaator COMBINATION'(A.,j,m) ⊃ toob täisarvu, mis on tihendatult salvestatud massiivi A., summaatorisse. Parameetritel on selles operaatoris järgmine tähendus:

A. - massiivi nimetus;

j - kahendkoha järjekorranumber;

m - nende kahendkohtade arv, milledesse on paigutatud summaatorisse toodav arv. Protseduurioperaatorit COMBINATION'(A.,j,m) ⊃ kasutatakse MALGOL-programmis funktsioonina, näiteks

t := COMBIN'(A.,j,m) ⊃

Protseduur COMBINATION' on teostatav kui massiiv A. on kirjeldatud ning kui täisarv mahub massiivis A. tema jaoks määratud kohale.

BOOLEAN'(A.,j,t) ⊃

Boole'i muutujate tõeväärtuste tihendatud salvestamine arvuti mälusse.

Iga Boole'i muutuja tõeväärtuse salvestamiseks on vaja üht kahendkohta.

Protseduuri BOOLEAN kasutatakse nii tõeväärtuste viimiseks massiivi A. kui ka tõeväärtuste toomiseks massiivist A. summaatorisse.

Boole'i muutuja t tõeväärtuse paigutamine tihendatult massiivi A. toimub protseduurioperaatoriga

BOOLEAN'(A.,j,t) ⊃

kus

A. on Boole'i muutujate tihendatud massiivi nimetus;

j on massiivi A. selle kahendkoha number, kuhu salvestatakse Boole'i muutuja t väärtus.

Protseduurioperaator BOOLEAN'(A.,j) ⊃ toob massiivist A. summaatorisse selle tõeväärtuse, mis on tihendatult salvestatud j-ndale kahendkohale. Selles protseduurioperaatoris on A. tihendatud massiivi nimetus ja j on kahendkoha järjekorranum-

ber. Tõeväärtus paigutatakse summaatori märgikohale. MALGOL-programmides kasutatakse protseduurioperaatorit BOOLEAN'(A.,j) funktsioonina.

Protseduur BOOLEAN' on teostatav ainult siis, kui massiiv A. on kirjeldatud ning kui t on Boole'i muutuja.

§ 32. Integraali arvutamine

Määratud integraali $\int_a^b f(x)dx$

ligikaudse väärtuse arvutamiseks Simpsoni valemi järgi etteantud täpsusega ε kasutatakse standardset protseduuri

SIMPSON'(F(),A,B,EPS,H,T) \exists .

Protseduurioperaatori parameetritel on järgmine tähendus:

A - integraali alumine raja;

B - integraali ülemine raja;

EPS - etteantud täpsus;

H - sammu pikkus

F() - mittestandardne protseduur, mis on kirjeldatud funktsioonina integraalialuse funktsiooni $f(x)$ või $f(x,T)$ arvutamiseks;

T - muutuja (aritmeetiline avaldis) või massiiv sõltuvalt sellest, kas integraalialune funktsioon sõltub ühest või mitmest parameetrist. Kui integraalialusel funktsioonil ei ole ühtegi parameetrit, siis T võib protseduurioperaatori tegelike parameetrite loetelus puududa.

Protseduurioperaatorit SIMPSON' kasutatakse MALGOL-programmides ainult funktsioonina.

N ä i t e i d:

Ülesanne. Arvutada

$$\int_0^1 (a + \sin x)dx$$

parameetri a väärtuste jaoks 0,1; 0,15; -10,3; -2 täpsusega kuni 10^{-5} . Integreerimissammuks võtta 0,1.

Lahendus.

```
PROCEDURE' G(X,A) ⊃ Y:=A+SIN'X ⊃
```

```
COMMENT' PÕHIPROGRAMM INTEGRAL ⊃
```

```
H: EPS:=1⋅10(-5) ⊃
```

```
FOR'A:= 0.1, 0.15, -10.3. -2 ⊃DO'
```

```
  BEGIN' INT:= SIMPSON'(G( ), 0, 1, EPS, 0.1, A) ⊃
```

```
    PRINT1'(A,INT) ⊃END' ⊃
```

```
  STOP' ⊃ START' ⊃ FINISH' ⊃
```

Ülesanne. Arvutada

$$\int_0^1 (a + bx^2 - c \cos x) dx$$

parameetrite a, b, c väärtuste jaoks, mis loetakse perfolin-
dilt. Nõutavaks täpsuseks on 10^{-5} ; integreerimisvahemik jao-
tada kümneks võrdseks osaks.

Lahendus. Ühendame parameetrite väärtused massiiviks
PARAM.

```
PROCEDURE' FUNKTS(X,PARAM.) ⊃
```

```
F:= PARAM.(1). + X?X?PARAM.(2). - (COS'X)?PARAM.(3). ⊃
```

```
COMMENT' PÕHIPROGRAMM INTEGRAL ⊃
```

```
H: ARRAY' PARAM.(1:3). ⊃
```

```
READAR'(PARAM.) ⊃ EPS:=1⋅10(-5) ⊃
```

```
INT:= SIMPSON'(FUNKTS( ), 0, 1, EPS, 0.1, PARAM.) ⊃
```

```
PRINTA'(PARAM.) ⊃ PRINT1'(INT) ⊃
```

```
STOP' ⊃ START'H ⊃ FINISH' ⊃
```

Standardsete protseduuride tabel

Peatumise tunnus	Peatumise põhjus	Protseduur
+++++++13	Massiivi elemendid ei mahu mälusse	INVERS' READ1'
+++++++21	Massiiv ei ole kirjeldatud	READAC'
+++++++41	Kontrollsummad ei ole võrdsed	
+++++++42 < p n q >	Massiivi pikkus perfolindil (q) ei võrdu selle massiivi jaoks reserveeritud mälupesade arvuga (p). n on massiivi mälupesade number	READAC' READAR'
+++++++43 < p 0000 q >	Muutujate arv protseduuri tegelike parameetrite loetelus (p) ei võrdu reserveeritud arvude arvuga (q)	READ1' READ10'
+++++++44	Kontrollsummad ei ole võrdsed	
+++++++45	Kontrollsummad ei ole võrdsed	
+++++++46 < p n q >	Massiivi pikkus perfolindil (q) ei võrdu selle massiivi jaoks reserveeritud mälupesade arvuga (p). n on massiivi mälupesade number	READAO'
+++++++51	Kolmnurkmaatriksi B. pikkus ei ole $\frac{n(n+1)}{2}$ (n on ruutmaatriksi A. ridade ja veergude arv.	ARRQ3'
+++++++52	Massiivide A., B. ja C. pikkused ei ole võrdsed	ADD'
+++++++53	Massiivide A. ja B. pikkused ei ole võrdsed	TRANS'

T a b e l. 3 (järg)

+++++++54	Maatriksite A. ja C ridade arvud ei ole võrdsed	
+++++++55	Maatriksi A. veergude arv ei võrdu maatriksi B. ridade arvuga	MULT'
+++++++56	Maatriksite B. ja C. veergude arvud ei ole võrdsed	
+++++++57	Kolmnurkmaatriksi A. pikkus ei ole $\frac{n(n+1)}{2}$ (n on ruutmaatriksi B. ridade ja veergude arv)	ARR3Q'
+++++++61	Kontrollsummad magnetlindile kirjutamise ja kontroll lugemise puhul ei ole võrdsed	WRITETAPE'
+++++++62	Kontrollsumma pärast magnetlindilt lugemist ei ole võrdne antud kontrollsummaga	READTAPE'
+++++++71	Massiivi ei ole kirjeldatud	
+++++++72	Arv ei mahu massiivi A.	COMBIN'
+++++++73	Arvul on rohkem kui m kohta	
+++++++74	Massiivi ei ole kirjeldatud	BOOLEAN'
+++++++75	t ei ole Boole'i muutuja	
+++++++81	Tekst ei mahu massiivi B.	
+++++++82	Teksti pikkus lindil ei ole n pesa	READTEXT'
+++++++83	Kontrollsummad ei ole võrdsed	
+++++++84	Massiivi ei ole kirjeldatud	TEXT'
+++++++85	Pesad $a + a + n$ ei kuulu massiivi C	

§ 33. Transleerimise käik

Translaatoriks nimetatakse programmi, mille alusel arvuti tõlgib MALGOL-programmid käskude keelde. Translaatorit hoitakse alaliselt magnetlintidel. Arvuti Minsk-22 translaatori pikkus on 5500 käsku. Standardsete protseduuride kogu võtab enda alla 2800 mälupeša.

MALGOL programmide transleerimise juures kasutatakse järgmisi seadmeid: perforaatorit nr.2, trükkimiseseadeldist БИМ-20 ja fotosisendit. Perforeerimiseks ja tekstide trükkimiseks kasutatakse telegraafiaparaati.

Transleerimise aeg sõltub programmi pikkusest ja siseseviidava informatsiooni hulgast. Keskmiselt kulub programmi transleerimiseks 2-15 min.

Transleeritav MALGOL-programm kantakse koos algusinformatsiooniga arvuti mälusse. Algusinformatsioon annab arvutile alguskäsud ja näitab arvutile, mitmest massiivist koosneb MALGOL-programm ja milline on programmi number. Arvuti loeb kogu perfolindile kantud informatsiooni mälusse. Peale seda toimub programmi transleerimine automaatselt.

Programmi transleerimise esimeseks etapiks on programmi silumine, s.t. vigade avastamine ja parandamine programmis. Arvuti kontrollib, kas MALGOL-programm on koostatud vastavalt MALGOL-i reeglitele. Kui arvuti avastab programmis vigu, trükib ta need välja. Programmeerija ülesandeks on koostada paranduste massiiv, kanda see perfolindile ja viia arvuti mälusse. Arvuti teeb programmis kõik ettekirjutatud parandused, kontrollib parandatud programmi uuesti ja vigade avastamise korral trükib need välja. Sellele järgneb uue vigade paranduse koostamine jne. Programmi parandamine kestab nii kaua, kuni arvuti enam uusi vigu välja ei trüki. Parandatud programmi jaoks koostab arvuti mälujaotuse, tõlgib programmi käskude keelde ja kirjutab selle magnetlindi sellesse

tsooni, mis vastab programmi numbrile. Arvuti perforeerib pa-
randatud programmi lindile, trükib välja mälujaotuse ja trans-
leeritud programmi.

§ 34. Programmi silumine

Programm jaotatakse perforeerimise ajal ridadeks märgi
< Ξ abil. Arvuti nummerdab read ja trükib programmi ridade
kaupa välja.

Juhul kui arvuti avastab programmis vea, trükib ta välja
vea karakteristiku (vt. tabel 4), rea numbrilise viimase kontrol-
litud identifikaatori ja eraldaja ning teksti kuni järgmise
märgini Ξ . Väljatrükitud teksti arvuti ei analüüsi. See tä-
hendab, et selles programmiosas võib olla vigu, mida trans-
laator ei avastanud. Seoses sellega, et osa programmist jäi
analüüsima, võib translaator lugeda ebaõigeteks mõned õiged
konstruktsioonid programmi ülejäänud osas.

Vigu tuleb otsida programmi selles osas, mis eelneb väl-
jatrükitud reale.

Programmides saab teha kolme tüüpi parandusi:

- 1) rea asendamine ühe või mitme uue reaga (selle paran-
duse tunnuseks on A);
- 2) ühe või mitme rea lisamine antud rea ette (selle pa-
randuse tunnuseks on L);
- 3) rea kustutamine (selle paranduse tunnuseks on K).

Iga parandus algab tähega R, millele järgneb rea number
kolmekohalise kaheksandarvuna, paranduse tunnus ning seejärel
kahe märgi \cup vahel üks või mitu uut rida. Tunnuse K järel
kirjutatakse üks \cup .

N ä i t e i d.

RO02 A \cup A := B - C \cup K := SQRT' A \cup \cup

RO10 L \cup IF' A THEN' X := C \cup \cup

RO40 K \cup

RO41 L \cup END' \cup \cup

Iga rea kohta saab vormistada ainult ühe paranduse. Pa-
randused perforeeritakse ühe või mitme massiivina sissevi-
misvärkide vahele.

Arvuti on võimeline avastama ainult neid vigu, mis on seotud MALGOL-keele ebaõige kasutamisega. Peale selle võib programmis esineda ka loogilisi vigu, s.t. vigu, mis on seotud algoritmi endaga. Selliste vigade avastamiseks lastakse arvutil teha kontrollarvutusi koos vahepealsete tulemuste trükkimisega. Selleks kirjutatakse programmi operaatoreid, mis lubavad programmi täitmist juhtida puldilt, näiteks

IF' KEY' OCTAL'10 THEN'PRINTA'(B.,C.) ☉.

Pärast silumise lõppemist need operaatorid kustutatakse. Programmi loogika kontrollimiseks võib arvutil lasta välja trükkida kõikide täidetud operaatorite märgendid ja kasutatud mittestandardsete alamprogrammide ja protseduuride nimetused. Märkendite väljatrükkimine toimub standardse protseduuri PRINTLABEL' abil.

§35. Programmide perforeerimine

MALGOL-programm kirjutatakse vabas vormis paberilehele. Kirjutada tuleb korralikult, mitte unustades apostroofe ja märke ☉. Selleks, et vahet teha tähe O ja number 0 vahel, on soovitatav täht O alla kriipsutada (O). Nendes programmi kohtades, kus probell on tingimata vajalik, märkida sümbol ⊣. Programmides on soovitatav ära näidata kohad, kuhu tuleb lüüa sisseviimise ja reavahetuse märgid.

MALGOL-programmid perforeeritakse telegraafiaparaadi abil lindile ühe või mitme massiivina. Iga massiiv algab sisseviimismärgiga "vene-ladina". Peale sisseviimismärki perforeeritakse mõned probellid, reavahetuse märk ($\langle \equiv \rangle$) ja seejärel MALGOL-programm. Kui sealjuures programm algab ladina tähega, tuleb veel kord lüüa märk "ladina". Iga massiiv lõpeb vähemalt viie probelliga ja sisseviimismärgiga "ladina-vene".

Massiivi maksimaalseks pikkuseks on 3072 märki. Pikem MALGOL-programm viiakse sisse mitme massiivina. Uut massiivi on loomulik alustada operaatori algusest. Iga massiivi vahele jäetakse 20-30 cm tühja linti.

Ridadeks jaotatakse programm reavahetusmärkide ($\langle \Xi \rangle$) abil. Rea maksimaalseks pikkuseks on 63 märki. Reavahetusmärkide põhjal nummerdab arvuti read.

Probelli võib lüüa iga kahe identifikaatori vahele või identifikaatori ja eraldaja vahele. Üleliigseid probelle ei tohi lüüa programmi nendesse kohtadesse, kus nende arv on kindlaks määratud programmeeri ja poolt.

Kui programmeerimisel tehakse vigu, võib neid kohe parandada märgi \mathbb{W} abil. Märk \mathbb{W} kustutab tema ees oleva märgi. Mitme märgi kustutamiseks kasutame mitut märki \mathbb{W} .

Näiteks kirjutis $A:=\mathbb{W}$ on identne kirjutisega $A;$; kirjutis $ABC, _ : X\mathbb{W}\mathbb{W}\mathbb{W}\mathbb{W}$ on identne kirjutisega AB .

Parandused perforeeritakse ühe või mitme massiivina programmi ette.

Informatsioon ja alguskäsud

0001) p m n

0002) -47 00 0001 0000.

0003) -45 00 0004 0005.

0004) -30 00 0002 0000.,

kus p on paranduste massiivide arv,

m - programmi massiivide arv,

n - programmi number,

perforeeritakse sisseviimismärkide vahele ja kleebitakse programmi ette.

Mälujaotuses ja vea karakteristikutes
kasutatavad sõnad

Sõna	Tähendus
NUMBER	arv
NO PLACE	ei mahu
UNKNOWN	ei ole teada
LACK	puudub
PARENTHESES	sulud
LONG EXPRESSION	avaldis on pikk
CORRECT	parandada
PROCEDURE NOT FINISHED	Protseduuri kirjeldus ei ole lõpetatud
???	operaatorsulud ei ole kooskõlas
MEMORY PLAN	mälujaotus
SUBROUTINE	alamprogramm
VARIABLES	muutujad
OPERATING VARIABLES	töömuutujad
CONSTANTS	konstandid
SWITCHES	lülitid
FUNCTIONS	elementaarfunktsioonid
PROCEDURES	protseduurid
LABELS	märgendid
TABLE OF ARRAYS	massiivide tabel
PROGRAM	programm
ELEMENTS OF ARRAYS	massiivide elemendid
START	programmi algus

T a b e l 5

Kitsendused MALGOL-programmide jaoks

Keele element	Maksimaalne arv programmis
Formaalsed parameetrid	15
Muutujad	319
Massiivid	87
Lülitid	7
Alamprogrammid + tsükliid	103
Operaatorite märgendid	287
Protseduurid (standardised + mittestandardised)	63
Elementaarfunktsioonid	31
Konstandid	156

Hind 15 kop.

A
3021

TÜ RAAMATUKOGU



1 0300 01129647 4