



# Programmeeri- mine arvutil

• NAIRII-C •

TALLINN • 1972



TALLINNA POLÜTEHNILINE INSTITUUT

Arvutusmatemaatika kateeder

PROGRAMMEERIMINE ARVUTIL "NAIRII-C"

Õppevahend

P. Piirikivi

J. Tepandi

Tallinn

1972

ТАЛЛИНСКИЙ ПОЛИТЕХНИЧЕСКИЙ ИНСТИТУТ  
Кафедра вычислительной математики

П. Пирикиви, Я. Тепанди

ПРОГРАММИРОВАНИЕ НА ВЫЧИСЛИТЕЛЬНОЙ МАШИНЕ  
"НАИРИ-С"

Учебное пособие

TARTU ÜLIKOOLI  
RAAMATUKOGU

Vastutav toimetaja I. Suuder

---

Trükkida antud 25. X 1972. Paber 60x84/16  
Trükipg. 5,75. Tingpg. 5,25. Arvestuspg. 4,44. Tiraaž 2000.  
MB-09520. TFI rotaprint, Tallinn, Koskla 2/9. Tell. 756  
Hind 16 kop.

## S i s s e j u h a t u s

### I. Ü L E V A A D E A R V U T I S T "N A I R I I - C"

Väikese võimsusega arvuti "Nairii-C" on ette nähtud teaduslik-tehniliste ja inseneriülesannete lahendamiseks. Arvuti on kaheaadressiline ning teeb tehteid fikseeritud ja liikuva komaga kahendarvudega. Arvutisse saab salvestada ja sealt väljastada numbrilist ja alfabeetilist informatsiooni.

Arvuti parameetrid on:

sõna pikkus - 36 kahendkohta;  
sisemälu maht - 1024 sõna;  
fikseeritud mälu - 5 sõna;  
püsिमälu maht - 16384 sõna;  
sisemälu poole pöördumise tsükkel - 24  $\mu$  sek;  
keskmine töökiirus - 300-2000 tehet/sek;

informatsiooni sisestamise kiirus:

perfolindilt - 100 märki/sek;  
kirjutusmasinalt - 8 märki/sek;

informatsiooni väljastamise kiirus:

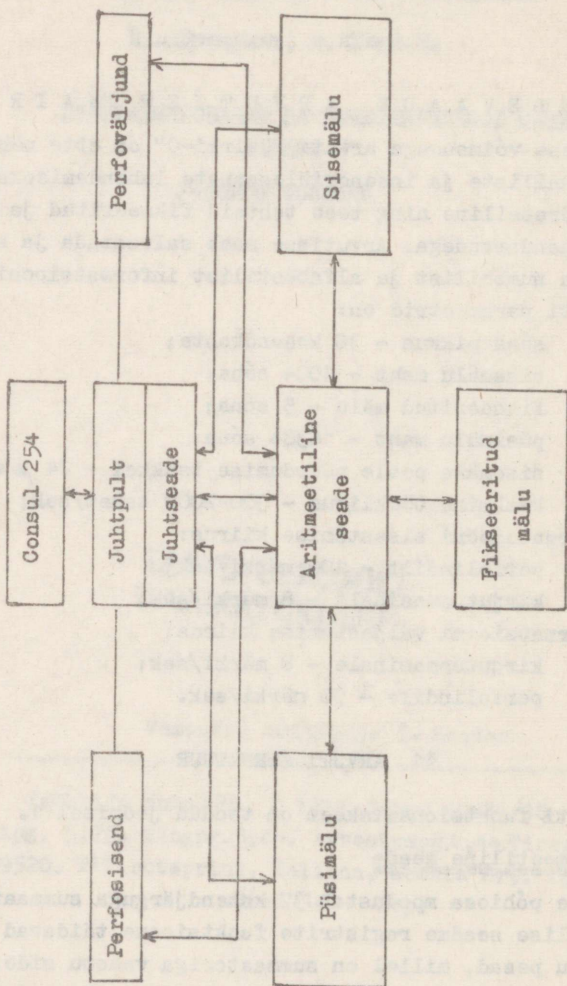
kirjutusmasinale - 8 märki/sek;  
perfolindile - 70 märki/sek.

### §1 ARVUTI STRUKTUUR

Arvuti funktsionaalskeem on toodud joonisel 1.

#### 1.1. Aritmeetiline seade

Selle põhiosa moodustab 37 kahendjärguga summaator. Aritmeetilise seadme registrite funktsioone täidavad fikseeritud mälu pesad, millel on summaatoriga vahetu side. Summaatori 37-st järgust on 37. järk tehniline lisajärk, mis on vajalik mõningate operatsioonide täitmiseks ja on programmeerijale kättesaamatu. Summaatoris toimuvad kõik aritmeetilised, loogilised ja abioperatsioonid. Summaatori järgud on visuaal-



Joonis 1.



se jälgimise otstarbeks välja toodud signalisatsioonipaneelile.

### 1.2. Juhtseade

Juhtseade on vajalik arvuti töö juhtimiseks programmi antud käskude järjekorras. Juhtseadme käsuloendaja (koosneb 14-st kahendjärgust) abil määratakse käskude täitmise järjekord. Käsu registris on 36 kahendjärku ja arvuti töö ajal täidetavate käskude koodid dešifreeritakse registris. Käsuloendaja ja käsuregister on jälgitav signalisatsioonipaneelil.

### 1.3. Mäluseade

Mäluseadmes eristame operatiivmälu, fikseeritud mälu ja püsिमälu. Kõik mäluseadmed on ette nähtud andmete ja programmide säilitamiseks. Operatiivmälu maht on 1024 sõna; iga sõna pikkus on 36 kahendjärku. Mälu poole pöördumise tsükkel on 24  $\mu$ sek.

Fikseeritud mälus on 5 36 kahendjärgulist sõna  $a_1, a_2, a_3, a_4, a_5$ . Selle mälu pesi kasutatakse aritmeetilise seadme registrina. Programmeerija võib kasutada ainult pesa aadressiga  $a_2$ .

Püsिमälu on ette nähtud mikroprogrammide ja konstantide säilitamiseks. Püsिमälus on 16 384 sõna, kusjuures neist 2048 on 72 kahendjärku pikkusega ja kasutatakse konstantide ning mikroprogrammide säilitamiseks. Ülejäänud püsिमälu sõnade pikkus on 36 kahendjärku ja neid kasutatakse standardsete abiprogrammide säilitamiseks.

### 1.4. Sisend- ja väljundseadmed

Sisendseadmeks on arvuti elektrikirjutusmasin "Consul 254" ja perfolintsisend FS-1500. Elektrikirjutusmasina klaviatuuril on sisendkeele sümbolid, vene ja ladina tähed, numbrid, indeksid, astmenäitajad ja matemaatilised tähised. Elektrikirjutusmasinal andmeid või programme trükkides sisestatakse need samaaegselt ka arvutisse. Ülesannete vastused trükitab arvuti elektrikirjutusmasina paberilindile. See-ga on elektrikirjutusmasin ka väljundseadmeks.

Perfolintsisendiga saab varem ettevalmistatud perfolindilt informatsiooni arvutisse sisestada.

Väljundseadmeks peale eespool nimetatud elektrikirjutusmasina on veel perfolintväljund, milleks on lintperforaator П/ -80. Lintperforaatori abil võib arvutist väljastada sama-sugust informatsiooni nagu elektrikirjutusmasina abil. Peale selle võib lintperforaatorit kasutada autonoomse seadmena perfolintide ettevalmistamiseks.

### 1.5. Juhtpult ja signalisatsioonipaneel

Juhtpult on vajalik arvuti ja inimese vaheliseks suhtlemiseks. Ta koosneb operaatoripuldil ja inseneripuldil. Operaatoripuldiks on elektrikirjutusmasin ja selle laud, kus peale sisendkeele klaviatuuri on klahvid arvuti mõningate seadmete sisse- ja väljalülitamiseks ning tööolukorda viimiseks. Mõningad operaatorile vajalikud klahvid on ka inseneripuldil, neid vaatleme hiljem.

Inseneripultil kasutatakse arvuti tehnilisel hooldamisel. Ta koosneb indikatsioonipuldil ja inseneri juhtpuldil. Inseneripuldil saab arvuti ja tema seadmeid sisse ja välja lülitada ning valida arvuti töörežiime. Osa inseneripuldil on paigutatud arvuti indikatsioonipuldi ette ja on pealt kaetud liikuva paneeliga, mida võib kasutada töölauana. Signalisatsioonipuldil saab arvuti seadmete tööd jälgida visuaalselt, signaallampide abil.

## §2. ANDMETE SISESTAMINE ARVUTISSE "NAIRII-C"

Andmeid viiakse arvutisse elektrikirjutusmasina klaviatuurilt või perfolintsisendil varem ettevalmistatud perfolindil abil.

### A. Klaviatuurilt sisestamisel:

- 1) sisse lülitada klahv " ВЫВОД НА ПЕЧАТЬ "
- 2) vajutada klahvile " ПУСК 1 "

### B. Perfolintsisendil sisestamisel:

- 1) vajutada klahvile " ПУСК 1 "
- 2) sisse lülitada klahv " ВВОД С ТРАНСМИТЕРА "

Sõltuvalt sisestatavatest andmetest tuleb trükkida arvuti elektrikirjutusmasinal:

- Mk - masinakoodi käsud,
- Mr - väiksemad arvud kui arv 2,
- Mn - liikuva komaga arvud,
- Mm - täisarvud,
- Mg - kahekordse pikkusega arvud.

M - on sisseviidava massiivi algaadress. Peale igat arvu trükitakse "┐" või reavahetus (vt. joonis 2), s.t. nende abil eraldatakse arvud üksteisest. Massiivi lõpu tunnuseks on "m".

Kui osutub vajalikuks sisestada teist tüüpi andmeid, siis tuleb trükkida "m" ja jätkata nii nagu eespool kirjeldatud.

Vea korral, mis avastatakse enne "┐" või reavahetust, tuleb trükkida "x" ja see arv uuesti. Kui viga avastatakse peale "┐" või reavahetuse löömist, siis tuleb trükkida "m", valesti sisestatud arvu aadress koos tüübiga ja see arv uuesti.

Näide: sisestame kahest väiksemad (fikseeritud koma režiimis) arvud 1,5; 0,67; -1,7 alates mälupeasest 100.

100 r

1,5 ┐ 0,67 ┐ -1,7 ┐ m

Näide: sisestame täisarvud 45; 6789; 2 alates pesast 110.

110 m

45 ┐ 6789 ┐ 2 ┐ m

Näide: sisestame liikuva koma režiimis arvud alates pesast 120.

4,678

456

0

5,67

-4,6

120 n

4,678 ┐ 456 ┐ 0 ┐ 5,67 ┐ -4,6 ┐ m

Näide: sisestame pikad arvud 1234567890,0987654321; -12345,76543; 0,00; 123,0 alates pesast 130.

130 g

1234567890,0987654321 ┐ -12345,76543 ┐ 0 ┐ 123,0 ┐ m

Arvud salvestatakse pesadesse

130; 131 (1.arv) 134; 135 (3.arv)

132; 133 (2.arv) 136; 137 (4.arv)

Käsu sisestamine ja kohene täitmine. Et masin täidaks momendil sisseviidud käsu, tuleb:

trükkida k,  
trükkida käsk väliskoodis,  
trükkida täht u,  
teha reavahetus.

Näide: kui on vaja viia arv 0 pessa 650, võib teha nii:

k  
n 0 k 650 u  
reavahetus

### §3. ANDMETE VÄLJATRÜKKIMINE MÄLUST (bb režiim)

See režiim võimaldab kontrollida mälu sisu järgmiselt:

1) sisse lülitada klahv "ВЫВОД НА ПЕЧАТЬ"

2) vajutada klahvile "ВЫДАЧА ПАМЯТИ".

Arvuti vastab sellele sümbolile "bb" trükkides ja aadressi registris oleva käsu aadressi väljastades.

Sõltuvalt väljastatavatest andmetest tuleb trükkida elektrikirjutusmasinale:

1) M i n, kus M - trükitava massiivi algaadress,

i - trükkimise kuju,

n - trükitavate mälupesade arv.

i = n - siis pesa sisu trükitakse kui liikuva koma arv,

i = m - " " " " " täisarv,

i = r - " " " " " kahest väiksem murdarv,

i = k - " " " " " käsk,

i = c - " " " " " 36-järguline kahendarv,

i = g - pesade M+2i ja M+2i+1 (i = 0,1 ... N-1) sisu trükitakse pikkade arvudena.

2) Arvuti käivitamiseks teha reavahetus.

Kui on sisse lülitatud klahv "ВАРИАНТ", siis arvuti peatub peale 50. arvu trükkimist. Trükkimise jätkamiseks teha reavahetus.

Kui trükkida M i n b ja reavahetus, siis arvuti trükkib

- a) pesa aadressi kümnendsüsteemis,
- b) pesa aadressi kaheksandsüsteemis,
- c) pesa sisu kaheksandsüsteemis,
- d) pesa sisu vastavalt variandile (i), Siin i = g on keelatud.

Näiteid mälu väljatrüki kohta.

Väljastada kolm arvu alates pesast 100 fikseeritud koma režiimis.

10043	
100	1,50000000000
101	0,66999999998
102	-1,70000000001

Väljastada täisarvudena kolm arvu alates pesast 110.

110m3	
110	45
111	6789
112	2

Trükkida välja liikuva komaga arvudena viis arvu alates pesast 120.

120n5	
120	4,67799997329
121	456,00000000000
122	0,00000000000
123	5,66999995708
124	-4,60000002384

Trükkida neli arvu pikkade arvudena alates pesast 130.

130g4	
130	1234567890,09876543207
132	-12345,76543000002
134	0,00000000000
136	123,00000000000

Trükkida välja neli liikuva komaga arvu alates pesast 120.

120n4β					
120	00170	1126	6213	2003	4,67799997329
121	00171	1620	0000	0011	456,00000000000
122	00172	0000	0000	0000	0,00000000000
123	00173	1325	6050	7403	5,66999995708

## II PROGRAMMEERIMINE ARVUTIL "NAIRII - C"

Arvutit "Nairii-C" saab kasutada arvutusrežiimis klahv-arvutina ja universaalrežiimis standardsete matemaatiliste ülesannete, algoritmilises keeles programmeeritud ülesannete, masinakäskudes programmeeritud ülesannete lahendamiseks. Auto- noomse töö režiimis saab kasutada perfolintsisendit, perfo- lintväljundit ja elektrikirjutusmasinat perfolintide etteval- mistamiseks ning kontrollimiseks.

Järgnevalt vaatleme neid töörežiime lähemalt.

### § 1. ARVUTUSREŽIIM

Arvutusrežiimis saab kasutada järgmisi sümboleid:

- a) numbreid 0, 1, ..., 9
- b) tähti c, b, y, g, л, м, z, н, x, a, б, г, ч, u, o, e, l, p, m, ш, s; t, н, ж, э, й, ь, ф, ц, щ, f, я, ю, э, i, j, k, n.

Neile tähtedele võib anda suvalisi väärtusi.

M ä r k u s. Kui kasutatakse tähti ж, э, й, в, ф, ц, щ, f, я, ю, э, siis on keelatud kasutada tähti i, j, k, n.

- c) tähte  $\pi$ , mille väärtus masinas on 3,1415,
- d) täisarvulisi astmenäitajaid 0, 1, ..., 9,
- e) märke +, -, /, x, !,  $\sqrt{\quad}$ , =,  $\perp$ .

Märk " = " on masinale arvutamise alustamise korralduseks. Peale märgi " = " kirjutusmasinale löömist alustab arvuti tööd.

#### Arvuti käivitamine

1. Elektrikirjutusmasina lüliti "1-0" (üleval vasakus nurgas) lülitada asendisse "0".
2. Vajutada inseneripuldil nupule "ВКМ".
3. Lüliti "1-0" elektrikirjutusmasinal lülitada asendisse "1".
4. Elektrikirjutusmasina klahv "ВКЛ ПЕЧАТИ" sisse lülitada.

#### Arvuti väljalülitamine

1. Elektrikirjutusmasina lüliti "1-0" lülitada asendisse "0".

## 2. Vajutada "Nairii" paneelil nupule "ОТКЛ".

Arvutusrežiimis saab arvutit "Nairii-C" kasutada klahv-arvutina aritmeetiliste avaldiste ja elementaarfunktsioonide väärtuste leidmiseks. Kirjutusmasina klaviatuurilt tuleb arvutisse viia matemaatiline avaldis, millele arvuti trükkib vastuse pärast " = " löömist. Arvutusrežiimis töö toimub järgmiselt:

- 1) inseneripuldil klahv "УНИВЕРСАЛЬНЫЙ-СЧЕТНЫЙ" lülitada asendisse "СЧЕТНЫЙ";
- 2) vajutada klahvile "ПУСК 1".

Peale seda arvuti trükkib režiimi tunnuse "CC", mis tähendab, et arvuti on valmis tööks.

### 1.1. Elementaarfunktsioonide sin, cos, tg, arcsin, arccos, arctg, exp, ln, lg, $\sqrt{\quad}$ , $\sqrt[n]{\quad}$ ( $n > 2$ ; $n$ on täisarv) arvutamine

Argumentideks võivad olla arvud, tähed, tähtede ja numbrite korrutised (ilma korrutusmärgita) ja kraadimõõdus suurused (kuni märkideni + - x /  $\sqrt{\quad}$  =, millega argument lõpeb). Võib arvutada kahe elementaarfunktsiooni superpositsiooni. Nii funktsiooni kui argumenti võib astendada positiivse täisarvuga.

Kõigi funktsioonide ette tuleb trükkida  $\sqrt{\quad}$ , mis on funktsiooni tunnuseks.

#### Näited:

$$\begin{aligned} & \sqrt{\quad} \cos 2,5 = \\ & \sqrt{\quad} \cos 2,5x = \\ & \text{aga } \sqrt{\quad} \cos 2,5 \times X = \text{arvutatakse } (\cos 2,5)x \\ & \sqrt{\quad} \exp^2 \sqrt{\quad} \cos^3 \sqrt{\quad} 60^\circ = \\ & \text{aga } \sqrt{\quad} \exp^2 2 \sqrt{\quad} \cos^3 \sqrt{\quad} 60^\circ = \text{arvutatakse } (\exp^2 2) \cos^3 60 \end{aligned}$$

Pärast märki " = " trükkib masin vastuse.

a. Arvutiga saab arvutada astmeid

$$A^n = \quad ,$$

kui  $n$  on täisarv ja  $A$  on arv või täht .

b. Et arvutada  $\sqrt[n]{A}$ , trükitakse järjest

$\sqrt[n]{A}$   
n (arv n on aste, vt. joon. 1; alumine registreer trükimasinal)

A  
=  
näiteks:  $\sqrt[3]{4} =$

c. Funktsiooni astendamiseks trükitakse astmenäitaja kohe peale funktsiooni sümbolit. Näiteks  $(\sin ax)^2$  trükitakse:

$\sin^2 ax =$

d. Kui funktsiooni argument on kraadimõõdus, tuleb tema ette trükkida " $\sqrt{\quad}$ ". Kraadimärk on astmenäitaja "0", minutimärk aste "1" ja sekundimärk aste "11".

$\sqrt{\sin \sqrt{30^{\circ}15^1}} =$   
 $\sqrt{\cos \sqrt{15^{\circ}}} =$   
 $\sqrt{\sin^2 \sqrt{\cos^3 \sqrt{30^{\circ}15^1}}} =$   
 $\sqrt{\ln \sqrt{15^{\circ}}} =$

e. Üleminekuks radiaanmõõdult kraadimõõdule trükitakse arv ja märk "=".

5,6 =

f. Et teisendada kraadimõõtu radiaanmõõduks, trükitakse märk " $\sqrt{\quad}$ ", nurk ja "=".

$\sqrt{30^{\circ}15^1} =$   
 $\sqrt{3,5^{\circ}4,7^13,7^11} =$   
 $\sqrt{3456^{\circ}1234^1} =$

g. Arvude 1 ... 12 jaoks võib arvutada faktoriaali:

5 ! =

kaheteistkümnest suurema arvu puhul trükitab arvuti:

n > 12 "HE MOY"

## 1.2. Aritmeetilised tehted

Arvutil on +, -, x, /. Sulgusid ei saa kasutada. Kui avaldis algab märgiga "-", on vaja enne trükkida 0.

Näide: et korrutada arvud -4,4 ja 5 tuleb trükkida:

0 -4,4 x 5 =

Kui üks operand on täht, pole tema ette korrutusmärgi "x" vaja.

Näide:

2 a =  
ab x 4 =  
4ab =

1.3. Tehted tähtede arväärtustega

a. Tehteks arväärtustega tuleb järjest trükkida:

täht  
märk "="  
arv  
VK (reavahetus)

Näide:

a = 5      VK  
b = 7      VK

Sama arväärtust võib omistada mitmetele tähtedele. Seda tehakse näiteks nii:

a = 5      VK  
= b      (reavahetuse teeb masin ise)  
= c

b. Eelneva tehte tulemuse omistamine tähele toimub analoogiliselt, trükitakse

märk "="  
valitud täht.

Näide:

$ax^2 + b \cos x =$   
arvuti trükkib vastuse ja teeb reavahetuse;  
trükkides                      = c,  
annab arvuti tähele c väärtuseks leitud  $ax^2 + b \cos x$ .

c. Mingi tähe arväärtuse teadasaamiseks trükitakse järjest:

täht  
märk "="  
┌

Näide:

a = ┌  
ja arvuti trükkib tähe a väärtuse.

d. Iga trükitud resultaadi võib võtta järgmise operatsiooni esimeseks operandiks:

$$4 + 6/2 = 7$$

$$+ 10 = 17$$

$$- 4 = 13$$

M ä r k u s. Kui arvutada ei ole võimalik (näit.  $\sqrt{x}$  puhul  $x < 0$  või  $\arccos x$  puhul  $x > 1$ ) trükitab arvuti vea põhjuse välja ja jääb seisma. Edasiarvutamiseks tuleb vajutada klahvile " ПУСК 1 " "Nai-rii" paneelil.

Kui eksitakse grammatika vastu, tehakse reavahetus automaatselt ja arvutust tuleb kor-  
rata.

3451  
cc

$\sin 2 = 0,909297421$   
 $\ln 2 = 0,693147175$   
 $\lg \sin 2 = -0,041294041$   
 $\cos 60^\circ = 0,500000000$   
 $\sqrt{60^\circ} = 1,023326709$

x=2

a=3  
 $\sin^2 \cos^3 ax = 0,599147193$

$2^64 = \infty$

10524

cc

$23^{12} = 21914624488112128,0$

$^{12}\sqrt{23} = 1,298605740$

=a

$a^{12} = 22,999999046$

a=3

$\sin^3 ax = -0,021814812$

$\ln 15^\circ = -1,340176761$

5,6=320°51'23"11

$3,5^0 14^1 1,4^{11} = 0,065165746$

7!=5040,000000000

0-4,5x7=-31,500000000

2a=6,000000000

B=5

aB=15,000000000

4aB=60,000000000

$ax^2 + 3 \cos x = 9,919265866$

=c

c=19,919265866

4+6/2=7,000000000

+10=17,000000000

-4=13,000000000

x2=26,000000000

H=10,000000388

$\pi = 3,141592651$

$\pi = 3$

$\pi = 3,000000000$

$\sqrt{-1} = x < C$

5669

cc

$\arccos 23 = x > 1$

Keerulisemad avaldised tuleb arvutada osade kaupa.

Näide:

Kui  $y = \left\{ \cos \left[ (\sin^2 3x + 4)^2 \ln a \right] + 2a \right\} x^2 + 4$ ,  
siis võib teha nii:

(x ja a väärtused tuleb ette anda)

x=8

a=2

$\sin^2 3x+4=4,820072174$

=c

$c^2 \ln a=16,103954792$

=c

$\cos a+2a=3,077385425$

$x^2+4=10,154770851$

=y

y=10,154770851

## §2. TÕUPÜLESANNETE LAHENDAMINE

Arvuti "Nairii-C" abil on võimalik lihtsal viisil lahendada rida standardseid ülesandeid. Neile ülesannetele vastavad standardsed programmid järgmiste tähistustega:

- 1) cy-lineaarse algebralise võrrandisüsteemi lahendamine,
- 2) Aa-matriksi determinandi, omaväärtuste, omavektorite ning pöördmatriksi leidmine,
- 3) ay-algebralise võrrandi lahendamine,
- 4) gy, pk-diferentsiaalvõrrandi süsteemi lahendamine,
- 5) matriksite korrutamine,
- 6) uA-määratud integraali arvutamine.

Nende ülesannete lahendamiseks on arvuti püsimalus standardsed programmid ning arvutile on vaja teatada ainult algtinnumused ja algandmed.

Lahendamiseks tuleb:

1) insereripuldil lülitada klahv "УНИВЕРСАЛЬНЫЙ-  
"СЧЕТНЫЙ" asendisse "УНИВЕРСАЛЬНЫЙ",

2) vajutada klahvile "ПУСК 1".

Pärast seda on arvuti tööks valmis. Igal järjekordsel lahendamisel tuleb uuesti vajutada klahvile "ПУСК 1".

"ПУСК 1" viib programmi informatsiooni sisseviimise algusse; algandmed sisestatakse liikuva koma režiimis.

Arvud sisestatakse arvutisse alates pesast "k" nõnda:

trükkida kn. Seejärel arvuti teeb automaatselt reavahetuse. Trükkida a, b, ...; iga arvu vahele lüüa "tühik" või "VK". Lõpuks trükkida täht " = ". Kui mõni arv on valesti löödud enne vahemärgi löömist, siis trükkida see arv lihtsalt uuesti.

Kui veidi hiljem märkate, et eespool on mõni arv valesti löödud, trükkige massiiv lõpuni koos tähega "u". Lugege mitmes arv oli valesti löödud - olgu see m-s arv. Vigase arvu pesa address on:  $N = k + m - 1$ . Trükkida  $N_n$  ja õiged arvud ning lõppu " = ". Mitme vea korral tuleb iga vea puhul address arvutada ja siis parandada.

## 2.1. Algebraalse võrrandi lahendamine (AY)

Programm on ette nähtud võrrandi

$$a_0 x^n + a_1 x^{n-1} + \dots + a_n = 0$$

juurte leidmiseks, kus

$a_i$  ( $i = 0, 1, \dots, n$ ) on reaalarvulised kordajad

$a_0 \neq 0$

$n \leq 38$ .

Lahendamiseks trükkida trükimasinal tähed ay või 11874 k ja võrrand eespool antud kujul. Pärast klahvile "=" löömist algab lahendamine.

Näide:

$$x^3 - 3x^2 + x + 1 = 0$$

Kui võrrand ei lahendu programmis antud arvutustäpsusega, trükkib arvuti

$$\delta = 0,00000023$$

$$\delta =$$

Sel korral tuleb anda arvutile arvutuste täpsus.

Näide:

$$\delta = 0,001$$

Lihtjuured leitakse täpsusega  $\epsilon = 10^{-4}$ .

Kordsete juured täpsus on väiksem.

Kui arvuti trükkib välja märgi "\*", siis on viga sisestamisel.

Kui eksitakse mõne kordaja sisseviimisel, siis tuleb

trükkida vastav võrrandi liige õige kordajaga uuesti. Seda võib teha enne vabaliikme trükkimist.

Suurtel n väärtustel töötab programm aeglaselt, samuti, kui puudub palju madalama astme kordajaid.

Näide 1: lahendada võrrand  $x^4 - 2x^3 + 4x^2 - x + 1 = 0$ .

$$x^4 - 2x^3 + 4x^2 - x + 1 = 0$$

$$x_4 = 0,066545 + i \quad 0,533254$$

$$x_3 = 0,066545 - i \quad 0,533254$$

$$x_2 = 0,933454 + i \quad 1,609784$$

$$x_1 = 0,933454 - i \quad 1,609784$$

Näide 2: eriti aeglaselt lahendatakse võrrandeid, kus osa x astmeid puudub, näiteks  $x^5 = 1$ .

$$x^5 - 1 = 0$$

$$x_5 = -0,809017 + i \quad 0,587785$$

$$x_4 = -0,809017 - i \quad 0,587785$$

$$x_3 = 0,309017 + i \quad 0,951056$$

$$x_2 = 0,309017 - i \quad 0,951056$$

$$x_1 = 1,000000$$

## 2.2. Määratud integraali arvutamine (ИЛ)

Programm on ette nähtud määratud integraali arvutamiseks

$$\int_a^b f(x) dx$$

Simpsoni meetodil täpsusega  $\epsilon$ ; ( $\epsilon > 10^{-8}$ ).

Lahendamiseks tuleb arvutisse viia rajad a, b ja täpsus alates mälupesast 70 ning funktsioon f(x). Trükitakse 70 n ja **üksteisest** tühikutega "┌" eraldatud rajad ja täpsus.

Edasi trükitakse "ИЛ" või "7996u" ja funktsioon. Funktsiooni trükkimisel tuleb juhendada automaatprogrammeerimise režiimi valemite trükkimise eeskirjadest. Pärast reavahetust alustab arvuti tööd.

**M ä r k u s.** Kui arvuti trükitab "HE MOFY" kohe pärast f(x) sisestamist, tähendab see, et f(x) on liiga pikk avaldis. Siis tuleb integraal arvutada osade kaupa.

Kui masin trükitab "HE MORY" pärast arvutamist, siis meetod ei koondu.

Funktsioon  $f(x)$  ei või sisaldada üle 125 arvulise konstandi (v.a. astmenäitajad).

Näide: arvutame

$$\int_0^3 (x^2 + \sin x) dx$$

70n  
0 3 0,001ш  
ИЛ  
 $x^2 + \sin x$   
10,98999345

### 2.3. Lineaarse algebralise võrrandisüsteemi lahendamine ( $n \leq 28$ )(CY)

Olgu antud süsteem:

$$a_{11}x_1 + \dots + a_{1n}x_n = b_1$$

-----

$$a_{n1}x_1 + \dots + a_{nn}x_n = b_n.$$

Lahendamiseks tuleb arvutisse viia süsteemi kordajad alates mälu 64 osast ja võrrandisüsteemi järk. Mälupesa algaadress antakse kujul 64 n ja pärast seda trükitakse kordajad  $a_{21}$ ,  $a_{12}$ , ...,  $a_{1n}$ ,  $b_1$  pärast sedatrükitakse  $a_{21}$ ,  $a_{22}$ , ...,  $a_{2n}$ ,  $b_2$ , ...,  $a_{n1}$ , ...,  $a_{nn}$ ,  $b_n$  (kordajad eraldatakse üksteisest tühikuga " ").

Pärast seda trükitakse tähed CY või 7688 u ja võrrandi järk n =

Pärast reavahetust alustab arvuti tööd. Kui arvuti trükitab "HE MORY" siis  $n > 28$ .

Kui arvuti trükitab "PEИИ HET", siis süsteem ei ole (ül-diselt) lahenduv.

Kui arvuti trükitab "HE COBИ", on tegemist juhusliku veaga ja lahendamist tuleb korrata.

### 2.4. Maatrikseid käsitlevad programmid ( $n \leq 12$ )

1. Maatriksi determinandi arvutamine (BO)
2. Pöördmaatriksi leidmine (OM)
3. Omaväärtuse leidmine (C4)
4. Omavektorite leidmine (CB)

Sõltuvalt sellest, mida arvutatakse, tuleb kasutada järgmisi tähiseid või aadresse

- BO - 11312
- OM - 11314
- CH - 11316
- CH, CB - 11318
- BO, OM, CH, CB - 11320
- BO, OM - 11322
- BO, CH - 11324
- OM, CH - 11326
- BO, CH, CB - 11328
- OM, CH, CB - 11330

Näide:

Maatriksi determinandi ja omaväärtuste arvutamise programmile vastab aadress 11324 u.

Lahendamiseks viiakse alates arvuti mälupeasast 61 sisse lähtemaatriks ridade kaupa. Peale seda trükitakse tähised või aadressid olenevalt sellest, mida soovitakse arvutada ja maatriksi astak  $n = \dots$

Pärast reavahetust alustab arvuti tööd.

Vastustes trükitakse iga arvutatud suuruse ees tema lühendatud nimetus

- ОПРД - (determinant)
- ОБР МАТР - (pöördmaatriks)
- СОБСТ ЧИСЛО - (omaväärtused)
- СОБСТ ВЕКТОР - (omavektorid)

omaväärtused trükitakse välja kujul

$$x_1 = \dots$$

$$x_n = \dots$$

Iga omavektori ette trükitakse sümbol  $\beta_i$ , kui  $i$  on arvuline indeks.

M ä r k u s e d. 1. Omaväärtuse ja omavektorite indeksid ei tarvitse ühtida. Et nende vastavust selgitada tuleb väljatrükkimise režiimis "bb" välja trükkida  $2n$  pesa liikuva koma režiimis alates pesast 838, kus paiknevad omaväärtused (igauks kahes kõrvuti olevas pesas). Omavektorid trükitakse

välja selles järjekorras, nagu nad on selles massiivis.

2. Kui mõni vajalik omavektor osutub nullvektoriks, tuleb lahendada ülesanne uuesti, kusjuures enne arvutamist lülitakse sisse klahv "ВАРИАНТ". Siis trükitakse iga  $\beta_i$  ( $i = 1, \dots, n$ ) järel välja  $n^2$  arvu, mis moodustavad  $n$  omavektorit antud omaväärtuse jaoks.
3. Kui matriksi pööramisel lähtemaatriks on kõdunud, trükitakse "МАТР. КОСВ."
4. Kui  $n > 12$ , trükitab masin "n > 12" HE MOFY" ja jääb ootama uut järku.

Näide: arvutame kõik võimalikud suurused matriksi

$$\begin{vmatrix} & 2 & & 45 & & 6 \\ & 4,6 & & -456 & & -7 \\ 567,7 & & & 34 & & 12 \end{vmatrix}$$

jaoks tabelist  $N = 11320$

61n  
2 45 6 4,6 -456 -7 567,7 34 12 w

11 320u

n=3

ОПРД = 1 362 388,10937500

ОБР МАТР

-0,00384178

-0,00024662

0,00177702

-0,00295738

-0,00248255

0,00003053

0,19012761

0,01870135

-0,00082135

СОБСТ ЧИСЛА

$x_3 = -47,719347$

$x_2 = 62,502358$

$x_1 = -456,783012$

СОБСТ ВЕКТ

$B_1$

605,06286621

-6130,30175781

-288,11523437

$B_2$

-24144,25415039

-4248,60894775

231937,32617187  
 $B_3$   
 26423,59179687  
 -3741,58911132  
 294510,18750000

BB  
 981  
 838n6  
 838 -456,78301239013  
 839 0,00000000000  
 840 -47,71934700012  
 841 0,00000000000  
 842 62,50235843658  
 843 0,00000000000

## 2.5. Maatriksite korrutamine

Programm korrutab nii ruut- kui ristkülikmaatrikseid. Viimaste jaoks tähistatakse

$n$  - 1. maatriksi ridade arv

$m$  - 1. maatriksi veergude arv

$p$  - 2. maatriksi veergude arv.

Lahendamiseks viiakse alates arvuti mälupeasast 51 sisse esimene ja teine maatriks ridade kaupa. Pärast seda trükitakse "2904u" ja ruutmaatriksite korral järk  $n = \dots$  ja tühik. Ristkülikmaatriksite korral tuleb trükkida  $n = \lfloor \dots \rfloor m = \dots \lfloor p = \dots \rfloor$ .

Näiteks olgu  $n = 5$ ;  $m = 3$ ;  $p = 1$ , siis

trükitakse  $n = \lfloor 5 \rfloor m = 3 \lfloor p = 1 \rfloor$ .

Arvud  $n$ ,  $m$  ja  $p$  peavad rahuldama võrratusi.

Ruutmaatriksite korrutamisel  $n \leq 17$ ,

ristkülikmaatriksite korrutamisel  $nm + mp + pn \leq 900$ .

## 2.6. Esimest järku diferentsiaalvõrrandite süsteemi lahendamine

Olgu antud süsteem

$$y_i' = f_i(x, y_1, \dots, y_n) \quad i = 1, \dots, n.$$

Süsteem lahendatakse Runge-Kutta meetodiga. Sammu valik võib olla automaatne (siis  $n \leq 17$ ) või ette antud (konstantne); siis  $n \leq 21$ . Võrrandites tuleb asendada  $y_1 \dots y_n$  tähtedega järgmises järjekorras:

c, b, y, g, л, M, z, H, x, a, δ, Γ, ч, u, o, e, l, p, m, ш, s.

Argument  $x$  asendatakse tähega  $t$ .

Saadakse seega:

$$c' = f_1(t, c, b, y, \dots)$$

$$b' = f_2(t, c, b, y, \dots)$$

$$y' = f_3(t, c, b, y, \dots)$$

Näide:

$$y_1 = y_2 + \sqrt{y_1/2} + 2x$$

$$y_2 = y_1/2y_2,$$

asendades saame

$$c' = b + \sqrt{c/2} + 2t$$

$$b' = c/2b.$$

Lahendamiseks tuleb viia arvuti pesadesse:

- 66n - täpsus (automaatse sammu valiku puhul),
- 78n - integreerimise samm. Kui automaatne sammu valik, siis esimene samm võrdub trükkimise sammuga,
- 79n - integreerimisvahemiku lõpp,
- 80n - argumenti  $t_0$  algväärtus,
- 81n + 80n - lahendite väärtused punktis  $t_0$ ,
- 197n - lahendite trükkimise samm.

Edasi tuleb trükkida vastavalt:

a) 7352n või "pk" (konstantne samm)

b) 7354u või "gy" (automaatne samm)

- trükkige võrrandite arv (n)

- "VK"

- kui masin trükkib  $c' =$  ,  $b' =$  , ...,

siis anda funktsioonid

$$f_1(t, \dots)$$

$$f_2(t, \dots)$$

ja reavahetus pärast igat funktsiooni. Kui mõni funktsioon on valesti loodud, anda "ПУСК 1" ja "pk" või "gy".

NB! Need funktsioonid trükkida nagu "АП" -s.

**M ä r k u s e d.** Kui arvuti trükkib  $n > 21$  (const. samm) või  $n > 17$  (autom. samm), siis tähendab, et võrrandeid on palju.

Kui arvuti trükkib "HE MOFY", siis see tähendab, et funktsioonid  $f_1, f_2, \dots$  on liiga mahukad.

Funktsioonides  $f_1, \dots$ , võib olla kokku mit-  
te üle 125 arvulise konstandi (v.a. astme-  
näitajad).

Lahendamisel säilivad arvutis funktsioo-  
nid  $f_1, f_2, \dots$ . Sama süsteemi teiste alg-  
andmetega uuesti integreerimiseks viiakse  
need lihtsalt sisse ja trükitakse 7428u.

Kui arvuti hakkab samu  $t$  väärtusi väl-  
ja trükkima, siis tähendab see, et ta ei  
suuda ülesannet lahendada.

Et arvuti trükiks tulemused välja ai-  
nult nõutud punktides, selleks lülitatakse  
sisse klahv "ВАРНАНТ".

Näide: Lahendame DV süsteemi

$$y_1' = y_2 + \sqrt{\frac{y_2}{2}} + 2x$$

$$y_2 = y_1/2y_2^2$$

algtingimustel  $x_0 = 0,1$ ;  $y_1^{(0)} = 0,02$ ;  $y_2^{(0)} = 0,1$ ; täpsusega  
üks tuhandik.

Lahendid tuleb välja trükkida vahemikus  $(0,1; 1)$  sammuga  $0,2$ .  
Sammu valik automaatne.

Et arvuti trükiks tulemused ainult nõutud punktides, vajuta-  
me alla klahvi "ВАРНАНТ".

```

66n
0,001m
78n
0,1 1 0,1 0,02 0,1m
197n
0,2m
87
n=2
c1 = B + √c/2+2t
B1 = c/2β2
t=0,30000000
c=0,17999652
B=0,29998714
t=0,50000000
c=0,49999404
B=0,49999334
t=0,69999998
c=0,97999233
B=0,69999521
t=0,89999997
    
```

c=1,61999095  
B=0,89999608  
t=1,00000000  
c=1,99999046  
B=0,99999639

M ä r k u s.

Selle süsteemi lahend on

$$\begin{cases} y_1 = 2x^2 \\ y_2 = x \end{cases}$$

### §3. AUTOMAATPROGRAMMEERIMINE

#### 3.1. Üldkirjeldus

Masinakäskudes programm on kiirem, kuid teda on raske kirjutada. Sageli võib käskudes programmi tegemise usaldada arvutile. Selleks kirjutatakse ülesande lahendamise programm üles algoritmilises keeles ja viiakse arvutisse. Programmeeriva programmi - translaatori - abil koostab arvuti käskudes programmi - tööprogrammi - ja kirjutab selle operatiivmallu.

Edasises kasutame arvuti "Nairii" algoritmilise keele nimetust "AP-keel" või lihtsalt "AP" (automaatprogrammeerimine). Programmeerivat programmi nimetame AP-translaatoriks. Tuleb arvestada, et AP-programm ei ole optimaalne - ta on pikem käsitsi tehtud programmist ning ei kasuta mälu efektiivselt. AP-keele loogilised võimalused on piiratud. Siiski võib lühemaid ja loogiliselt lihtsamaid ülesandeid selles keeles lahendada. Märgitud puudusi korvab osaliselt võimalus lülitada AP-programmi käskudes kirjutatud programmilõike.

Enamikku eespool kirjeldatud standardprogramme ei saa kasutada koos AP-programmidega.

Järgnevalt antakse AP-keele kirjeldus ja programmeerimise reeglid.

AP-s kasutatakse järgmisi teletaibi sümboleid:

arvud 0, 1, 2, 3, 4, 5, 6, 7, 8, 9;

väikesed, reaajoone suhtes tõstetud arvud astmenäitaja trükkimiseks - 0, 1, 2, 3, 4, 5, 6, 7, 8, 9;

väikesed, reaajoone alla nihutatud arvud indeksite trükkimiseks - 0, 1, 2, 3, 4, 5, 6, 7, 8, 9;

tähed (AP tähestiku järjekorras) -  $\pi, c, \beta, \gamma, \vartheta, \pi, M, z, H,$   
 $x, a, \sigma, \Gamma, \varphi, u, o, e,$   
 $l, p, m, \text{III}, s, t, bl, i, j,$   
 $k, n.$

Reajoone alla nihutatud tähed täheliste indeksite trükkimiseks - i, j, k;  
 operatsioonide märgid +, -, /,  $\sqrt{\quad}$ ;  
 suhete märgid =,  $\neq$ , >,  $\geq$ , <,  $\leq$ ;  
 sulud ( );  
 absoluutväärtuse märk | ;  
 tühik  $\square$  ;  
 teletaibi kelgu tagasiviimine ja reavahetus  $\leftarrow$  .

### Standardfunktsioonid.

AP-s on olemas järgmised standardfunktsioonid:

$\sqrt{A}$ , lg A (kümnendlogaritm), ln A (natüraallogaritm),  $e^A$ , sin A, cos A, tg A, arcsin A, arccos A, arctg A. Siin A on funktsiooni argument (lähemalt funktsiooni argumendist vt. allpool). Programmide üleskirjutuses on standardfunktsioonide sümboliteks järgmised tähestiku märkide kombinatsioonid:  
 $\sqrt{\quad}$ ,  $\square$  exp,  $\square$  lg,  $\square$  ln,  $\square$  sin,  $\square$  cos,  $\square$  tg,  
 $\square$  arcsin,  $\square$  arccos,  $\square$  arctg.

NB! Programmi viimisel arvutisse peab iga standardfunktsiooni sümboli (välja arvatud  $\sqrt{\quad}$ ) ette trükkima tühiku ( $\square$ ), mis on funktsiooni tunnuseks. Ilma tühikuta võtab arvuti trükitud tähti nagu muutujate korrutist. Näiteks üleskirjutus arccos transleeritakse sel juhul korrutiseks a . r . c . c .  
 . o . s .

### Fikseeritud ja liikuva koma arvud

Arvulised suurused võivad programmis olla kas konkreetsete arvude või tähtede kujul. Nimetame arvuna kirjapandud suurust konstandiks, tähena kirjapandud suurust aga muutujaks - sõltumata sellest, kas tema väärtus tegelikult programmi jooksul muutub. .

Konstandid ja muutujate väärtused võivad AP-s olla kahte tüüpi: reaalarvud (real) ja täisarvud (integer). Vastavalt sellele eristame reaalarvulisi ja täisarvulisi konstante ja muutujate väärtusi. Tähtede kasutamine eriti tüüpi muutujate jaoks on AP-s rangelt fikseeritud. Reaalarvulisi muutujaid võib tähistada järgmiste tähtedega:

$\pi$ , c,  $\beta$ ,  $\gamma$ , A, J, M, z, H, x, a, O, T, q, u, o, e, l, p, m, III, s, t, H.

Täisarvuliste muutujate jaoks kasutatakse tähti

Reaalarvud kujutatakse masinas liikuva koma arvudena. Nad salvestatakse ligikaudselt ja nendega teostatavate tehetulemus on samuti ligikaudne. Täisarvud kujutatakse masinas täpselt. Täisarvulised muutujad on näiteks: indeksid, astmenäitajad, operaatorite numbrid (märgendid), tsüklihoendajad.

### 3.2. Avaldised

AP edasiseks kirjeldamiseks on vaja täpsustada mõnede sõnade tähendust.

Teguriks nimetame arvu, muutujat, standardfunktsiooni või suvalist ümarsulgude sees asuvat aritmeetilist avaldist (vt. allpool), samuti nimetatud avaldise positiivset täisarvulist astet.

Üksliikmeks on üks tegur, või tegurite hulk, mis on üksteisest eraldatud jagamis- ja korrutamismärkide abil. Teiste sõnadega, üksliige on algebraline avaldis, milles viimasena täidetakse üks järgmistest operatsioonidest: korrutamine, jagamine, astendamine, juurimine, standardfunktsiooni arvutamine.

Aritmeetiline avaldis on üksliige või üksliikmete jada, mis on teineteisest eraldatud pluss- või miinusmärkide abil.

Valemiks nimetame üleskirjutust:

reaalarvulist tüüpi muutuja = aritmeetiline avaldis.

Programmide kirjutamisel AP-s peab meeles pidama, et standardfunktsiooni argumendiks on funktsiooni sümboli järel kirjapandud üksliige. Samuti on jagajaks üksliige, mis kirjutatakse peale jagatismärgi /. Seega avaldistes

$\exp x$   $\cos x$   $\exp x + a$  ja  $\sqrt{(a+b)/c} + xy$   
arvutab masin vastavalt:

$$e^{a \cos x} + e^x + a \quad \text{ja} \quad \sqrt{\frac{a+b}{c}} + xy.$$

#### Aritmeetiliste avaldiste kirjutamise reeglid

Aritmeetiliste avaldiste kirjutamisel peab arvestama järgmisi reegleid.

1. Aritmeetilistesse avaldistesse võivad kuuluda reaalarvulised muutujad, AP-s lubatud standardfunktsioonid, märgid

+, -, /,  $\sqrt{\quad}$ , ümarsulud, astmenäitajad, muutujad täheliste indeksitega i j k.

2. Korrutamismärki ei kirjutata. Seepärast peab kahe arvu korrutamiseks ühe neist panema sulgudesse, näiteks  $2,75 \cdot 8,94$  kirjutatakse  $3,75 (8,94)$ .

3. Kasutatakse vaid ümarsulge.

4. Kõik avaldised kirjutatakse ühes reas. Jagamismärgiks on "/". Seepärast, kui lugeja ja nimetaja ei ole üksliikmed, peab need panema sulgudesse, näiteks avaldis  $\frac{a+b}{x+y}$  tuleb kirjutada kujul  $(a+b)/(x+y)$ . Kui avaldis on pikk ja ei mahu ühte ritta, tuleb teletsibi kelk käsitsi tagasi lükata ja keerata uus rida.

5. Juurealune avaldis on üksliige, mis seisab märgi  $\sqrt{\quad}$  järel. Kui juurealune avaldis ei ole üksliige, siis asetatakse ta sulgudesse. Näiteks, avaldis  $\sqrt{ax+b}$  kirjutatakse üles kujul  $\sqrt{(ax+b)}$ .

6. Kune standardfunktsiooni argumendiks on üksliige, peab mitme funktsiooni korrutamisel kõik funktsioonid peale viimase panema sulgudesse.

Näide:

avaldis  $\sin x \cdot \cos y \cdot \lg z$  tuleb kirjutada  $(\sin x)(\cos y) \lg z$

Avaldised

$$\frac{y \cdot \sin a \cdot \lg b}{r},$$

$$a + \frac{ab+cy}{3y^2} + 50 \frac{a^2+y^2}{\frac{a}{b} + cy},$$

$$\frac{\pi}{2} + \arcsin \frac{3 \sin at}{\sqrt{25 + 9 \sin^2 at + (3 \cos at - 10)^2}} + \sqrt{5,1} \cos 2at$$

tuleb üles kirjutada nii:

$$y(\sin a)(\lg b)/r$$

$$a + (ab+cy)/3y^2 + 50(a^2+y^2)/(a/b+cy)$$

$$0,5\pi + \arcsin \frac{3 \sin at}{\sqrt{(25+9(\sin at)^2 + (3 \cos at - 10)^2)}} + (\sqrt{5,1}) \cos 2at.$$

7. Astmenäitajate kirjutamiseks kasutatakse väikesi tõstetud arve. Astmenäitajaks võib olla vaid naturaal arv. Avaldis  $A^p$ , kus p on murdarv või aritmeetiline avaldis, tuleb esitada

kujul:

$\lfloor \exp \lfloor \ln A \rfloor$

Näiteks, avaldised

$\sin \sqrt[3]{2,1+x}$  ja  $(x+a)^{2,5(x+b)}$

tuleb programmeerida kujul:

$\lfloor \sin \lfloor \exp (1/3) \lfloor \ln (2,1+x) \rfloor$  ja

$\lfloor \exp 2,5 (x+b) \lfloor \ln (x+a) \rfloor$ .

### 3.3. Operaatorid

Operaatoriks nimetame AP-keele sümbolite jada rea algusest kuni reavahetuseni ( $\leftarrow$ ). AP-s kirjutatud programm koosneb üksteise alla kirjutatud loogiliselt seotud operaatoritest. Iga operaatori ees on järjekorranumber. Pärast järjekorranumbrit trükitakse tühik ning siis operaatori nimetus. Nimetusele järgneb operatsioonide jada, mis täidetakse operaatori sees - operaatori sisu. Operaator lõpeb reavahetusega. Kui programmi sisseviimisel kirjutusmasinalt mõne operaatori sisu ei mahu ühele reale, tuleb viia kirjutusmasina kelk käsitsi tagasi ja keerata rida edasi, kasutamata seejuures klahve. Seega kirjutatakse operaator üles järgmiselt:

Operaatori number  $\lfloor$  operaatori nimetus, operaatori sisu, reavahetus.

Operaatori järjekorranumbriteks on naturaalarvud 1...132. Masin täidab operaatoreid nende numbrite kasvamise järjekorras. Seda järjekorda võib muuta juhtimisoperaatorite abil. Programmi täitmist võib alata suvalisest operaatorist.

Operaatorite nimetusena on AP-s kasutamisel järgmised seitseteist sõna:

ДОПУСТИМ	- omistame
ВЫЧИСЛИМ	- arvutame
ВСТАВИМ	- muudame
ВВЕДЕМ	- sisestame
РЕШИМ	- lahendame
ПЕЧАТАЕМ	- trükime
ИНТЕРВАЛ	- intervall
ИДТИ К	- minna
ЕСЛИ	- kui
ПРОГРАММА	- programm

КОНЧАЕМ - lõpetame  
 СПРОСИМ - küsime  
 ОСТАНОВ - peatus  
 ХРАНИМ - säilitame  
 НАЧЕРТИМ - joonestame  
 МАССИВ - massiiv  
 ИСПОЛНИМ - täidame

Operaatori programmeerimisel ja sisseviimisel on vaja rangelt kinni pidada operaatori kirjutamise reeglitest - AP süntaksist. Vigaseid operaatoreid arvuti ei transleeri, vaid trükitab vea tunnuse - täрни ja teeb reavahetuse. Siis võib operaatori uuesti sisestada.

### 3.4. Operaatorite kirjeldus

Edaspidi tähistab täht M operaatorite üleskirjutust selgitavates näidetes, operaatori järjekorranumbrit (märgist).

Operaator "ДОПУСТИМ" - omistamisoperaator.

Operaatori sisuks on tühikutega teineteisest eraldatud võrduste  $A = B$  jada. Võrduste arv operaatoris ei ole piiratud. Vasakuks ja paremaks pooleks võrduses võivad olla konstandid, muutujad või suvalise pesa sisu (selle aadress antakse siis sulgudes). Lubatud on järgmist tüüpi võrdused:

täisarvuliste muutujate jaoks (i, j, k, n):

täht =  $\begin{cases} \text{arv} \\ \text{täht}, \end{cases}$

reaalarvuliste muutujate jaoks:

täht või (aadress) =  $\begin{cases} \text{arv} \\ \text{täht} \\ |\text{täht}| - (\text{tähe absoluutväärtus}) \\ -|\text{täht}| - \text{tähe absoluutväärtus} \\ \text{(aadress)} \quad \text{miinusemärgiga} \end{cases}$

#### Näide

M  $\lfloor$  ДОПУСТИМ  $a=2,75 \lfloor x=y \lfloor i=2 \lfloor n=2$  reavahetus

M  $\lfloor$  ДОПУСТИМ  $c=(270) \lfloor (271)=0 \lfloor (950)=(280)$

Ühes võrduses ei tohi olla eri tüüpi muutujaid. Võrdustes  $A = B$  ja  $A = -B$  ei tohi mõlemal pool võrdusmärki olla sama täht. Et muutujale x omistada tema absoluutväärtust, võib talitada nii:

M ⊂ ДОПУСТИМ  $a=|x|$  ⊂  $x=a$  reavahetus, või

M ⊂ ДОПУСТИМ  $a=-|x|$  ⊂  $x=a$  reavahetus.

### Операатор " ВЪЧИСЛИМ "

Selle operaatori sisuks on võrdus. Võrduse vasakul pool on reaalarvuline muutuja (võib olla indekseeritud), paremal pool suvaline aritmeetiline avaldis. Kui programmis kasutatakse mitu korda operaatorit " ВЪЧИСЛИМ ", mille sisuks on seesama võrdus, võib programmi üleskirjutamist lihtsustada. Selleks kirjutatakse võrdus programmi alguses enne operaatoreid ilma järjekorranumbrita. Võrduse vasakuks pooleks peab siis olema indeksita muutuja, näiteks  $y$ . Vastavasse programmi kohta kirjutatakse operaator:

M ⊂ ВЪЧИСЛИМ  $y$  reavahetus.

Vt. ka §3 näide nr.4.

### Операатор " ВСТАВИМ " - sammoperaator

Операatori sisuks on suvaline arv võrdusi.

$$A = A \pm B,$$

kus  $A$  on täisarvuline või reaalarvuline muutuja ja  $B$  on vastavat tüüpi arv. Võrduste vahele kirjutatakse tühik. Operaatori täitmisel saab muutuja  $A$  uue väärtuse, mis on esialgselt arvu  $B$  võrra suurem (väiksem).

### Näide

M ⊂ ВСТАВИМ  $x=x+2,5$  ⊂  $i=i+1$  ←

Selles operaatoris ei tohi kasutada indeksiga muutujaid.

Masinat soovitatakse kontrollida järgmise lihtsa programiga:

1 ⊂ ДОПУСТИМ  $x=3$  ⊂  $i=3$  ⊂  $n=0$  ←

2 ⊂ ВСТАВИМ  $x=x-1$  ⊂  $i=i-1$  ⊂  $n=n+1$  ←

3 ⊂ ПЕЧАТАЕМ С 3 ЗНАКАМИ  $x$  ⊂  $i$  ←

4 ⊂ ЕСЛИ  $n \leq 6$  ⊂ ИДИТИ К 2 ←

5 ⊂ КОНЧАЕМ

ИСПОЛНИМ 1. ←

### Операатор " ВВЕДЕМ "

Операatori sisuks on reaalarvuliste muutujate nimekiri, mis on üles kirjutatud järgmiselt:

M ⊂ ВВЕДЕМ  $x$  ⊂  $y$  ⊂ ...  $z$  ←

vt. §3 näide nr.3.

Операatori täitmisel peatub masin lugemiskäsul (puldil

süttib lamp "КОМАНДА ВВОД"). Järgnevalt tuleb trükkalilt või transmitterilt viia sisse muutujate arvulised väärtused samas järjekorras, nagu need on antud nimekirjas. Peale igat arvu antakse tühik või tehakse reavahetus. Peale viimase arvu sisseviimist ja reavahetust läheb masin automaatselt üle uue operaatori täitmisele. Selle operaatoriga on mugav sissestada arve ülesande lahendamise käigus.

#### Operaator "ПЕЧАТАЕМ"

Operaatorit kasutatakse tulemuste väljatrükkimiseks.

Operaatori kuju on järgmine:

$M \sqcup \text{ПЕЧАТАЕМ} \text{ } C \text{ } K \text{ } \text{ЗНАКАМИ} \text{ } A \sqcup B \sqcup \dots \text{ } C \leftarrow$ ,  
 kus  $A, B, \dots, C$  on suvalised muutujad, aga  $k = 0, 1, \dots, 9$  näitab, mitme kohaga need muutujad välja trükitakse. Täisarvulised muutujad trükib masin välja ilma komakohtadeta.

#### Näide

$M \sqcup \text{ПЕЧАТАЕМ} \text{ } C \text{ } 2 \text{ } \text{ЗНАКАМИ} \text{ } x \sqcup a \sqcup i \leftarrow$

Masin trükib  $x$  ja  $a$  kahe kohaga peale koma ja  $i$  väärtuse täisarvuna.

$$x = 5,76$$

$$a = 6,93$$

$$i = 0.$$

Täisarvulised muutujad on soovitatav trükkida rea lõppu.

Operaator "ИНТЕРВАЛ N" jätab trükkalil  $N$  rida vahele.

Operaator "ИДТИ К N" on tingimusteta ülemineku operaator. Masin läheb operaatori  $N$  täitmisele.

Operaator "ЕСЛИ ... ИДТИ К" on tingimusliku ülemineku operaator. Peale sõna "ЕСЛИ" kirjutatakse mingi tingimus ja tühik, peale sõna "ИДТИ К" trükitakse operaatori

number, mille täitmisele tuleb minna, kui tingimus kehtib.

Kui ta ei kehti, läheb masin järgmise operaatori täitmisele.

Tingimus võib olla järgmist tüüpi:

$$A = B, A \neq B, A > B, A \gg B, A < B, A \leq B.$$

Vasakuks pooleks  $A$  võib olla täisarvuline või reaalarvuline indeksita muutuja. Paremaks pooleks  $B$  võib olla sama tüüpi muutuja kui  $A$ .

#### Näide

$11 \sqcup \text{ЕСЛИ} \text{ } a < 3,7 \text{ } \text{ИДТИ} \text{ } K \text{ } 3 \leftarrow$ .

Kui  $a < 3,7$  minnakse operaatori 3 täitmisele, kui  $a \gg 3,7$ , minnakse operaatorile 12.

$4 \sqcup \text{ЕСЛИ} \text{ } i \neq 10 \text{ } \text{ИДТИ} \text{ } K \text{ } 7 \leftarrow$ .

Kui  $i \neq 10$ , minnakse operaatorile 7, vastupidisel juhul operaatorile 5.

15  $\sqsubset$  ЕСЛИ  $x=y$   $\sqsubset$  ИДТИ К 9  $\leftarrow$ .

Kui  $x=y$ , minnakse operaatorile 9, vastupidisel juhul operaatorile 16.

Reaalarvuliste muutujate puhul tuleb ettevaatlikult kasutada tingimusi

$=, \neq, \gg, \ll$ .

Nimelt on reaalarvulised muutujad masinas ligikaudselt kujutatud ja operatsioone nendega teostatakse ligikaudselt. Ranged võrdused või võrratused võivad mitte kehtida, isegi kui muutujad saavutavad programmis ettenähtud väärtuse.

#### Näide

olgu vaja arvutada  $f(x)$  väärtustel  $x = 0; 0,3; 0,6; 0,9; 1,2; 1,5$ . Seda võib teha näiteks järgmise programmi järgi:

```
1  $\sqsubset$  ДОПУСТИМ  $x = 0 \leftarrow$   
2  $\sqsubset$  ВЫЧИСЛИМ  $y = f(x) \leftarrow$   
3  $\sqsubset$  ПЕЧАТАЕМ С 5 ЗНАКАМИ  $x \sqsubset y \leftarrow$   
4  $\sqsubset$  ВСТАВИМ  $x = x + 0,3 \leftarrow$   
5  $\sqsubset$  ЕСЛИ  $x < 1,51 \sqsubset$  ИДТИ К 2  $\leftarrow$   
6  $\sqsubset$  КОНЧАЕМ
```

Oleks aga ebaõige selles programmis kasutada tingimuslikku operaatorit

```
5  $\sqsubset$  ЕСЛИ  $x \neq 1,5 \sqsubset$  ИДТИ К 2  $\leftarrow$ 
```

Tõepoolest, arv 0,3 kirjutatakse kahendsüsteemis lõpmatu perioodilise kahendmurruna ja ei ole arvutis antud täpselt. Liites arvu 0,3 viis korda, ei saa me täpselt 1,5, tingimus  $x \neq 1,5$  täitub alati.

#### Operaator "ПРОГРАММА N"

Selle operaatori abil võib operaatorprogrammile juurde lülitada käskudes kirjutatud programmilõike ning pöörduda tagasi AP-programmi järgmisele operaatorile. Naturaalarv N on alamprogrammi alusaardess. Operaatorite "ПРОГРАММА" arv ei ole piiratud. Tagasipöördepesa number on 66, seega peab iga alamprogrammi lõpus olema käsk u 66 n.

Operaator "КОНЧАЕМ" peatab masina.

Operaator " СПРОСИМ " annab programmeerijale võimaluse omal valikul muutujaid välja trükkida. Peale operaatorit mingit sisu ei kirjutata, s.t. operaatori kuju on järgmine:

M  $\sqsubset$  СПРОСИМ

Kelgu tagasiviimine toimub automaatselt peale sõna " СПРОСИМ "

Operaatori täitmine sõltub klahvi "ВАРИАНТ" asendist. Kui ta ei ole alla vajutatud, jäetakse operaator " СПРОСИМ " vahele. Kui ta on alla vajutatud, trükkib arvuti operaatori numbri, kaks tähekest ja peatub. Puldil põleb lamp "КОМАНДА ВВОД". Siis tuleb anda muutuja (täht või indekseeritud täht), mille väärtust tahetakse teada ja võrdusmärk. Arvuti trükkib muutuja väärtuse ja seiskub uuesti lugemiskäsul. Nii- viisi võib teha kuitahes palju kordi.

Edasi lahendamiseks, s.o. järgmisele operatsioonile üleminekuks, tuleb tuua klahv "ВАРИАНТ" algseisu ja teha reava- hetus.

Operaator " ОСТАНОВ ". Selle operatsiooniga võib seisa- tada masina programmi sees. Lahendamise jätkamiseks tuleb vajutada klahvile " ПУСК 2 ".

Operaatorid "ХРАНИМ" ja "НАЧЕРТИМ".

AP-s on võimalik trükkida ühe või kahe funktsiooni graa- fikuid. Graafikuid trükitakse eelnevalt väljaarvutatud ja operatiivmälus säilitatavate funktsiooni väärtuste järgi. Funktsiooni väärtuste hoidmiseks on operaator "ХРАНИМ".

Operaator kirjutatakse tsükli sisse peale operaatoreid, mis säilitatavaid funktsiooni väärtusi arvutavad. Kahe funktsiooni graafikule ehitamisel tuleb nende väärtused arvutada ühes tsüklis ja säilitada ühe operaatoriga "ХРАНИМ". Опе- раатор kirjutatakse üles nii:

ühe funktsiooni väärtuste hoidmiseks

M  $\sqsubset$  ХРАНИМ    n  $\sqsubset$  A  $\leftarrow$  ;

kahe funktsiooni väärtuste hoidmiseks

M  $\sqsubset$  ХРАНИМ    n  $\sqsubset$  A  $\sqsubset$  B  $\leftarrow$  ;

kus A ja B on muutujate tähistused ja n punktide hulk, mil- les funktsioon on arvutatud.

Näide olgu funktsioonide  $y = f_1(x)$  ja  $z = f_2(x)$  väärtused arvutatud lõigul  $[a, b]$  sammuga h. Siis

$n = \left[ \frac{b-a}{h} \right] + 1$ , kus  $\left[ \frac{b-a}{h} \right]$  on avaldise  $\frac{b-a}{h}$  täisosa. Olgu  $a = 1$ ;  $b = 2,5$ ;  $h = 0,2$ , siis  
 $n = \left[ \frac{2,5-1}{0,2} \right] + 1 = 8$  ja operaator kirjutatakse  
 se nii:

$M \sqsubset \text{ХРАНИМ } 8 \sqsubset y \sqsubset z \leftarrow$ .

Operaator "НАЧЕРТИМ" on graafiku joonestamiseks. See operatsioon kirjutatakse väljaspool tsükli, milles arvutati funktsioonide väärtused. Ühe või kahe graafiku trükkimiseks kirjutatakse operaator vastavalt nii üles:

$M \sqsubset \text{НАЧЕРТИМ } 1 \text{ ГР}, \text{ või}$

$M \sqsubset \text{НАЧЕРТИМ } 2 \text{ ГР}$ .

Ühe graafiku punktid trükitakse tärnidena. Kahe graafiku puhul trükitakse esimese graafiku punktid tärnidena  $*$ , teise graafiku punktid indeksnullina  $0$ .  $x$ -telg on graafikutel suunatud ülalt alla, s.o. vastupidi paberi liikumise suunale,  $y$ -telg - vasakult paremale. Graafiku piirid on tähistatud ristiga  $X$ , koordinaatide algus nulliga.

Operaator "РЕШИМ" on ette nähtud kahe ülesande - lineaarse algebralise võrrandisüsteemi (järk  $n \leq 22$ ) ja algebralise võrrandi (järk  $m \leq 38$ ) - lahendamiseks.

Võrrandisüsteemi lahendamiseks tuleb eelnevalt defineerida kahedimensiooniline massiiv, millele omistatakse süsteemi kordajate ja vabaliikmete väärtused. Peale seda tuleb kirjutada operaator "РЕШИМ" järgmiselt:

$M \sqsubset \text{РЕШИМ } snAx = 0$ .

Täht  $s$  tuleb tingimata kirjutada;  $n$  asemel kirjutatakse võrrandi järk;  $A$  asemel kirjutatakse täht, mida kasutati kordajate massiivi tähistamiseks; ülejäänud  $x = 0$  lisab arvuti automaatselt.

Näide  $M \sqsubset \text{РЕШИМ } S 10 cx = 0$ .

See tähendab, et tuleb lahendada lineaarse algebralise võrrandisüsteemi kordajatega  $c \ i \ j$ ;

$(i = 0, 1 \dots, 9 \quad i \ j = 0, 1 \dots, 10)$ .

Algebralise võrrandi lahendamiseks defineeritakse ühedimensiooniline massiiv ja omistatakse sellele võrrandi kordajate väärtused. Operaator kirjutatakse kujul:

$M \sqsubset \text{РЕШИМ } pAz^m = 0$ .

Täht p kirjutatakse tingimata; A asemel on kordajate massiivi nimetus; z asemel on suvaline täht, mis märgib tundmatut (v.a. i, j, n); m asemel on võrrandi järk.

Näide Võrrandi

$$b_0x^5 + b_1x^4 + b_2x^3 + b_3x^2 + b_4x + b_5 = 0$$

lahendamise operaator kirjutatakse üles nii:

$$M \text{ РЕШИМ } pbx^5 = 0.$$

Operaator "РЕШИМ" peab asuma programmi lõpus, kuna peale lineaarse võrrandisüsteemi või algebralise võrrandi lahendamist võib AP-programm olla rikitud. Pärast operaatori "РЕШИМ" täitmist peatub arvuti ise.

Operaatori "МАССИВ" abil saab andmeid operatiivmälus asuva AP-programmi kohta. Operaatoril on kaks varianti:

"МАССИВ k" ja "МАССИВ b".

Kui trükitakse "МАССИВ k", trükitab arvuti tööprogrammi pikkuse antud momendiks. Kui trükitakse "МАССИВ b" trükitab arvuti välja tööprogrammi. Käskude aadresse ei trükita, kuna tööprogramm algab alati pesast 370. Operaator "МАССИВ" viiakse sisse ilma järjekorranumbrita.

Operaator "ИСПОЛНИМ M" suunab arvuti ülesande lahendamisele alates operaatorist M. Ülesande lahendamine algab pärast reavahetust. Sealjuures arvuti kontrollib, kas tööprogramm, massiivid ja graafikute jaoks säilitatavad funktsioonide väärtused mahuvad operatiivmällu. Kui ruumi jääb väheks, trükitab arvuti "НЕ В СИЛАХ". Operaator "ИСПОЛНИМ" viiakse sisse ilma järjekorranumbrita.

### 3.5. Näited

Näide 1.

$$\text{Funktsiooni } y = \frac{364 x^2 e^{-7,5x}}{x^2 - 1,8x + 1,2} \text{ arvutamine}$$

vahemikus  $0,1 \leq x \leq 1,3$  sammuga  $\Delta x = 0,2$ .

an

"НАИРИ"

6-1-72 r 12

1 ДОПУСТИМ  $x=0,1$

2 ВЫЧИСЛИМ  $y=364x^2(\exp(-7,5x))/(x^2-1,8x+1,2)$

3 ПЕЧАТАЕМ С 5 ЗНАКАМИ  $x$   $y$

4 ВСТАВИМ  $x=x+0,2$

5 ЕСЛИ  $x$  1,31 ИДТИ К 2

6 КОНЧАЕМ

ИСПОЛНИМ 1

x=0,09999	y=1,66933
x=0,30000	y=4,60383
x=0,50000	y=3,89111
x=0,70000	y=2,17662
x=0,90000	y=0,88518
x=1,10000	y=0,26760
x=1,30000	y=0,06520

### Näide 2.

Parameetrist a sõltuva funktsiooni

$$y = \frac{4 \sqrt{1 - ax^2} \sqrt{1 - x^2}}{1 + \sqrt{a^2 + x^2}}$$

arvutamine vahemikus  $0,1 \leq x \leq 0,9$  sammuga  $\Delta x = 0,2$ . Parameetri a väärtused on 0,15; 0,35; 0,54.

Sellistes ülesannetes on mugav kasutada operaatorit "ВВЕДЕМ". Peale masina käivitamist operaatoriga "ИСПОЛНИМ 1" peatub arvuti lugemiskäsul. Trükime parameetri a väärtuse (näiteks 0,15) ja viime kelgu tagasi. Peale seda arvutab ja trükib arvuti funktsiooni väärtused vahemikus  $0,1 \leq x \leq 0,9$  ja läheb üle esimese operaatori täitmisele, kus peatub uuesti lugemiskäsul. Viime sisse uue a väärtuse ja nii edasi.

an

"НАИРИ"

5-1-72 г 3

1 ВВЕДЕМ a

2 ДОПУСТИМ  $x=0,1$

3 ВЫЧИСЛИМ  $y=4(\sqrt{1-ax^2})(\sqrt{1-x^2})/(1+\sqrt{a^2+x^2})$

4 ПЕЧАТАЕМ С 5 ЗНАКАМИ x y

5 ВСТАВИМ  $x=x+0,2$

6 ЕСЛИ  $x < 0,9$  ИДТИ К 3

7 ИНТЕРВАЛ 2

8 ИДТИ 1

ИСПОЛНИМ 1

0,15

x=0,09999	y=3,36951
x=0,30000	y=2,83801
x=0,50000	x=2,23291
x=0,70000	y=1,60242
x=0,90000	y=0,85452

0,35

x=0,09999	y=2,91272
x=0,30000	y=2,57031
x=0,50000	y=2,05490
x=0,70000	y=1,45858
x=0,90000	y=0,75082

0,54	
x=0,09999	y=2,56212
x=0,30000	y=2,30066
x=0,50000	y=1,85594
x=0,70000	y=1,30019
x=0,90000	y=0,63807

Näide 3.

Liitfunktsioonide

$$y = \frac{e^u \sqrt{5+u}}{2,5 + tg u + \sin x}, \quad z = \frac{e^v \sqrt{5+v}}{2,5 + tg v + \sin x}$$

arvutamine, kus  $u = x^2 + 2x$  ja  $v = 1 + \sqrt{x}$ . Funktsioonid leitakse vahemikus  $0,1 \leq x \leq 1,3$  sammuga  $\Delta x = 0,2$ . Näide illustreerib valemi andmist enne operaatorit. Selline moodus on otstarbekas, kui arvutatakse sama valemi järgi, kuid erinevate argumentidega.

```

an
5-1-72 r 5
y=(EXP(u)*sqrt(5+u))/(2,5+tg u+sin x)
1 ДОПУСТИМ x=0,1
2 ВЫЧИСЛИМ u=x^2+2x
3 ВЫЧИСЛИМ y
4 ДОПУСТИМ z=y
5 ВЫЧИСЛИМ u=1+sqrt(x)
6 ВЫЧИСЛИМ y
7 ПЕЧАТАЕМ С 5 ЗНАКАМИ x y z
8 ВСТАВИМ x=x+0,2
9 ЕСЛИ x < 1,3 ИДТИ К 2
10 КОНЧАЕМ
ИСПОЛНИМ 1

```

x=0,09999	y=1,45473	z=1,00104
x=0,30000	y=0,26076	z=1,31343
x=0,50000	y=-3,31168	z=1,45698
x=0,70000	y=-31,07311	z=146,55951
x=0,90000	y=24,21253	z=13,91870
x=1,10000	y=14,09888	z=23,93976
x=1,30000	y=11,94740	z=39,09579

Näide 4.

Trükkida funktsioonide

$$y = (x^2 - 1)(x - 2) \sin \pi x$$

$$z = (x^2 - 1)(x - 2) \cos \pi x$$

graafikud vahemikus  $[-1,2]$  sammuga  $0,0625$ . Arv 0 graafiku ülalosas näitab x-telje asukohta.

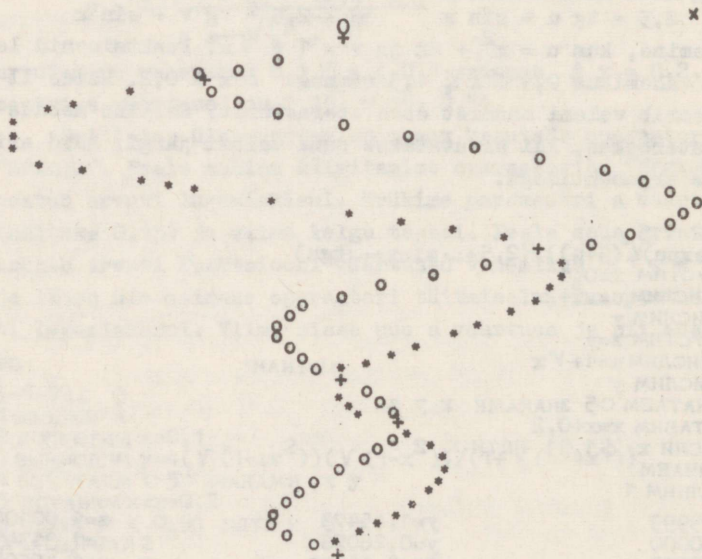
```

an
5-1-72 r 6
1 ДОПУСТИМ x=-1,2
2 ВЫЧИСЛИМ y=(x^2-1)(x-2) sin pi x
3 ВЫЧИСЛИМ z=(x^2-1)(x-2) cos pi x

```

4 ХРАНИМ 49у z  
5 ВСТАВИМ  $x=x+0,0625$   
6 ЕСЛИ  $x \leq 2$  ИДТИ К 2  
7 НАЧЕРТИМ 2 гр  
8 КОНЧАЕМ  
ИСПОЛНИМ 1

min=-1,933076068  
max= 2,014967769



### 3.6. Indeksitega muutujad

Massiivid. Massiivide kirjeldamine.

Reaalarvuliste muutujate kirjeldamiseks võib AP-s kasutada ühe või kahe indeksiga tähti. Ühe ja sama (indekseeritud) tähega kirjeldatud muutujate hulka nimetatakse massiiviks, iga sellist muutujat aga massiivi elemendiks. Vastavalt sellele, kas indekseid on üks või kaks, nimetatakse massiivi ühe- või kahemõõtmeliseks. Tähte, mida indekseeritakse, nimetatakse massiivi identifikaatoriks

Indekseeritud muutujaid sisaldava programmi alguses peab olema massiivide kirjeldus. Ühemootmeliste massiivide jaoks on see järgmine:

indekstäht = indeksi maksimaalvärtus  $\leftarrow$  tähtede loetelu  $\leftarrow$ .

Kahedimensioonilised massiivid kirjeldatakse nii:

indekstäht = esimese indeksi maksimaalvärtus  $\leftarrow$

indekstäht = teise indeksi maksimaalvärtus  $\leftarrow$  tähtede loetelu  $\leftarrow$ .

Siin indekstäht on üks langetatud tähtedest i j k ja tähtede loetelu koosneb neist muutujatest, mida me indekseerime sama maksimaalse väärtuseni. Indeksite minimaalsed väärtused on kõigil muutujatel võrdsed nulliga. Programmi alguses võib olla suvaline arv selliseid kirjeldusi. Igas kirjelduses võib kasutada samu indekstähti.

Näide: Kirjeldus

$i = 8 \leftarrow a \leftarrow b \leftarrow c \leftarrow$

$i = 2 \leftarrow j = 5 \leftarrow m \leftarrow e \leftarrow g \leftarrow$

defineerib järgmised massiivid

$a_0, a_1, \dots, a_8$

$b_0, b_1, \dots, b_8$

$c_0, c_1, \dots, c_8$

$m_{00}, m_{01}, \dots, m_{05}, \quad m_{10}, m_{11}, \dots, m_{15}, \quad m_{20}, m_{21}, \dots, m_{25}$

$e_{00}, e_{01}, \dots, e_{05}, \quad e_{10}, e_{11}, \dots, e_{15}, \quad e_{20}, e_{21}, \dots, e_{25}$

$\xi_{00}, \xi_{01}, \dots, \xi_{05}, \quad \xi_{10}, \xi_{11}, \dots, \xi_{15}, \quad \xi_{20}, \xi_{21}, \dots, \xi_{25}$

Peale massiivide kirjeldust võib programmis olla enamalt üks valem operaatori "ВЫЧИСЛИМ" jaoks ning siis tulevad operaatorid. Indekseeritud muutujaid ei tohi programmis kasutada ilma indeksiteta.

Massiivi kirjelduses antud indeksite maksimaalvärtusi, samuti indekstähti, kasutatakse masinas ainult kirjeldamise momendil mälu jaotamiseks. Programmi tekstis ei ole seepärast oluline kasutada muutujate juures samu indekstähti, nagu vastava massiivi defineerimisel.

Kui programmis on kahe indeksiga muutujad, tuleb ühes operaatoris "ДОПУСТИМ" muutujale n anda väärtus, mis on tei-

se indeksi maksimaalväärtusest ühe võrra suurem. Seda tuleb teha enne indekseeritud muutujate kasutamist, muidu programm ei tööta õieti. Vt. näida nr.2 §5.

### 3.1. Indeksiga muutujad operaatorites

AP süntaks lubab indekseeritud muutujaid kasutada operaatorites "ДОПУСТИМ", "ВЫЧИСЛИМ", "ВВЕДЕМ", "ПЕЧАТАЕМ", "СПРОСИМ". Kõigis operaatorites v.a. "СПРОСИМ", võib kasutada vaid tähelisi indekseid. Seepärast peab enne konkreetse indekseeritud muutuja kasutamist operaatoritega "ДОПУСТИМ" või "ВСТАВИМ" andma vajalikud arvulised väärtused neile täisarvulistele muutujatele (i, j, k), millega indekseeritakse.

Näide:

korrutist  $c_{57} = a_{56} \cdot b_{65}$  võib arvutada järgmiselt:

M  $\hookrightarrow$  ДОПУСТИМ i = 5  $\hookrightarrow$  j = 6  $\hookrightarrow$  k = 7  $\leftarrow$

M + 1  $\hookrightarrow$  ВЫЧИСЛИМ  $c_{ik} = a_{ij} b_{ji}$   $\leftarrow$

(eeldame, et teise indeksi maksimaalväärtus on kõigil massiividel sama).

Vaatama nüüd üksikuid operaatoreid.

Operaator "ДОПУСТИМ". Indekseeritud muutujaid võib siin kasutada nii:

indekseeritud muutuja =  $\begin{cases} \text{indekseerimata muutuja; arv;} \\ \text{(aadress)} \end{cases}$

Operaator "ВЫЧИСЛИМ". Võrdse wasak pool võib olla indekseeritud muutuja, samuti võib võrdse parem pool sisaldada indekseeritud muutujaid (kõik indeksid on tähelised!).

Operaator "ВВЕДЕМ". Sisestada võib indekseeritud muutujaid.

Näide: viia arvutisse muutujate  $a_i$  ja  $b_i$  ( $i = 0, 1, \dots, 9$ ) väärtused.

1  $\hookrightarrow$  ДОПУСТИМ i = 0  $\leftarrow$

2  $\hookrightarrow$  ВВЕДЕМ  $a_i \hookrightarrow b_i$   $\leftarrow$

3  $\hookrightarrow$  ВСТАВИМ i = i + 1  $\leftarrow$

4  $\hookrightarrow$  ЕСЛИ i  $\leq$  9  $\hookrightarrow$  ИДИ К 2  $\leftarrow$  .

Operaator "ПЕЧАТАЕМ". Operaatori sisus võivad olla indekseeritud muutujad. Arvuti trükib muutuja koos indeksi jooksva arvulise väärtusega.

Operaator "СПРОСИМ". Muutuja väärtuse teadasaamiseks trükitakse ta arvuliste indeksitega, kahe indeksi vahele tuleb jätta tühik, näiteks  $a_5 \sqcup 6$ .

### 3.8. Näited

Näide 1. Kümneõõtmeliste vektorite

$$a = (a_0, a_1, \dots, a_9),$$

$$b = (b_0, b_1, \dots, b_9)$$

skalaarkorrutise arvutamine. Esimesed neli operaatorit on algandmete sisestamiseks. Ülejäänud operaatorid arvutavad summa

$$c = \sum_{i=0}^9 a_i b_i \text{ ja trükivad tulemuse.}$$

an

"НАИРИ"

5-1-72 r 7

i=9 a b

1 ДОПУСТИМ i=0

2 ВВЕДЕМ  $a_i b_i$

3 ВСТАВИМ  $i=i+1$

4 ЕСЛИ  $i \leq 9$  ИДТИ К 2

5 ДОПУСТИМ  $i=0$   $c=0$

6 ВЫЧИСЛИМ  $c=c+a_i b_i$

7 ВСТАВИМ  $i=i+1$

8 ЕСЛИ  $i \leq 9$  ИДТИ К 6

9 ПЕЧАТАЕМ С 5 ЗНАКАМИ С

10 КОНЧАЕМ

ИСПОЛНИМ 1

1 1 2 -1 3 2 4 -2 5 3 6 -3 7 4 8 -4 9 5 10 6

c=95,00000

Näide 2. Maatriksi

$$A = \|a_{ij}\| \quad (i = 0, 1, 2, 3, 4; j = 0, 1, 2, 3, 4, 5)$$

normi arvutamine. Arvutusvalem:

$$c = \sqrt{\sum_{ij} a_{ij}^2}$$

Esimeses operaatoris omistame muutujale n väärtuse, mis on ühe võrra suurem teise indeksi maksimaalväärtusest, nagu on nõutav kahedimensioonilisi massiive sisaldavas programmis. (§4). Operaatoritega 1...7 viiakse massiiv A ridade kaupa operatiivmällu. Rea indeks on i, veeru indeks j. Operaatorites 8...14 arvutatakse normi ruut  $\sum_{ij} a_{ij}^2$ . Operaatorites

14...16 kontrollitakse, kas kõik maatriksi read on kasutatud, arvutatakse ruutude summast ruutjuur ja trükitakse tulemus.

an "НАИРИ"

5-1-72 r 10

i=4 j=5 a

```
1 ДОПУСТИМ i=0 n=6
2 ДОПУСТИМ j=0
3 ВВЕДЕМ aij
4 ВСТАВИМ j=j+1
5 ЕСЛИ i < 6 3
6 ВСТАВИМ i=i+1
7 ЕСЛИ i < 5 ИДТИ К 2
8 ДОПУСТИМ i=0 c=0
9 ДОПУСТИМ j=0
10 ВЫЧИСЛИМ c=c+aij 2
11 ВСТАВИМ j=j+1
12 ЕСЛИ j < 6 ИДТИ К 10
13 ВСТАВИМ i=i+1
14 ЕСЛИ i < 5 ИДТИ К 9
15 ВЫЧИСЛИМ c=√c
16 ПЕЧАТАЕМ С 4 ЗНАКАМИ c
17 КОНЧАЕМ
ИСПОЛНИМ 1
1 2 3 4 5 6
2 1 3 4 5 6
3 2 1 4 5 6
4 3 2 1 5 6
5 4 3 2 1 6
c=21,3307
```

Näide 3. Ruutmaatriksi jälje arvutamine.

Maatriksi jälg võrdub diagonaalelementide summaga  $\sum_{i,j} a_{ij}$ . See programm on sarnane programmiga N 2. Selgitame vaid operaatorit 9. Kuna summeeritakse ainult diagonaalelemendid, siis indeksite i ja j väärtused summas  $\sum_{i,j} a_{ij}$  on võrdsed, seepärast võib neid tähistada sama tähega, näiteks i.

an  
0-0-0 r 1008

i=4 j=4 a

```
1 ДОПУСТИМ i=0 n=5
2 ДОПУСТИМ j=0
3 ВВЕДЕМ aij
4 ВСТАВИМ j=j+1
5 ЕСЛИ j < 5 ИДТИ К 3
6 ВСТАВИМ i=i+1
7 ЕСЛИ i < 5 ИДТИ К 2
8 ДОПУСТИМ i=0 c=0
9 ВЫЧИСЛИМ c=c+aii
10 ВСТАВИМ i=i+1
```

11 ЕСЛИ  $i < 5$  ИДТИ К 9  
12 ПЕЧАТАЕМ 4 ЗНАКАМИ С  
13 КОНЧАЕМ  
ИСПОЛНИМ 1

1 2 3 4 5  
2 1 4 5 3  
3 5 1 2 4  
4 5 3 1 2  
5 4 3 2 1  
с=5,0000

### 3.9. Andmeid translaatori töö kohta

Mälujaotus. Igale reaaltüüpi indeksita muutujale vastab operatiivmälu pesa massiivist 32...55 (vt. lisa "Translaatori mälujaotus").

Konstandi  $\pi = 3,1415\dots$  hoitakse püsिमälus ja saadetakse igal AP poole pöördumisel pesa 32, mis vastab tähele  $\pi$ . Muutujatele  $i, j, k, n$  vastavad pesad 56...59. Indeks-tähtedele  $i, j, k$  vastavad pesad 60, 61, 62; neid pesi kasutavad vaid programmid, mis sisaldavad indekseeritud muutujaid.

Erinevalt indeksita muutujatest, milledest igale vastab fikseeritud operatiivmälu pesa, määratakse massiivide asukoht mälus translaatori poolt. Massiivid paiknevad pesades 981 -  $N + 980$ , kus  $N$  on indekseeritud muutujatele reserveeritud pesade üldarv.

Vaatleme kõigepealt ühemõõtmeliste massiivide kirjelduse desifreerimist masinas. Olgu  $M$  muutujate  $a, b, c$ , indeksi maksimaalvärtus. Massiivide kirjeldus on siis järgmine:

$$i = M a b c .$$

Programmi täitmisel paiknevad muutujad  $a_0, a_1, \dots, a_M$  pesades 980 -  $M \dots 980$ , muutujad  $b_0, b_1, \dots, b_M$  pesades 980 -  $2M - 1 \dots 980 - M - 1$ , muutujad  $c_0, c_1, \dots, c_M$  pesades 980 -  $3M - 2 \dots 980 - 2M - 2$ .

Üldiselt, kui  $M$  on indeksi maksimaalvärtus, siis massiivide kirjelduses  $m$ -ndal kohal olev muutuja paikneb pesades 981 -  $m(M + 1) \dots 980 - (m - 1)(M + 1)$ .

Kui maksimaalvärtusi on mitu ja kirjeldamine toimus mitmes järgus, paiknevad massiivid analoogiliselt, s.t. hiljem defineeritud massiiv asub mälus eespool.

Kahedimensionaalsete massiivide defineerimine käib nii:

$$i = M \sqcup j = N \sqcup \text{ tähtede nimestik,}$$

kus  $M$  on esimese ja  $N$  teise indeksi maksimaalväärtus. Esimese tähe jaoks tähtede nimestikust reserveeritakse pesad  $981 - (M + 1)(N + 1) \dots 980$ , teise tähe jaoks pesad  $981 - 2(M + 1)(N + 1) \dots 980 - (M + 1)(N + 1)$ ; üldiselt  $m$ -nda tähe jaoks pesad  $981 - m(M + 1)(N + 1) \dots 980 - (m - 1)(M + 1)(N + 1)$ .

Iga massiivi algusaadress kirjutatakse täisarvuna pesa, mis vastab massiivi tähistavale tähele (Iisa 5).

Selgitame öeldut näitega. Olgu antud kirjeldus

$$i = 19 \quad a \quad b \quad c .$$

Selle kirjelduse põhjal reserveerib masin muutujate  $a_i$  jaoks pesad  $961 \dots 980$ , muutujate  $b_i$  jaoks pesad  $941 \dots 960$ , muutujate  $c_i$  jaoks pesad  $921 \dots 940$ . Pesadesse 42, 34, 33, mis vastavad tähtedele  $a, b, c$  kirjutatakse täisarvud 961, 941, 921.

Olgu nüüd antud kirjeldus

$$i = 9 \quad j = 4 \quad a \quad b \quad c .$$

Iga massiiv  $a_{ij}, b_{ij}, c_{ij}$  sisaldab 50 muutujat  $a_{00} + a_{94}, b_{00} + b_{94}, c_{00} + c_{94}$ . Muutujate  $a_{ij}$  jaoks reserveeritakse pesad  $931 - 980$ , muutujate  $b_{ij}$  jaoks pesad  $881 - 930$ , ja muutujate  $c_{ij}$  jaoks pesad  $831 - 880$ . Seejuures pesadesse 42, 34, 33 kirjutatakse vastavad täisarvud 931, 881, 831.

Kui programmi täitmisel jõutakse ühe indeksiga muutujani, määratakse tema koht mälus valemiga  $A + i$ , kus  $A$  on massiivi algusaadress ja  $i$  on indeksi jooksev väärtus.

Kahedimensionaalseid massiive sisaldavas programmis tuleb alguses operaatoris "ДОПУСТИМ" omistada muutujale  $n$  väärtus  $N + 1$ , kus  $N$  on teise muutuja maksimaalväärtus. Suuruse  $a_{ij}$  asukoht mälus määratakse valemist

$$A + ni + j,$$

kus  $A$  on massiivi algusaadress,  $i$  ja  $j$  on indeksite jooksvad väärtused.

### Konstantide massiivi paigutamine

Programmis esinevad reaaltüüpi konstandid kirjutatakse operatiivmallu alates pesast 201. Programmis korduvalt esinev konstant kirjutatakse vaid ühte mälupessa. Erinevaid konstante ei tohi olla üle 125, sest konstantide massiiv

paikneb pesades 201 ... 325. Selle massiivi esimese vaba pesa aadress hoitakse täisarvuna pesas 19. Kui mällu kirjutatakse uus konstant, suurendatakse pesa 19 sisu ühe võrra. Kui translaatori poole pöörduda koodiga "na", siis pesa 19 säilib.

AP-programmi transleerimine. AP-programm transleeritakse masinasse sisseviimisel. **Tööprogramm** kirjutatakse operatiivmällu alates pesast 370. Igale AP-programmi operaatorile vastab tööprogrammi iseseisev blokk. Nimetame sellist blokki operaatori programmiks ja pesast 370 algavat käskude massiivi tööprogrammiks. Igale operaatori numbrile vastab fikseeritud pesa operatiivmällu massiivist 69 - 200. Nimetame seda algaadresside massiiviks.

AP-programmi transleerimisel on operaatori alguse tunnuseks operaatori number rea algul, lõpu tunnuseks reavaetus. Viinud sisse operaatori numbri N, kirjutab masin pesa  $68 + N$  täisarvu, mis võrdub antud operaatori programmi algusaadressiga. Edasi transleeritakse operaatori sisu, ja operaatori programm kirjutatakse tööprogrammi alates pesa  $68 + N$  salvestatud aadressist.

Operaatori programmi lõpus on üleminekukäsk, mis määrab järgmisena täidetava operaatori. Kui operaator N ei ole ülemineku- või peatamise operaator, siis antakse operaatori programmi lõpus juhtimine pesas  $68 + N + 1$  säilitatava aadressi järgi; sellega määratakse järgmisena täidetav operaator. Operaatori "ИДТИ К М" programm koosneb üleminekukäsust pesas  $68 + M$  säilitatavale aadressile. Operaatori "КОНЧАЕМ" programm lõpeb peatumiskäsuga. Selline tööprogrammi ülesehitus võimaldab sisestada operaatoreid suvalises, mitte tingimata numbrite järjekorras ning viia muudatusi transleeritud programmi, samuti alata programmi täitmist suvalisest operaatorist.

Tööprogrammi aetus operatiivmällus on määratud 64 pesa sisuga. Kui pöördume translaatori poole koodiga "an", kirjutatakse **sinna** täisarv 370. Võtnud vastu operaatori numbrimärgendi N - viib masin pesa 64 sisu pesa  $68 + N$  kui järgmise operaatori algusaadressi. Peale operaatori sisestamist ja reavahetust suurendatakse pesas 64 asuvat täisarvu ope-

raatori programmi käskude arvu võrra. Seega hoitakse pesas 64 esimese vaba pesa aadressi tööprogrammi massiivis. See aadress muutub peale operaatorei täielikku sisseviimist.

### 3.10. Vigade leidmine ja parandamine

Kui operaatorei sisendamisel tehakse süntaktiline viga, trükkib masin täрни ja annab juhtimise tagasi operaatorei märgendi vastuvõtmise blokki. Vigast operaatoreit tööprogrammi ei lisata, sest kelgu tagasiviimist pole tehtud ja pesa 64 sisu ei muutu. Edasi tuleb uuesti sisse viia märgend ja operaatorei, mille programm salvestatakse vigase operaatorei asemel.

Olgu operaatoreis märgendiga N sisuline viga, seda masin ei arvesta. Kui programmeerija märkab viga peale reavahetust, võib ta selle operaatorei sisestada uuesti sama numbriga. Uus operaatorei programm paigutatakse vabale kohale tööprogrammi massiivis, vastavalt muutub ka pesa 68 + N sisu. Eelnev operaatorei programmi salvestus ei mõju masina tööle, kuid võtab mälus ruumi.

Kui tahetakse muuta AP-programmi, tuleb tema poole pöörduda tähtedega "na". Siis säilib kõigi mälu jaotust määravate pesade, sealhulgas pesa 64 sisu. Eelnevalt transleeritud programm säilib samuti. Peale koodi "na" sisestatud operaatorei programmid kirjutatakse esialgse tööprogrammi järele. Muudetakse ka algusaadresse määravate pesade sisud.

### 3.11. AP-programmi sisseviimine ja ülesande lahendamine

Enne programmi sisseviimist on vaja:

- 1) vajutada klahvile " ПУСК 1",
- 2) trükkida "an".

Peale koodi "an" trükkib masin kuupäeva, translaatori poole pöördumise järjekorranumbri ja sõna "НАИРИ". Edasi tuleb trükkida programm. Kui sisestatakse masina jaoks mõtetut informatsiooni, s.t. eksitakse AP grammatika vastu, trükkib masin täрни - vigase informatsiooni tunnuse. Siis tuleb operaatorei uuesti sisse viia.

Kui tahetakse programmis mõnda operaatoreit muuta või lisada, tuleb vajutada klahvile " ПУСК 1" ja trükkida tähed

"na". Siis trükkida muudetava operaatori järjekorranumber ja operaator ise. Operaatoreid võib sisestada suvalises järjekorras.

Kuupäeva sisseviimiseks masinasse tuleb:

vajutada klahvile " ПУСК 1 ",

trükkida täht r ,

trükkida kuupäev, kuu number ja aasta, eraldades neid miinusmärgiga.

Näide: 24 - 5 1969 r.

Lahendamiseks trükitakse programmi lõppu operaator " ИСПОЛНИМ N". Jõudnud selle operaatorini, kontrollib masin, kas tööprogramm, indeksitega muutujad ja operaatoriga " ХРАНИМ" säilitatavad arvud mahuvad pesadesse 370 ... 980. Kui mahuvad, antakse juhtimine pesas 68 + N säilitataval aadressil. Vastasel juhul trükkib masin " НЕ В СИЛАХ " ja seiskub.

Kui operatiivmällu ei mahtunud vaid väike osa programmist, võib ülesande siiski lahendada. Selleks trükitakse programmi "mälu väljatrükk" abil algaadresside massiiv täisarvudena; vajalikud on vaid viimaste operaatorite algusaadressid. Nende järgi leitakse esimese operaator, mis ei mahtunud mällu. Nüüd leitakse mälus vaba koht, tõenäoliselt pesades 68 ... 200 või 201 ... 325. Näiteks kui operaatorite järjekorranumbrid on alla 40, siis pesad 109 ... 200 on vabad. Edasi viiakse vaba massiivi algusaadress täisarvuna pesa 64, trükitakse "na" ja sisestatakse mittemahtunud operaatorid. Ei tohi unustada, et kui viimased vajavad liiga palju pesi, rikutakse ära varemkoostatud programm. Sellise töö puhul puudub massiivi pikkuse automaatne kontroll, seda võib kontrollida vaid 64 pesa sisu järgi.

### 3.12. Näited

Summa  $\sum_{i=0}^7 a_i a_{i+2}$  arvutamine.

Esimesed neli operaatorit sisendavad masinasse muutujad  $a_0, \dots, a_9$ . Operaatorid 5 - 8 moodustavad tsükli, milles arvutatakse summa. Kuna AP-s ei või kasutada liitindekseid, näiteks indeksit  $i + 2$ , tähistame  $j = i + 2$ . Operaatoris 5 omistatakse tähtedele  $i$  ja  $j$  nende algväärtused 0 ja 2. Ope-

raatoris 6 arvutatakse korrutiste  $a_i a_j$  summa. Operaatoris 7 muudetakse muutujaid ja  $j$  nii, et  $j$  on alati kahe võrra suurem kui  $i$ . Tsükel tehakse läbi kaheksa korda, siis trükitakse tulemus ja masin peatub.

an "НАИРИ"  
5-1-72 Г 12  
i=9 a

1 ДОПУСТИМ  $i=0$   
2 ВВЕДЕМ  $a_i$   
3 ВСТАВИМ  $i=i+1$   
4 ЕСЛИ  $i < 10$  ИДТИ К 2  
5 ДОПУСТИМ  $i=0$   $j=2$   $c=0$   
6 ВЫЧИСЛИМ  $c=c+a_i a_j$   
7 ВСТАВИМ  $i=i+1$   $j=j+1$   
8 ЕСЛИ  $i < 8$  ИДТИ К 6  
9 ПЕЧАТАЕМ С 6 ЗНАКАМИ c  
10 КОНЧАЕМ  
ИСПОЛНИМ 1

1 2 3 4 5 6 7 8 9 10  
c=276,000000

### Näide 2.

Maatriksi ja vektori korrutis  $A \times B = C$ ,

$$A = \begin{pmatrix} a_{00} & a_{01} & a_{02} & a_{03} \\ a_{10} & a_{11} & a_{12} & a_{13} \\ a_{20} & a_{21} & a_{22} & a_{23} \\ a_{30} & a_{31} & a_{32} & a_{33} \end{pmatrix} \quad B = \begin{pmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \end{pmatrix} \quad C = \begin{pmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \end{pmatrix}$$

Korrutamise toimub nii:  $c_i = \sum_{j=0}^3 a_{ij} b_j$ .

Selles ülesandes kasutatakse kahe mõõtmelise massiivi A ja ühemõõtmelise massiivi B. Operaatorid 1 - 7 sisestavad massiivi A, operaatorid 8 - 11 - massiivi B. Operaatorites 12 - 19 arvutatakse järjekordne element.

an "НАИРИ"  
5-1-72 13  
i=3 j=3 a  
i=3 b

1 ДОПУСТИМ  $i=0$   $n=4$   
2 ДОПУСТИМ  $j=0$   
3 ВВЕДЕМ  $a_{ij}$   
4 ВСТАВИМ  $j=j+1$   
5 ЕСЛИ  $j < 4$  ИДТИ К 3  
6 ВСТАВИМ  $i=i+1$   
7 ЕСЛИ  $i < 2$  ИДТИ К 2  
8 ДОПУСТИМ  $j=0$

```

9 ВВЕДЕМ  $b_j$ 
10 ВСТАВИМ  $j=j+1$ 
11 ЕСЛИ  $j < 4$  ИДТИ К 9
12 ДОПУСТИМ  $i=0$ 
13 ДОПУСТИМ  $j=0$   $c=0$ 
14 ВЫЧИСЛИМ  $c=c+a_{ij}b_j$ 
15 ВСТАВИМ  $j=j+1$ 
16 ЕСЛИ  $j < 4$  ИДТИ К 14
17 ПЕЧАТАЕМ С 4 ЗНАКАМИ С
18 ВСТАВИМ  $i=i+1$ 
19 ЕСЛИ  $i < 4$  ИДТИ К 13
20 КОНЧАЕМ
    ИСПОЛНИМ 1

1 1 1 1
2 2 2 2
3 3 3 3
4 4 4 4
1 2 3 4
c=10,0000
c=20,0000
c=30,0000
c=40,0000

```

#### §4. ARVUD MASINAS. KÄSKUDE SÜSTEEM.

##### 4.1. Arvude kujutamine

Arvuti "Nairii" opereerib järgmiste arvudega:

- fikseeritud koma arvud,
- liikuva koma arvud,
- pikad arvud,
- täisarvud,
- kompleksarvud.

Fikseeritud koma arvud asuvad järkudes 1 ... 35 (pesa kahendkohtade numeratsioon kulgeb paremalt vasakule). Koma on fikseeritud 35. ja 36. järgu vahel, järk 36 on märgikoht. Arvud kirjutatakse täiendkoodis. Selliselt võib kujutada:

positiivseid arve vahemikust  $2^{-34} \leq x \leq 2 - 2^{-34}$ ,

negatiivseid arve vahemikust  $-2 \leq x \leq -2^{-24}$ ,

arvu 0.

Liikuva koma arvud on teisendatud kujule

$$x = M \cdot 2^p,$$

kus  $M$  on arvu mantiss ja  $p$  on arvu järk. Arv  $p$  on täisarv vahemikust  $-64 \leq p \leq 63$  ja tema väärtused valitakse nii, et mantiss rahuldaks võrdust

$$1/2 \leq M < 1, \text{ kui } x > 0,$$

$$-1 \leq M < -1/2, \text{ kui } x < 0.$$

Null kirjutatakse arvuna, mille mantiss ja järk võrduvad nulliga.

Liikuva koma arvude kujutamisel on arvuti pesa jaotus järgmine:

pesa järkudes 8 ... 34 asub mantiss,

pesa järkudes 35, 36 asub mantissi märk,

pesa järkudes 1 ... 6 asub järk,

pesa järgus 7 asub järgu märk.

Mantiss ja järk kirjutatakse täiendkoodis. Liikuva koma arvude täpsus on 9 kümnendkohta, nende absoluutvärtus asub vahemikus

$$10^{-19} \dots 10^{19}.$$

Pikad arvud asuvad kahes järjestikus pesas. Esimesse pessa, järkudesse 1 ... 35 kirjutatakse arvu täisosa, teise pessa, järkudesse 1 ... 35, murdosa. Arvu märk asub esimese pesa 36-ndas järgus. Teise pesa järke 35, 36 ei kasutata. Arvud on salvestatud täiendkoodis.

Täisarvud asuvad järkudes 1 ... 35. Arvu märk on 36-ndas järgus. Täisarve võib vaadata fikseeritud komaga arvudena, kusjuures koma koht on esimesest järgust paremal.

Täisarvud kujutatakse masinas täiendkoodis. Nende suurus asub vahemikus  $-2^{35} \leq x \leq 2^{35} - 1$ .

Kompleksarvud asuvad kahes järjestikus pesas. Esimeses on reaalosa liikuva koma arvuna, teises imaginaarosa samuti liikuva koma arvuna.

Kuna lõpliku arvu järkude abil võib kujutada vaid lõplikku hulka arve, siis arvude kujutamine fikseeritud koma, liikuva koma, pika või kompleksarvuna on üldiselt ligikaudne. Näiteks kümnendmurd 0,1 kujutatakse kahendsüsteemis lõpmatu perioodilise murruna. Seepärast ei või arvu 0,1 ühelgi ülalmainitud kujul masinas täpselt esitada. Seevastu säilitatakse täisarvud masinas alati täpselt - suvalise täisarvu kirjutamiseks vahemikust  $-2^{35} \leq x \leq 2^{35} - 1$  piisab 36-st kahendkohast.

#### 4.2. Käskude süsteem

Arvuti "Nairii" on kaheaadressiline. See tähendab, et käsu aadressosas võib olla kaks erinevat aadressi. Enamikust aritmeetilistest ja loogilistest tehetest võtab peale aadressosas näidatud mälupeade osa fikseeritud pesa a2. Selliste tehete tulemus kirjutatakse reeglina teisel aadressil ja pesa a2. Pesa a2 sisu on mitmetes käskudes üheks operandiks. Paljusid käske võib modifitseerida nii, et üks operand kirjutatakse käsku sisse vahetult esimese aadressi asemel.

Suvaline käsk võib olla tingimuslik. Tingimusliku käsu täitmine sõltub sellest, kas teatud pesa sisu rahuldab tema lõppu märgitud tingimust. Selle pesa aadress mahub samuti pesa lõppu.

Suvalist käsku saab formeerida. Indeksipesaks on operatiivmälu esimene pesa.

Arvuti käsud jagunevad kahte klassi: masinaoperatsioonideks ja pseudooperatsioonideks. Masinaoperatsioonid on suvalise programmi või alamprogrammi minimaalseks koostisosaks. Pseudooperatsioonid on seevastu alamprogrammid, millele omistatakse masinaoperatsioonidele sarnased koodid ja programmeeri- ja võib nendega opereerida nagu masinaoperatsioonidega. Tuleb arvestada, et pseudooperatsioonid vajavad mõningaid mälupea tööpesadena. Täielik "Nairii" käskude süsteem on toodud lisas, tabelites 2 ja 3.

#### 4.3. Käskude sise- ja väliskoodid

Eristatakse käsu sise- ja väliskoodi. Väliskood on arvude, tähtede ja teiste sümbolite kombinatsioon. Programmeeri- ja kirjutab programmi väliskoodid. Sisekood kujutab endast 36-kohalist kahendarvu. Mälu-, juhtimis- ja muudes seadmetes (välja arvatud sisendis ja väljundis) on käsud sisekoodis. Käsu sisestamisel masinasse teisendab spetsiaalne sisendprogramm (programm DII) väliskoodi sisekoodiks. Väliskood sisaldab järgmisi elemente: operatsiooni kood, parameeter  $A_1$ , modifikatsiooni tunnus, teine aadress, tingimuse tunnus, tingimuspesa aadress, formeerimise tunnus. Väliskoodis ei pruugi kõik need elemendid sees olla. Näiteks juhtimiskäskudes  $u$  ja  $e_1$  puudub teine aadress. Vaatame väliskoodi elemente

selles järjestuses, nagu nad kirjutatakse käsus.

Operatsiooni kood võib olla täht või indeksiga täht. Pseudooperatsioonide koodid on kahetähelised. Näiteks: c - liitmine,  $e_1$  - üleminek,  $yn$  - liikuva koma arvude korrutamise.

Kõikide operatsioonide koodide loetelu on toodud lisa, tabelites 2 ja 3.

Parameeter  $A_1$  kujutab endast positiivset täisarvu. Tema tähendust, sõltuvalt käsu modifikatsioonist selgitatakse allpool punktis 5.

Käsu modifikatsiooni tunnus tähistatakse väliskoodis ühega tähtedest H, k, n,  $\mathbb{I}$ , Sisekoodis asub modifikatsiooni tunnus 34. ja 35. järgus (vt. lisa, tab. 1). Tabelites 2 ja 3 on modifikatsiooni tunnus tähistatud tähega  $\theta$ . Seega tuleb tähe  $\theta$  asemele kirjutada üks tähtedest H, k, n või  $\mathbb{I}$ .

Teine address on tabelites 2 ja 3 tähistatud  $A_2$ . See on alati operatiivmälu address.

Tingimuse märk on üks märkidest =,  $\neq$ , >,  $\gg$ , <,  $\ll$ , /,  $\infty$  ja esineb vaid tingimuskäskudes.

Tingimuspesa address on tabelis 1 tähistatud tähega m. Kui seda addressi käsus pole, kontrollitakse tingimuse suhtes pesa a2 sisu.

Formeerimise tunnus on väliskoodis märk +, mis kirjutatakse vaid juhul, kui käsku on vaja formeerida. Tabelis 1 on selle tunnuse tähis F.

Väliskoodi adressid ja arvulised parameetrid kirjutatakse kümnendsüsteemis.

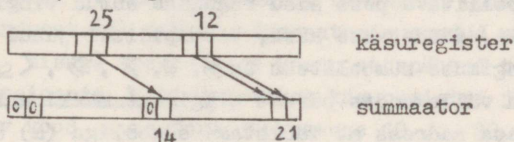
#### 4.4. Tingimusteta käskude modifikatsioonid

Tingimusteta käskudel on neli modifikatsiooni: H, k, n,  $\mathbb{I}$ .

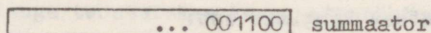
Modifikatsiooni H korral on parameetris  $A_1$  operatiivmälu või püsivmälu address. Kui  $A_1 \leq 1023$ , siis see arv on operatiivmälu address. Adresse 1024 - 2047 ei kasutata. Kui  $A_1 \geq 2048$ , on tegemist püsivmälu addressiga. Sisendkoodis kirjutatakse  $A_1$  järkudesse 12 ... 25, teine address järkudesse 1 ... 11. Addresside  $A_1$  ja  $A_2$  (või a2) sisud on käsus tehtava

tehte operandideks.

Modifikatsioon k korral kirjutatakse parameeter  $A_1$  sisendkoodis järkudesse 12 ... 25. Käsu täitmisel viiakse see käsuregistrist summaatori järkudesse 1 ... 14. Ülejäänud summaatori järkudesse kirjutatakse nullid. Teiste sõnadega, käsuregistri järkudes 12 ... 25 asuv kood nihutatakse 11 kohta paremale ja kantakse summaatorisse. Käsu täitmisel on see kood üks arvudest, millega tehte teostatakse.



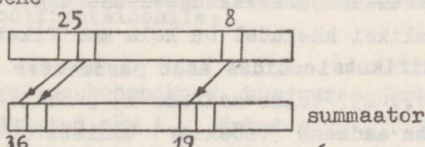
Näiteks käsk c 25 k 34 tähendab: liita arv 25, mis asub summaatoris järgmisel kohal



ja pesa 34 sisu. Tulemus kirjutada pesa 34.

Modifikatsioon n korral asub parameetri  $A_1$  käsk sisekoodi järkudes 8 ... 25. Käsu täitmisel kirjutatakse käsuregistri järkudes 8 ... 25 asuv kood summaatori järkudesse 1 ... 18, s.t. kood nihutatakse seitse kohta paremale. Ülejäänud summaatori järkudesse kirjutatakse nullid. Summaatoris oleva arvu ja teise aadressi või a2 aisuga tehakse käsus ettenähtud tehe. Teine aadress asub järkudes ühest seitsmeni. Seega peab selles modifikatsioonis teine aadress olema alla 128.

Modifikatsioon l korral asub parameeter  $A_1$  käsu sisendkoodis järkudes 8 ... 25. Käsu täitmisel kirjutatakse käsuregistri järkudes 8 ... 25 asuv kood summaatori järkudesse 19 ... 36, s.t. kood nihutatakse 11 kohta vasakule. Summaatori ülejäänud järkudesse kirjutatakse nullid. Summaatoris oleva arvu ja teise aadressi või pesa a2 sisuga tehakse käsus ettenähtud tehe



Selle modifikatsiooni korral samuti  $A_2 \leq 127$ .

#### 4.5. Tingimuslikud käsud

Tingimusliku käsu väliskoodis kirjutatakse peale teist aadressi tingimuse sümbol ja pesa aadress, mille sisu kontrollitakse. Sisekoodis asub tingimusliku käsu tunnus 36. järkus. See on üks, kui käsk on tingimuslik ja null vastupidisel juhul.

Tingimusliku käsu täitmisel selgitatakse kõigepealt, kas kontrollitava pesa sisu rahuldab antud tingimust. Kui see on nii, täidetakse käsk, vastupidisel juhul jäetakse vahele. Tingimuse sümboliteks on  $=$ ,  $\neq$ ,  $>$ ,  $\gg$ ,  $<$ ,  $\ll$ ,  $/$ ,  $\infty$ .

Olgu väliskoodis pärast tingimuse märki antud kontrollitava pesa aadress  $m$ . Tähistame sümboliga  $(m)$  arvu, mis asub pesas  $m$ . Kui tingimus on üks sümbolitest  $=$ ,  $\neq$ ,  $>$ ,  $\gg$ ,  $<$ ,  $\ll$ , võrreldakse arvu  $(m)$  nulliga, s.t. kontrollitakse ühte järgmistest tingimustest:

$(m) = 0$ ,  $(m) \neq 0$ ,  $(m) > 0$ ,  $(m) \gg 0$ ,  $(m) < 0$ ,  $(m) \ll 0$ , ja käsk täidetakse, kui see tingimus kehtib. Sümbolile 1 vastav tingimus kehtib, kui klahv "BAPIAHT" on alla vajutatud. Sümboli  $\infty$  puhul täidetakse käsk, kui pesas  $m$  asuva kahendkoodi 35. ja 36. järk langevad kokku.

Näiteks käsk  $c\ 25\ H\ 34 > 3$ . Kui pesas 3 asub positiivne arv, liidetakse pesade 25 ja 34 sisu ning tulemus salvestatakse pesa 34. Kui pesas 3 asub negatiivne arv või null, jätab masin selle käsu vahele.

Kui käsus ei märgita teist aadressi  $A_2$ , (näiteks juhtimiskäskudes), siis on tingimuse sümboli koht kohe peale modifikatsiooni tunnust. Tingimuslikus käsus ei pruugi olla kontrollitava pesa aadressi. Siis kontrollitakse tingimuse suhtes pesa  $a_2$  sisu.

Näiteks käsk  $c\ 25\ H\ 34 \gg$  täidetakse, kui pesas  $a_2$  on positiivne arv või null.

#### 4.6. Tingimuslike käskude modifikatsioonid

Tingimuslikel käskudel on kolm modifikatsiooni:  $H$ ,  $n$ ,  $d$ . Kõikides modifikatsioonides asub parameeter  $A_1$  sisekoodi järkudes 12 ... 25, teine aadress  $A_2$  järkudes 1 ... 4. Kontrollitava pesa aadress (tabelis 1 tähistatud tähega  $m$ ) asub järkudes 8 ... 11. Järkudesse 5 ... 7 kirjutatakse tingimuse

kood. Seega tingimuslikes käskudes  $A_2 \leq 15$  ja  $m \leq 15$ .

Modifikatsioon H on  $A_1$  operatiivmälu või püsिमälu pesa, kus hoitakse ühte operandi. Endiselt, kui  $A_1 \leq 1023$ , siis on ta püsिमälu aadress, kui  $A_1 \geq 2048$ , on ta püsिमälu aadress. Aadresse 1024 ... 2047 ei kasutata.

Modifikatsioon n kirjutatakse käsuregistri järkudes 12 ... 25 asuv kood summaatori järkudesse 1 ... 14, s.t., nihutades seda koodi 11 kohta paremale. Ülejäänud summaatori järkudesse kirjutatakse nullid. Summaatoris saadud arvuga ja pesa  $A_2$  või  $a_2$  sisuga teostatakse käsus ettenähtud tehe.

Modifikatsioon J kirjutatakse käsuregistri järkudes 12 ... 25 asuv kood summaatori järkudesse 19 ... 32, s.t., nihutades seda koodi 7 järku vasakule. Ülejäänud summaatori järkudesse kirjutatakse nullid. Summaatoris saadud arvuga ja pesa  $A_2$  või  $a_2$  sisuga teostatakse käsus ettenähtud tehe.

#### 4.7. Käskude formeerimine

Formeerimise tunnuseks väliskoodis on märk + käsu lõpus, sisekoodis "üks" 26-ndas järgus. Formeeritava käsu täitmisel summeerib masin käsu kahendkoodi operatiivmälu esimeses pesas asuva koodiga. Saadud kood antakse käsuregistrisse ja kujutab endast käsku, mida masin täitma hakkab. Mäluseadmes säilib formeeritav käsk esialgsel kujul.

Formeeritavate käskude kasutamist illustreerib antud peatüki näide 3. §6.

#### 4.8. Tinglikud tähistused

Käskude süsteemi kirjeldamisel kasutame järgmisi tähistusi:

- (A) - kood, mis asub pesas aadressiga A (pesa A sisu);
- $\theta$  - modifikatsiooni üldine märk. Reaalsetes käskudes on  $\theta$  asemel üks tähtedest H, k, n, J;
- $[A_1]_{\theta}$  - üks koodidest  $[A_1]_H$ ,  $[A_1]_k$ ,  $[A_1]_n$ ,  $[A_1]_J$  vastavalt käsu modifikatsioonile;
- $[A_1]_H = (A_1)$ ;
- $[A_1]_k$  - 36-järguline kahendkood, kusjuures järkudes 1 ... 14 on kirjutatud arv  $A_1$ , järkudes 15 ... 36 nullid;
- $[A_1]_n$  - 36-järguline kahendkood. Tingimusteta käsus asub

järkudes 1 ... 18 arv  $A_1$ , järkudes 19 ... 36 nullid. Tingimuslikus käsus on järkudes 1 ... 14 arv  $A_1$ , järkudes 15...36 nullid;

$[A_1]_{\pi}$  - 36-järguline kahendkood. Tingimusteta käsus on järkudes 1 ... 18 nullid, järkudes 19 ... 35 arv  $A_1$ .

Tingimuslikus käsus on järkudes 1 ... 18 nullid, järkudes 19 ... 32 arv  $A_1$ ;

$(A)^k$  - kompleksarv, mille reaali- ja imaginaariosa asuvad liikuva koma arvudena vastavalt pesades  $k$  ja  $k + 1$ ;

$[A_1]_{\theta}^{14}$  - koodi  $[A_1]_{\theta}$  järgud 1 ... 14;

$(A)^{\theta}$  - pikk arv pesades  $A$  ja  $A + 1$ ;

$\vec{A}$  - koodi  $(A)$  nihe paremale;

$\overleftarrow{A}$  - koodi  $(A)$  nihe vasakule;

$\Rightarrow A$  - operatsiooni tulemuse saatmine pesa aadressiga  $A$ ;

$\Rightarrow A^{\theta}$  - pika arvu saatmine pesadesse  $A$  ja  $A + 1$ .

#### 4.9. Masinaoperatsioonid

Masinaoperatsiooni kood tähistatakse väliskoodis ühe (indeksiga) tähega. Sisekoodis on masinaoperatsiooni tunnuseks null 33. järgus.

Eelnevas paragrahvis kirjeldatud modifikatsioonid käivad käsu aadressosa kohta. Neid kasutatakse nii masina kui ka pseudooperatsioonides. Suurema osa masinaoperatsioonide jaoks on olemas veel tehte koodi modifikatsioonid. Väliskoodis tähistatakse modifitseeritud operatsiooni koodi indekseeritud tähega. Näiteks masinaoperatsioonil "liitmine" on neli operatsiooni koodi modifikatsiooni  $c, c_1, c_2, c_3$ . Käsul "liitmine mooduli 2 järgi" on kaks modifikatsiooni -  $M, M_1$  jne.

Kui aritmeetiliste, loogiliste ja mõningate teiste käskude kood on indeksita täht, siis vastavad operatsioonid täidetakse kui kaheaadressilised. Samad operatsioonid, mille koodiks on täht üks, täidetakse kolmeaadressilistena.

Tehte operandideks on koodid  $(a_2)$  ja  $[A_1]_{\theta}$ , tulemus kirjutatakse pesa  $a_2$  või pesadesse  $a_2$  ja  $A_2$ . Näiteks, käsu  $c A_1 H A_2$  täitmisel liidetakse koodid  $(A_1)$  ja  $(A_2)$  ja tulemus kirjutatakse pesadesse  $a_2$  ja  $A_2$ . Käsu  $c_1 A_1 H A_2$  täitmisel liidetakse koodid  $(a_2)$  ja  $(A_1)$  ning tulemus salvestatakse pesadesse  $A_2$  ja  $a_2$ . See reegel ei kehti aga kõikide masina-

operatsioonide kohta. Näiteks operatsioonikoodid  $u, u_1, u_2, u_3$  või  $e, e_1$  tähistavad täiesti erinevaid käske. Seepärast on masinaoperatsioonide tabelis (lisa tab. 2) antud kõik operatsiooni koodide lubatud modifikatsioonid.

Tehte koodide modifikatsioonid on olemas ainult masinaoperatsioonide jaoks.

#### 4.10. Aritmeetilised operatsioonid

Aritmeetiliste tehete sooritamiseks on järgmised 16 käsku: liitmine -  $c, c_1, c_2, c_3$ ; lahutamine -  $b, b_1, b_2, b_3$ ; korrutamine -  $y, y_1, y_2, y_3$  ja jagamine -  $g, g_1, g_2, g_3$ . Operatsioonid teostatakse fikseeritud koma arvudega. Käskudes  $c, b, y, g$  tehakse tehe arvudega  $[A_1]_0$  ja  $(A_2)$  ning tulemus kirjutatakse pesadesse  $A_2$  ja  $a_2$ . Käskudes  $c_1, b_1, y_1, g_1$  tehakse tehe arvude  $[A_1]_0$  ja  $(a_2)$  vahel, tulemus kirjutatakse pesadesse  $A_2$  ja  $a_2$ . Näiteks käsu  $c25H24$  täitmisel liidetakse arvud ning tulemus salvestatakse pesa 24 ning  $a_2$ . Käsu  $b12524$  täitmisel lahutatakse pesa  $a_2$  sisust pesa 25 sisu ja kirjutatakse tulemus pesadesse  $a_2$  ja 24.

Käskudes  $c, c_1, b, b_1, y, y_1, g, g_1$  ei analüüsita tehte tulemust ületäitumise suhtes. Kui tulemus ei rahulda tingimust  $-2 \leq x < 2$ , jätkab masin arvutamist, andes vale tulemuse.

Operatsioonides  $c_2, c_3, b_2, b_3, y_2, y_3, g_2, g_3$  analüüsitakse tulemust ületäitumise suhtes. Kui tulemuse absoluutväärtus on ühest suurem või ühega võrdne, kirjutatakse järgmise käsu aadress pesa 1022 ja juhtimine antakse pesa 1023. Ülejäänus on käsud  $c_2, b_2, y_2, g_2$  analoogilised käskudega  $c, b, y, g$  ja käsud  $c_3, b_3, y_3, g_3$  käskudega  $c_1, b_1, y_1, g_1$ .

#### 4.11. Nihked

Koodide nihkeid tehakse käskudega  $a, a_1, 0, 0, 1, 1, 1, 1$ . Nendes käskudes ei kasutata modifikatsiooni  $n$ .

Aritmeetiline nihe paremale -  $a, a_1$ . Operatsioonis  $a$  nihutatakse pesas  $A_2$  olevat koodi, operatsioonis  $a_1$  - pesas  $a_2$  olevat koodi. Tulemus kirjutatakse pesadesse  $A_2$  ja  $a_2$ . Järkude arv, mille võrra nihutatakse, asub koodi  $[A_1]_0$  järkudes 1 ... 7. Kui nihutatakse positiivset arvu, kirjutatakse vabanenud järkudesse nullid, kui negatiivset - ühed.

Asugu näiteks pesades 25 ja 26 koodid:

25 0101100...

26 1100100...

Peale operatsioone a 4n 25 ja a 4n 26 asuvad neis pesades koodid:

25 000001011000...

26 11111100100...

Loogiline nihe vasakule -  $\delta, \delta_1$ .

Operatsioon  $\delta$  nihutatakse pesas  $A_2$  olevat arvu, operatsioon  $\delta_1$  - pesas  $a_2$  olevat arvu. Nihutatavate järkude arv on kirjutatud koodi  $[A_1]_{\Theta}$  järkudesse 1 ... 7. Vabanenud järkudesse kirjutatakse nullid. Tulemus kirjutatakse pesadesse  $A_2$  ja  $a_2$ .

Pika arvu nihe - paremale  $r$  ja vasakule  $r_1$ .

Nihutatakse pesades  $A_2$  ja  $A_2 + 1$  asuvat pikka arvu. Nihe  $r$  arv asub koodi  $[A_1]_{\Theta}$  järkudes 1 ... 7. Tehte  $r$  korral kirjutatakse need koodi  $(A_2)$  järgud, mis nihutatakse esimesest kohast paremale, pesa  $A_2$  alates 34. järgust. Järke 35 ja 36 ei kasutata; nooremad järgud, mis nihutatakse pesast  $A_2 + 1$  välja, lähevad kaotsi. Tehte  $r_1$  korral kirjutatakse need koodi  $(A_2 + 1)$  järgud, mis nihutatakse vasakule üle 34. järgu, pesa  $A_2$  noorematesse järkudesse. Vanemad järgud, mis nihutatakse üle pesa  $A_2$  vasakpoolse otsa, lähevad kaotsi. Tulemus kirjutatakse pesadesse  $A_2$  ja  $A_2 + 1$ .

#### 4.12. Loogilised operatsioonid

Loogiline liitmine -  $l, l_1$ , loogiline korrutamine -  $\pi, \pi_1$ , liitmine mooduliga 2 -  $M, M_1$ . Operatsiooni  $l, \pi, M$  tehakse koodidega  $[A_1]_{\Theta}$  ja  $(A_2)$ , tulemused kirjutatakse pesa  $a_2$ . Operatsioonid  $l_1, \pi_1, M_1$  tehakse koodidega  $[A_1]_{\Theta}$  ja  $a_2$ , tulemus kirjutatakse pesadesse  $A_2$  ja  $a_2$ .

Loogiline liitmine  $\vee$  toimub järgmiste reeglite järgi:

$$0 \vee 0 = 0, \quad 0 \vee 1 = 1, \quad 1 \vee 0 = 1, \quad 1 \vee 1 = 1.$$

Seda operatsiooni kasutatakse koodide ühendamiseks.

Loogiline korrutamistoimub järgmiste reeglite järgi:

$$0 \wedge 0 = 0, \quad 0 \wedge 1 = 0, \quad 1 \wedge 0 = 0, \quad 1 \wedge 1 = 1.$$

Seda operatsiooni kasutatakse koodiosa eraldamiseks.

Liitmine mooduliga 2 toimub järgmiste reeglite järgi:

$$0 \oplus 0 = 0, \quad 0 \oplus 1 = 1, \quad 1 \oplus 0 = 1, \quad 1 \oplus 1 = 0.$$

Seda operatsiooni kasutatakse koodide võrdlemiseks.

#### 4.13. Operatsioonid koodidega

Koodi ülekanne -  $n, n_1$ . Väliskoodis  $nA_1 \ominus A_2$  või  $n_1 \ominus A_2$ . Käsu  $nA_1 H A_2$  täitmisel kirjutatakse pesas  $A_1$  asuv kood pesadesse  $A_2$  ja  $a_2$ . Pesas  $A_1$  olev kood säilib. Käske  $nA_1 nA_2$ ,  $nA_1 kA_2$ ,  $nA_1 H A_2$  kasutatakse koodide formeerimiseks pesas  $A_2$ . Näiteks, käsk  $n2048kA_2$  saadab pesa  $A_2$  kaheteistkümnendasse järku ühe, ülejäänud järkudes on nullid. Käsk  $n1023kA_2$  saadab ühed  $1 \div 10$  järku pesa  $A_2$  ja nullid ülejäänud järkudesse. Käsud  $nOnA_2$  ja  $nOkA_2$  puhastavad pesad  $A_2$  ja  $a_2$ . Käsk  $nA_1 \ominus$  viib pesa  $a_2$ . Näiteks käsk  $nOn$  puhastab pesa  $a_2$ .

Käsuga  $n_1 \ominus A_2$  viiakse kood pesast  $a_2$  üle pesa  $A_2$ .  
Normaliseerimine  $H, H_1$ . Normaliseerituiks loetakse arvud  $00, 1... \text{ või } 11, 0... \text{ . Null loetakse normaliseeritud arvuku. Arvud kujul } 01, 0... ; 01, 1... ; 10, 0... ; 10, 1... ; 11, 1... ; 00, 0... \text{ on normaliseerimata, kusjuures esimeses neljas näites on normaliseerimine rikutud vasakule, viimases kahes paremale.}$

#### 4.14. Juhtimisoperatsioonid

Käsud  $u, u_1, u_2, u_3, l, l_1, k, k_1, x$  annavad programmeerijatele võimaluse muuta käskude täitmise järjekorda ja peatada masinat.

Üleminek tagasipöördeta -  $u$ . Käsu väliskoodis teist aadressi ei kirjutata. Käsu täitmisel kantakse koodi  $[A_1] \ominus$  järkud  $1 \dots 14$  üle käsuloendajasse. Seepärast täidetakse peale käsku  $uA_1 \ominus$  käsk pesast aadressiga  $[A_1] \ominus^{14}$ . Pesa  $a_2$  sisu säilib.

Näide: Käsk  $u100n$  annab juhtimise aadressil 100. Käsk  $u100H$  annab juhtimise aadressil, mis asub sajanda pesa neljateistkümnemes nooremas järgus. Käsk  $u150n <$  on tingimuslik: kui  $(a_2) < 0$ , minna üle käsule 150. pesas, vastasel juhul täita järgmine käsk.

Üleminek tagasipöördumisega -  $u_1$ . Väliskood  $u_1 A_1 \ominus A_2$ . Seda kasutatakse pöördumisel alamprogrammi poole. Käsu täitmisel antakse juhtimine aadressil  $[A_1] \ominus^{14}$ . Pessa  $A_2$  kirjutatakse käsk  $uNn$ , kus  $N$  on üleminekukäsu järgmise käsu aadress. Alamprogrammi lõpus peab olema käsk  $uA_2n$ . Käsu  $u_1$  abil võib mitu korda pöörduda sama alamprogrammi poole, kusjuures

alati on kindlustatud tagasipöördumine üleminekukasust järgmisele käsule. Pesa  $a_2$  sisu säilib. Vt. punkt 19 näide 2.

Üleminek võtme järgi -  $u_3$ . Väliskood  $u_3 A_1 \theta$ . Käsk täidetakse, kui juhtimisplüüdil on sees klahv "КЛЮЧ". Ülejäänus on käsk  $u_3$  analoogiline käsule  $u$ .

Käsulooendaja sisu salvestamine -  $e$ . Väliskood  $e A_1 \theta A_2$ . Käsu  $e$  täitmisel summeeritakse käsulooendaja sisu koodiga  $[A_1] \theta$ . Tulemus salvestatakse pesadesse  $A_2$  ja  $a_2$ . Arvestame et käsulooendajas olev arv on ühe võrra suurem käsk  $e$  aadressist, sest peale järjekordse arvu kandmist mälust käsuregistrisse suurendatakse käsulooendaja sisu ühe võrra.

Suhteline tingimusteta üleminek -  $e_1$ . Väliskood  $e_1 A_1 \theta$ . Selle käsu täitmisel summeeritakse käsulooendaja sisu koodiga  $[A_1] \theta^{14}$  ja tulemus saadakse käsulooendajasse. Seega käsk ütleb: "minna niimitu käsku edasi", või "pöörduda niipalju käske tagasi". Kõige sagedamini kasutatakse siin modifikatsioon  $n$ . Käsu eripära on selles, et temas ei näidata konkreetseid aadresse, vaid antakse suhteline üleminek mingi arvu aadresside võrra "edasi" või "tagasi". Seepärast käsu  $e$  kirjutus ei sõltu programmi asukohast mälus.

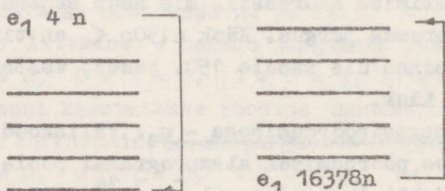
Oletame, et käsk  $e_1 A_1 n$  asub pesas aadressiga  $p$  ja tuleb teha üleminek käsule aadressiga  $t$ . Kui minnakse "edasi", s.t.  $t > p$ , võib  $A_1$  arvutada valemi järgi

$$A_1 = t - p - 1.$$

Kui pöördutakse "tagasi", s.t.  $t < p$ , siis

$$A_1 = 16383 - (p - t)$$

Näiteks:



Siin tähendab horisontaalkriips mingit käsku, üleminekud on näidatud noolega.

Peatus -  $k, k_1$ . Väliskoodid  $k A_1 \theta$ ;  $k_1 \theta \theta$ . Käsul  $k$  peatub

masin ja summaatorisse antakse kood  $[A_1]_{\Theta}$ . Summaatoris olevat koodi võib lugeda juhtimispuldi alumiselt lampiderealalt. Käsul  $k_1$  masin peatub ja summaatorisse antakse pesas  $a_2$  olev kood.

Tühikäsk - x. Väliskood võib koosneda ainult tähest x.

#### 4.15. Välisseadmete pools pöördumise käsud

Sümboli trükk - 0,  $O_1$ . Masinas "Nairii" vastab igale tele-  
letaibi sümbolile kahendkood, mida nimetatakse sümboli sise-  
koodiks. Sisekoodid asuvad püsimalus alates aadressist 2176. Käsu  $0A_1H$  täitmisel trükib masin sümboli, mille sisekood asub pesas aadressiga  $A_1$ . Sümbolid ja vastavad sisekoodide aadressid on toodud lisas tab. 4. Kõige mugavam käsu jaoks on modifikatsioon H. Teiste modifikatsioonide puhul peab arv  $[A_1]_{\Theta}$  olema trükitava sümboli kood. Käskude 0 ja  $O_1$  täitmisel käsu  $a_2$  sisu säilib.

Iugemine - r. Käsk on informatsiooni sisendamiseks tele-  
taibilt või transmitterilt operatiivmällu. Käsu täitmisel kirjutatakse sümboli sisekood pesa  $A_2$ . Pesa  $a_2$  sisu säilib. Märkime, et püsimalus on kõikide masinas tarvitata-  
vate arvutüüpide jaoks olemas sisendamis- ja väljestamisprgrammid. Standardprogrammides on algandmete sisendamine samuti olemas. Iugemiskäsku on vaja siis, kui programmeerija soovib sisenda-  
da algandmeid sõltumatult olemasolevatest programmides.

#### 4.16. Pseudooperatsioonid

Pseudooperatsiooni tunnus sisekoodis on üks 33. järgus. Üleminekul pseudooperatsioonile täidetakse järgmised etteval-  
mistavad käsud:

1. Operatiivmälu kolmanda pesa järkudesse 8 ... 36 kirjutatakse koodi  $[A_1]_{\Theta}$  järgud 8 ... 36; kolmanda pesa ülejäänud järkudesse 1 ... 7 kirjutatakse nullid.
2. Operatiivmälu neljanda pesa järkudesse 1 ... 7 kirjutatakse koodi  $[A_1]_{\Theta}$  järgud 1 ... 7. Kõheksandasse järku kirjutatakse sama mis seitsmendasse; järkudesse 9 ... 36 nullid.
3. Operatiivmälu viienda pesa järkudesse 8 ... 36 kirjutatakse koodi  $(A_2)$  järgud 8 ... 36; järkudesse 1 ... 7 kirjutata-

takse nullid.

4. Operatiivmälu kuuenda pesa järkudesse 1 ... 7 kirjutatakse koodi  $[A_1]_6$  järgud 1 ... 7. Kaheksandasse järku kirjutatakse sama, mis seitsmendasse; järkudesse 9 ... 36 salvestatakse nullid.
5. Operatiivmälu seitsmenda pesa järkudesse 1 ... 14 kirjutatakse käsuloendaja sisu, s.t. järgmise käsu aadress. Järkudesse 19 ... 28 kirjutatakse arv  $A_2$ . Ülejäänud seitsmenda pesa järkudesse kirjutatakse nullid.
6. Operatsiooni koodile liidetakse 4352 ja tulemus saadetakse käsuloendajasse. Müüd algab antud pseudooperatsiooni realiseeriva alamprogrammi täitmine. Iga pseudooperatsiooni programm lõpeb käsuga  $u_27$  H. Selle käsu täitmine sõltub klahvi "ПОЛУАВТОМАТИЧЕСКИЙ" asetusest. Kui ta on üleval, on käsk  $u_27$  H ekvivalentne käsuga  $u_7$  H. Vajutatud klahvi korral on käsk  $u_27$  H ekvivalentne käsuga  $k_7$  H, s.t. masin peatub ja puldile ilmub seitsmenda pesa sisu.

Paljudes pseudooperatsioonides uuritakse, kas lähteandmed on lubatavad, kontrollitakse ületäitumist ja signaliseeritakse ületäitumise puhul. Näiteks kui operatsiooni puhul on juurealuse avaldise märk negatiivne, trüüb masin  $x < 0$  ja peatub.

Kui masin seiskub pseudooperatsiooni täitmisel, võib juhtimispuldilt summaatori neljateistkümmes nooremas järgus lugeda järgneva käsu aadressi.

Lisas on antud pseudooperatsioonide tabel (tabel 3). See on analoogiline masinaoperatsioonide tabeliga, lisatud on ainult lahter "tööpesad", kus on kirjas käsu täitmisel tarvitsminevad operatiivmälu pesad. Samuti on iga pseudokäsu jaoks kirjas võimalikud avaripeatused ja vastav signaalsatsioon.

#### 4.17. Mõningate pseudooperatsioonide selgitus

Liikuva koma arvude denormaliseerimine -  $gH$  on määratud liikuva koma arvu teisendamiseks pikaks arvuks. Selles käsus vaadatakse koodi  $[A_1]_3$  kui liikuva koma kujul antud arvu. Eraldatakse arvu täis- ja murdosa ja kirjutatakse täisosa pessa  $A_2$ , murdosa pessa  $A_2 + 1$ . Kui lähte arvu järk

on suurem kui 34, trükitakse  $\infty$  ja masin peatub.

Täisarvude jagamine - gm. Jagatakse jäägiga ( $A_2$ ):  $[A_1]_{\Theta}$  tulemuse täisosas kirjutatakse pessa  $A_2$ , jääk kolmandasse pessa. Kui  $[A_1]_{\Theta} = 0$ , trükitakse  $\infty$  ja masin peatub.

Märkus. Täisarve saab liita ja lahutada masinaoperatsioonidega "liitmine" ja "lahutamine".

Pseudooperatsioonides pikkade ja kompleksarvudega (koodid cg, bg, yg, gg, om, hg, ng, ck, bk, yk, gk) on modifikatsioonidel eelnevast erinev tähendus. Nimelt modifikatsioonides k ja n on  $A_1$  pika või kompleksarvu esimese pesa aadress; modifikatsioonis H on  $A_1$  pesa aadress, kuhu on kirjutatud esimese pesa aadress. Modifikatsiooni  $\Gamma$  ei kasutata. Näiteks käsk cg 50k 60 tähendab: liita pikad arvud pesadest 50, 51 ja 60, 61 ning paigutada tulemus pesadesse 60, 61. Tabelis 3 on tulbas "täidetav operatsioon" modifikatsioonidele k ja n vastav operatsioon.

Pika arvu jaatmine - on. Pikk arv pesadest  $A_1$  ja  $A_1 + 1$  kirjutatakse (modifikatsioonides k ja n) pesadesse  $A_2$  ja  $A_2 + 1$ .

Pika arvu normaliseerimine - hg. Võimaldab üle minna pika arvu kujult liikuva koma kujule. Pikk arv pesadest  $A_1$  ja  $A_1 + 1$  teisendatakse kujule  $M \cdot 2^p$  ja kirjutatakse pesa  $A_2$ .

#### 4.18. Trükkimise pseudokäsud

Liikuva koma arvude trükk - nn. Käsu väliskood on nn  $A_1 H q$ . Selle käsuga trükitakse pesa  $A_1$  sisu kümnendarvuna. Sõltuvalt arvu q kujust on võimalikud kaks trükkimisvarianti.

1)  $0 < q \leq 31$

arv trükitakse q kohaga peale koma.

2)  $k < q \leq k + 9$ , kus k on üks arvudest 128, 256, 512. Arv trükitakse q - k kohaga peale koma ja jäetakse niipalju tühikuid, et ta koos tühikutega võtaks enda alla 20 märgikohta. Käsku kasutatakse sellisel kujul, kui tulemused tuleb trükkida mitmes tulbas.

Täisarvude trükk - nm. Käsu väliskood on nm  $A_1 \Theta$ . Kood  $[A_1]_{\Theta}$  viiakse kümnendsüsteemi ja trükitakse täisarvuna.

Pikkade arvude trükk. Käsu väliskood on  $ngA_1kq$  või  $ngA_1nq$  (lähemalt modifikatsioonidest vt. pikkade arvude operatsioonide kirjelduses). Selle käsu kohaselt trükitakse pesade  $A_1$  ja  $A_1 + 1$  sisud kümnendarvudena  $q$  kohta peale koma.

Murdarvude trükk - nr. Väliskood nr  $A_1 \theta q$ . Käsu täitmisel teisendatakse kood  $[A_1] \theta$  kui fikseeritud koma arv kümnendsüsteemi ja trükitakse  $q$  kohta peale koma.

Käsu trükk - nk. Käsu täitmisel vaadatakse koodi  $[A_1] \theta$  mingi käsu sisekoodina, teisendatakse väliskoodi ja väliskood trükitakse. Kui koodi  $[A_1] \theta$  järkudes 27 ... 32 ei ole ühegi käsu sisekoodi, siis teisendatakse kood  $[A_1] \theta$  kaheksandküsteemi ja trükitakse.

Sisu trükk - nc. Selles käsus trükitakse kood  $[A_1] \theta$  kaheksandküsteemis, s.t. nullide ja ühete kombinatsioonina.

Arvuliste indeksite trükk - nu. Selle käsu järgi teisendatakse kood  $[A_1] \theta$  kui täisarv kümnendsüsteemi, trükitakse arvulise indeksina ja jäetakse tühik. Peab kehtima tingimus

$$0 \leq [A_1] \theta \leq 1000.$$

Kui  $[A_1] \theta > 1000$  või  $[A_1] \theta < 0$ , trükitab masin \*nu ja peatub.

#### 4.19. Näited

Näide 1. Vaatame funktsiooni

$$y = 364x^2e^{-7,5x}$$

väärtuste arvutamise programmi. Argument muutub vahemikus  $0,1 \leq x \leq 1,2$  sammuga  $h = 0,1$ . Programm asub pesades 110... 129. Alganud asuvad pesades 100 ... 104 järgmiselt:

100	$x_0 = 0,1$
101	$h = 0,1$
102	$M = 1,2$
103	$= 364,0$
104	$= 7,5$
106	$x_1$ , arvu $x$ jooksev väärtus.

Programm töötab järgmiselt. Pesas 100 asub käsk  $n100m106$  toimetab argumenti algväärtuse  $x_0$  pessa, kus hoitakse argumenti jooksvat väärtust. Käskudes 111 ... 113 arvutatakse  $364 x_1^2$  ja saadetakse 107. pessa. Käskudes 114 ... 116 arvuta-

takse  $e^{-7,5x}$  i. Käsk 117 arvutab  $y_1$  väärtuse. Käsud 118...124 trükiivad  $x_1$  ja  $y_1$  väärtused. Käsk 125 suurendab  $x$  jooksvat väärtust sammu  $h$  võrra. Käskudes 126 ja 127 arvutatakse  $M - x_1$  ja tulemus saadetakse pesadesse 107 ja  $a_2$ . Käsk  $u111n$  on tinglik. Üleminek 111. aadressile toimub siis, kui vahe  $M - x_1$  on positiivne. Kui  $M - x_1 \leq 0$ , juhtimist ei muudeta ja täidetakse käsk kon - peatus.

110k20

110	n100 H 106
111	n106 H 107
112	yn107 H 107
113	yn103 H 107
114	n104 H 108
115	yn106 H 108
116	ex108 H 108
117	yn107 H 108
118	o2220 H
119	o2250 H
120	mn106 H 515
121	o2214 H
122	o2250 H
123	mn108 H 5
124	o2274 H
125	cn101 H 106
126	n106 H 107
127	pn102 H 107
128	u111n
129	kOn

100n5

100	0,099999999962
101	0,099999999962
102	1,209999999344
103	364,000000000000
104	-7,500000000000

110u

x=0,099	y=1,71941
x=0,199	y=3,24877
x=0,300	y=3,45287
x=0,400	y=2,89959
x=0,500	y=2,14011
x=0,600	y=1,45572
x=0,700	y=0,93594
x=0,800	y=0,57745
x=0,900	y=0,34522
x=1,000	y=0,20132
x=1,099	y=0,11506
x=1,199	y=0,06468

Näide 2. Arvutada funktsiooni

$$y = 1 + x + chx + ch(1 + 3x).$$

Paigutame algandmed mällu järgmiselt:

100	1,0
101	3,0
102	0,5
103	x

Programmi y arvutamiseks asub pesades 110 ... 126, ch arvutamise alamprogramm pesades 130 ... 136. Käsud 110, 111 arvutavad  $1 + x$ . Käsk 112 n103 H 105 saadab argumendi chx arvutamise alamprogrammi tööpesa. Käsk u<sub>1</sub>130k107 annab juhtimise chx arvutamise alamprogrammi. Seejuures kirjutatakse 107. pesa automaatselt käsk u114n. Funktsiooni chx väärtus asub 106. pesas. Käskudes 115 ... 118 arvutatakse  $1 + 3x$ , saadetakse see 105. pesa ja pöördutakse tagasi hüperboolse koosinuse arvutamise alamprogrammi. Seekord on 107. pesas käsk u119n. Ülejäänud käskudes arvutatakse y, trükitakse x ja y väärtused ja peatatakse masin.

110k17

110	n103 H 104
111	cn100 H 104
112	n103 H 105
113	u <sub>1</sub> 130k107
114	cn106 H 104
115	n103 H 105
116	yn101 H 105
117	cn100 H 105
118	u <sub>1</sub> 130k107
119	cn106 H 104
120	o2220 H
121	o2250 H
122	nn103 H 514
123	o2214 H
124	o2250 H
125	nn104 H 5
126	k0 H

130k7

130	n0k106
131	oβ105 H 106
132	ex105 H 105
133	ex106 H 106
134	cn105 H 106
135	yn102 H 106
136	u107n

100n4

100	1,000000000000
101	3,000000000000
102	0,500000000000
103	0,250000000000

110u

x=0,25

y=5,24560

Näide 3. Arvutame kahe kümnendmootmelise vektori  $\bar{a} = (a_0, a_1, \dots, a_9)$  ja  $\bar{b} = (b_0, b_1, \dots, b_9)$  skalaarkorrutise

$$(\bar{a}, \bar{b}) = \sum_{i=0}^9 a_i b_i.$$

Asetsegu vektori  $\bar{a}$  koordinaadid pesades 800 ... 809 ja vektori  $\bar{b}$  koordinaadid pesades 810 ... 819. Kogume korrutiste  $a_i b_i$  summa pessa 100, pesa 101 aga kasutame käskude formeerimiskonstandi moodustamiseks. Üheaegselt on see pesa tsükli loendaja.

Selgitame mõnda käsku. Pesa 101 puhastatakse käsuga 121 ning käsus 126 liidetakse talle kümme korda kaheteistkümnenda järgu üks (esimese aadressi viimase järgu üks). Selliselt saadud formeerimiskonstant viiakse esimesse pessa käsuga 122. Konstandi võiks moodustada kohe esimesse pessa, sest selles näites rohkem esimest pesa ei kasutata. Suuremates programides aga, kus esimeses pesas on vaheldumisi erinevad formeerimiskonstandid, on vaja neid säilitada nagu selles näites. Tsükli lõppu kontrollitakse käskudega 127 ... 129.

120k12

120 n0k100  
 121 n0k101  
 122 n101 H 1  
 123 n800 H 102+  
 124 yn810 H 102+  
 125 cn102 H 100  
 126 c2048k101  
 127 a<sub>1</sub>11n  
 128 a<sub>1</sub>10n  
 129 e<sub>1</sub>16376n ≠  
 130 nn100 H 5  
 131 k101 H

800n20

800	1,0000000000	810	1,0000000000
801	2,0000000000	811	-1,0000000000
802	3,0000000000	812	2,0000000000
803	4,0000000000	813	-2,0000000000
804	5,0000000000	814	3,0000000000
805	6,0000000000	815	-3,0000000000
806	7,0000000000	816	4,0000000000
807	8,0000000000	817	-4,0000000000
808	9,0000000000	818	5,0000000000
809	10,0000000000	819	6,0000000000

120u

95,00000

## §5. ÜLEMINEK ÜHEST ARVUTI REŽIIMIST TEISE

Alati, kui ühest režiimist teise üleminekul tahetakse ülesande lahendamise katkestada ja pärast uuesti jätkata, peab selleks vajutama alla ja tooma üles klahvi "ПОЛУАВТОМАТИЧЕСКИЙ"

Üleminekuks universaalrežiimist arvutusrežiimi on vaja:

- vajutada alla klahv "УНИВЕРСАЛЬНЫЙ-СЧЕТНЫЙ";
- vajutada klahvile "ПУСК 1"; masin trükkib käsuloendaja sisu ja сс.

Üleminekuks arvutusrežiimist universaalrežiimi on vaja:

- kui tuleb jätkata poolelijäänud universaalrežiimi programmi täitmist, trükkida yy ning tuua tagasi klahv "УНИВЕРСАЛЬНЫЙ-СЧЕТНЫЙ";
- kui tuleb pöörduda universaalrežiimi lähteseisundisse (DII algusesse), tuua tagasi klahv "УНИВЕРСАЛЬНЫЙ-СЧЕТНЫЙ" ja vajutada klahvile "ПУСК 1".

Üleminekuks universaalrežiimist mälu väljatrüki režiimi

on vaja:

- vajutada klahvile: "ВЫДАЧА ПАМЯТИ"; masin trükkib bb ja käsuloendaja sisu, mis üheaegselt säilitatakse operatiivmälus.

Üleminekuks mälu väljatrüki režiimist universaalrežiimi

on vaja:

- kui tuleb jätkata poolelijäänud universaalrežiimi programmi, trükkida yy;
- kui tuleb pöörduda universaalrežiimi lähteseisundisse (DII algusesse), kontrollida, kas põleb lamp "УНИВЕРСАЛЬНЫЙ" ja vajutada klahvile "ПУСК 1".

Üleminekuks mälu väljatrüki režiimist arvutusrežiimi on

vaja:

- kui peale arvutusrežiimi on vaja jätkata poolelijäänud trükkimist mälu väljatrüki režiimis, trükkida сс;
- ülejäänud juhtudel vajutada alla klahv "УНИВЕРСАЛЬНЫЙ-СЧЕТНЫЙ" ja vajutada klahvile "ПУСК 1".

Üleminekuks arvutusrežiimist mälu väljatrüki režiimi on

vaja:

- kui peale mälu väljatrüki on vaja jätkata poolelijäänud

tööd arvutusrežiimis, trükkida bb;  
- ülejäänud juhtudel vajutada klahvi "ВЫДАЧА ПАМЯТИ".  
Märkus. Kui üleminekul universaalrežiimist arvutus- või mä-  
lu väljatrüki režiimi põlesid lambid "КОМАНДА ВВОД" või  
"КОМАНДА ВЫВОД", siis ei soovitata pöörduda tagasi univer-  
saalrežiimi tähtedega yy(ei täideta lugemis- või pöördumis-  
käsku, millel masin peatus).

## §6. TÖÖ PERFORAATORI JA TRANSMITTERIGA

### 6.1. AP-programmi perforeerimine

Perforeerimiseks on kaks võimalust.

I. Masinas oleva programmi võib igal momendil - peale sisse-  
viimist, peale töötamist, töö vaheajal, peale paranduste  
sisseviimist, välja perforeerida nii:

1.1. Programmis olgu operaator

ПРОГРАММА 14440

Kui arvuti jõuab selle operaatorini, jääb ta seisma.

1.2. Trükkimasina all vajutada sisse klahvid

" ВКЛ.

" РАБОТА

ПЕРФОРАТОРА"

ПЕРФОРАТОРА"

1.3. Nairii puldil vajutada sisse klahv

"ВЫВОД НА ПЕРФОРАЦИЮ"

tuua muud klahvid välja;

vajutada klahvile "ПУСК 2".

1.4. Oodata, kuni masin lõpetab perforeerimise.

1.5. Taastada trükkimasina ja arvuti klahvistiku esialg-  
ne seis.

Märkus. Kui programmis operaatorit ПРОГРАММА 14440 ei ole,  
võib teha nii:

vajutada klahvile " ПУСК 1";

lüüa trükkimasinal 14440 u;

täita punktid 1. 2. - 1. 5.

II. Teine võimalus (programmi trükkimine kocs perforeerimi-  
sega).

2.1. Trükkimasina all vajutada sisse klahvid



algne seis

3.7. Lasta transmitteri õlg alla; vabale panna tükk linti

### 6.3. Ülesande lahendamine ja algandmete perforeerimine

Ülesande lahendamiseks on vaja (kui perforeerite esimese meetodi järgi)

vajutada klahvile "ПУСК 1"

trükkida "na"

trükkida "ИСПОЛНИМ N", kus N on esimesena täidetava operatsiooni number.

Operaatoriga "ВВЕДЕМ" sisseviidavaid algandmeid võib samuti perforeerida. Lahendamisel tuleb lint asetada transmitterile nii, et fotodiiodide all oleks parajasti esimesele arvumärgile eelnev sünkroauk. Teise meetodiga perforeerimisel näiteks on seda kõige mugavam teha nii, et algandmed lüüakse kohe peale programmi. Kui lindi lõppu on perforeeritud "ucN" ja reavahetus, asub masin kohe peale programmi sisseviimist lahendama.

Märkus. Kui peale programmi või algandmete sisseviimist transmitter ei peatu, tuleb:

tuua tagasi klahv "ВВОД С ТРАНСМИТЕРА"

tõsta transmitteri õlg

tuua tagasi klahv "ВКЛ. ТРАНСМИТЕРА"

vajutada klahvile "ПУСК 1" edasise informatsiooni sissesteriseks.

### 6.4. Väljaperforeerimine režiimis "bb"

Perforeerimiseks on vaja:

- veenduda, et klahv "РАБОТА ПЕРФОРАТОРА" on vajutatud;
  - anda 10 cm tühja linti (vt. punkt 6.1.);
  - vajutada klahvile "ВЫДАЧА ПАМЯТИ" (klahv "ВЫВОД НА ПЕЧАТЬ" olgu sees);
  - kui tahetakse korruga perforeerida ja trükkida, siis vajutada klahv "ВЫВОД НА ПЕРФОРАЦИЮ" sisse;
  - trükkida;
  - väljaviidava massiivi esimene aadress üks tähtedest k, m, r, n, g
- aadresside arv,

täht "n",

täht "k",

reavahetus;

- kui tuleb ainult perforeerida, siis peale tähe "k" trükkimist tuua välja klahv "ВЫВОД НА ПЕЧАТЬ", teha reavahetus ja vajutada klahvile "ПУСК 2".

Perforeeritakse välja esimene aadress ja vastav täht ning pärast nõutav informatsioon; edasi perforeeritakse tähed ш ja н ja 35 korda tühiku kood.

Pärast perforeerimise lõppu tuleb anda tühja linti ja tuua välja klahv "ВЫВОД НА ПЕРФОРАЦИЮ

Nii väljaperforeeritud informatsioon on valmis uuesti masinasse sisseviimiseks. Täht н peatab transmitteri peale sisseviimist.

Märkus. Kui peale tähte "n" mitte trükkida tähte "k", siis lõpus tähte н ei perforeerita.

Seega peale sisseviimist transmitter ei peatu. Programmi sisseviimine toimub samuti nagu punktis 6.2.

#### 6.5. Transmitter ja perforaator universaalrežiimis

Informatsiooni sisestamiseks koos perforeerimisega on vaja:

- vajutada sisse klahvid "РАБОТА ПЕРФОРАТОРА" ja "ВКЛ. ПЕРФОРАТОРА" trükkali korpusel;
- anda tühja linti (vt. 6.1.);
- vajutada sisse klahvid "ПЕЧАТЬ, ПЕРФОРАЦИЯ" ja "ВЫВОД НА ПЕЧАТЬ" arvuti puldil;
- vajutada klahvile "ПУСК 1";
- viia sisse nõutav informatsioon, nagu tavaliselt.

Kui tahetakse kohe minna ülesande lahendamisele, trükkida programmi lõpus täht "ш", programmi esimene aadress ja täht "u". Vastupidisel juhul vajutada enne tähte "ш" alla klahv "СВЯСЬ-АВТОНОМНЫЙ".

Kui peale lindi sisseviimist ei taheta suunata masinat lahendama, trükkida autonoomses režiimis "шн".

Lindi lõpus on soovitatav trükkida 10 cm tühiku koodi.

Lõpuks tuleb anda tühja linti ja taastada klahvistiku esialgne asend.

Informatsiooni sisestamiseks transmitterilt talitada punkti 6.2. järgi.

Märkused. 1. Kui lindi alguses ei ole perforeeritud esimest aadressi:

vajutada klahvile "ПУСК 1"

trükkida esimene aadress ja vastav täht;

vajutada alla klahv "ВВОД С ТРАНСМИТТЕРА"

vajutada klahvile "ПУСК 2"

2. Et sisestamisel transmitteriga vältida trükkimist, tuleb tuua välja klahv "ВЫВОД НА ПЕЧАТЬ"

3. Masina jaoks mõttetu informatsiooni sisestamisel arvuti peatub ja väljundregistrisse antakse täpne kood.

4. Lahendustulemuste trüki blokeerimiseks tuleb välja tuua klahvid " ВКЛ. ТРАНСМИТТЕРА " ja "ВЫВОД НА ПЕЧАТЬ" ning sisse vajutada klahv " ВВОД С ТРАНСМИТТЕРА "



Arvuti "Neirii-c" käskude modifikatsioonid

M <sub>1</sub> M <sub>2</sub>	mantiss																järk																Liikuvu koma arv					
	131072	65536	32768	16384	8192	4096	2048	1024	512	256	128	64	32	16	8	4	2	1	131072	65536	32768	16384	8192	4096	2048	1024	512	256	128	64	32	16		8	4	2	1	
36	35	34	33	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	Järkude numbrid		
0	0	0	operatsiooni kood																A <sub>2</sub> - O3Y address																			
0	0	0	A <sub>1</sub> - E3Y või O3Y address																A <sub>2</sub> ≤ 1023																H			
0	0	1	A <sub>1</sub> ⇒ summaator 1...14 j.																A <sub>2</sub> ≤ 1023																K			
0	1	0	A <sub>1</sub> ⇒ summaator 1...18 j.																A <sub>2</sub> ≤ 127																Π			
0	1	1	A <sub>1</sub> ⇒ summaator 19...36 j.																A <sub>2</sub> ≤ 127																Π			
1	1	0																	m ≤ 15																tingimused			
1	1	0	A <sub>1</sub> - O3Y või D3Y address																A <sub>2</sub> ≤ 15																H			
1	1	0	A <sub>1</sub> ⇒ summaator 1...14 j.																																Π			
1	1	1	A <sub>1</sub> ⇒ summaator 32...19 j.																																Π			
																																						modifikatsioonid
																																						tingimuslike käskude modifikatsioonid

\* Arv A<sub>1</sub> vastavalt summaatori kaheksandkohtadele modifikatsiooni Π jaoks.

\*\* Arv A<sub>1</sub> vastavalt summaatori kaheksandkohtadele modifikatsiooni K ja Π jaoks.

Näiteks eraldatase kümnes järk koodist (A<sub>2</sub>) käsuga 1512ka<sub>2</sub>; kolmekümnes järk koodist (A<sub>2</sub>) eraldatase käsuga 120451A<sub>2</sub>.

Masinaoperatsioonid

Operatsiooni välis- ja sisekood*	Operatsiooni nimetus	Käsu kood	Tehtav tehe	Märkused
1	2	3	4	5
c (01)	liitmine	cA <sub>1</sub> QA <sub>2</sub>	(A <sub>2</sub> ) + [A <sub>1</sub> ] 0 ⇒ A <sub>2</sub> ,s <sub>2</sub>	Tulemuse ületäitumist ei kontrollilita
c <sub>1</sub> (41)	"	c <sub>1</sub> A <sub>1</sub> QA <sub>2</sub>	(s <sub>2</sub> ) + [A <sub>1</sub> ] 0 ⇒ A <sub>2</sub> ,s <sub>2</sub>	
b (02)	lahutamine	bA <sub>1</sub> QA <sub>2</sub>	(A <sub>2</sub> ) - [A <sub>1</sub> ] 0 ⇒ A <sub>2</sub> ,s <sub>2</sub>	
b <sub>1</sub> (42)	"	b <sub>1</sub> A <sub>1</sub> QA <sub>2</sub>	(s <sub>2</sub> ) - [A <sub>1</sub> ] 0 ⇒ A <sub>2</sub> ,s <sub>2</sub>	
y (03)	korrutamine	yA <sub>1</sub> QA <sub>2</sub>	(A <sub>2</sub> ) x [A <sub>1</sub> ] 0 ⇒ A <sub>2</sub> ,s <sub>2</sub>	
y <sub>1</sub> (43)	"	y <sub>1</sub> A <sub>1</sub> QA <sub>2</sub>	(s <sub>2</sub> ) x [A <sub>1</sub> ] 0 ⇒ A <sub>2</sub> ,s <sub>2</sub>	
g (04)	jagamine	gA <sub>1</sub> QA <sub>2</sub>	(A <sub>2</sub> ) : [A <sub>1</sub> ] 0 ⇒ A <sub>2</sub> ,s <sub>2</sub>	
g <sub>1</sub> (44)	"	g <sub>1</sub> A <sub>1</sub> QA <sub>2</sub>	(s <sub>2</sub> ) : [A <sub>1</sub> ] 0 ⇒ A <sub>2</sub> ,s <sub>2</sub>	
c <sub>2</sub> (11)	liitmine	c <sub>2</sub> A <sub>1</sub> QA <sub>2</sub>	(A <sub>2</sub> ) + [A <sub>1</sub> ] 0 ⇒ A <sub>2</sub> ,s <sub>2</sub>	
c <sub>3</sub> (51)	"	c <sub>3</sub> A <sub>1</sub> QA <sub>2</sub>	(s <sub>2</sub> ) + [A <sub>1</sub> ] 0 ⇒ A <sub>2</sub> ,s <sub>2</sub>	
b <sub>2</sub> (12)	lahutamine	b <sub>2</sub> A <sub>1</sub> QA <sub>2</sub>	(A <sub>2</sub> ) - [A <sub>1</sub> ] 0 ⇒ A <sub>2</sub> ,s <sub>2</sub>	
b <sub>3</sub> (52)	"	b <sub>3</sub> A <sub>1</sub> QA <sub>2</sub>	(s <sub>2</sub> ) - [A <sub>1</sub> ] 0 ⇒ A <sub>2</sub> ,s <sub>2</sub>	
y <sub>2</sub> (13)	korrutamine	y <sub>2</sub> A <sub>1</sub> QA <sub>2</sub>	(A <sub>2</sub> ) x [A <sub>1</sub> ] 0 ⇒ A <sub>2</sub> ,s <sub>2</sub>	

Tulemust kontrollitakse ületäitumise suhtes

Järg	1	2	3	4	5
$\gamma_3$ (53)		korrutamine	$\gamma_3 A_1 Q A_2$	$(a_2) \times [A_1] \theta \Rightarrow A_2, a_2$	
$\delta_2$ (14)		jagamine	$\delta_2 A_1 Q A_2$	$(A_2) : [A_1] \theta \Rightarrow A_2, a_2$	
$\delta_3$ (54)		"	$\delta_3 A_1 Q A_2$	$(a_2) : [A_1] \theta \Rightarrow A_2, a_2$	
$a$ (05)		aritmeetiline ne nihe pä- remale	$a A_1 Q A_2$	$(A_2) \text{ nihe } [A_1] \theta \Rightarrow A_2, a_2$	
$a_1$ (45)		"	$a_1 A_1 Q A_2$	$(a_2) \text{ nihe } [A_1] \theta \Rightarrow A_2, a_2$	
$\sigma$ (06)		loogiline ni- he vasakuile	$\sigma A_1 Q A_2$	$(A_2) \text{ nihe } [A_1] \theta \Rightarrow A_2, a_2$	
$\sigma_1$ (46)		"	$\sigma_1 A_1 Q A_2$	$(a_2) \text{ nihe } [A_1] \theta \Rightarrow A_2, a_2$	
$\Gamma$ (23)		pika arvu aritmeetiline nihe paremale	$\Gamma A_1 Q A_2$	$(A_2) \text{ nihe } [A_1] \theta \Rightarrow A_2$	$0 \neq \Gamma$
$\Gamma_1$ (63)		pika arvu loogiline ni- he vasakuile	$\Gamma_1 A_1 Q A_2$	$(A_2) \text{ nihe } [A_1] \theta \Rightarrow A_2$	
$n$ (10)		arvu ülekanne	$n A_1 Q A_2$	$[A_1] \theta \Rightarrow A_2, a_2$	
$n_1$ (50)		"	$n_1 A_1 Q A_2$	$(a_2) \Rightarrow A_2, a_2$	
$H$ (07)		normaliseerimise- rimine	$H A_1 Q A_2$	$(A_1) \text{ norm.} \Rightarrow A_2$	
$H_4$ (47)		"	$H_4 O Q A_2$	$(a_2) \text{ norm.} \Rightarrow A_2$	
$l$ (15)		loogiline liitmine	$l A_1 Q A_2$	$(A_2) \vee [A_1] \theta \Rightarrow a_2$	

Kui normaliseerimine on rikutud vasakuile, on nihete arv positiivne, vastasel juhul negatiivne; nihete arv kirjutatakse täiendkoodis pessa ( $a_2$ )

OVO=0, OV1=1,



1	2	3	4	5
k (37)	peatus	$kA_1Q$	$[A_1] \theta \Rightarrow CM$ ja peatus	
$k_1$ (77)	peatus	$k_1OQ$	$(a_2) \Rightarrow CM$ ja peatus	
x (00)	tühi operat- sioon	x või x00	$(KAL) \Rightarrow KAL$	$(a_2)$ säilib
o (21)	väljatrükk või perforree- rimine	$oA_1Q$	$A_1 \Rightarrow$ trükk	$(a_2)$ säilib
$o_1$ (61)	"	$o_1A_1Q$	$(a_2) + 2^{11} + 2^7 \Rightarrow$ trükk	
r (24)	sisseviimi- ne trükke- lilt või transmitte- rilt	$rQA_2$		sümboli sisekood saade- takse pessa $A_2$

Märkused \* Operatsiooni sisekood antud kaheksandsüsteemis

\*\* KAL - käsu aadressi loendaaja

Välis- kood	Nimetus	Töö- pesad	Käskkood	Tehtav tehe	Märkused
1	2	3	4	5	6
Pseudooperatsioonid tööks liikuva koma arvudega.					
cn	liitmine	3...8	cnA <sub>1</sub> QA <sub>2</sub>	$(A_2) + [A_1] \theta \Rightarrow A_2, a_2$	Ületäitumisel trü- kib masin $\infty$ ja peatub.
bn	lahutamine	3...8	bnA <sub>1</sub> QA <sub>2</sub>	$[A_1] \theta - (A_2) \Rightarrow A_2, a_2$	
ob	vastupidine lahutamine	3...8	obA <sub>1</sub> QA <sub>2</sub>	$(A_2) + [A_1] \theta \Rightarrow A_2;$	
yn	korrutamine	3...8	ynA <sub>1</sub> QA <sub>2</sub>	$(A_2) \times [A_1] \theta \Rightarrow A_2, a_2$	
gn	jagamine	3...8	gnA <sub>1</sub> QA <sub>2</sub>	$[A_1] \theta : (A_2) \Rightarrow A_2, a_2$	
og	vastupidine jagamine	3...8	ogA <sub>1</sub> QA <sub>2</sub>	$(A_2) : [A_1] \theta \Rightarrow A_2, a_2$	Ületäitumisel või jaga ja võrdumisel 0-ga trükitab masin ja peatub.
bM	moodulite lahutamine	3...8	bM A <sub>1</sub> QA <sub>2</sub>	$ (A_2)  -  [A_1] \theta  \Rightarrow A_2, a_2$	
kn	$\sqrt{\quad}$ arvutamine	3...7	knA <sub>1</sub> QA <sub>2</sub>	$\sqrt{[A_1] \theta} \Rightarrow A_2;$	Kui $[A_1] \theta < 0$ trükitab masin $x < 0$ ja peatub.
ln	ln arvutamine	3...7	lnA <sub>1</sub> QA <sub>2</sub>	$\ln [A_1] \theta \Rightarrow A_2;$	Kui $[A_1] \theta \leq 0$ trükitab masin $x \leq 0$ ja peatub.
lg	lg arvutamine	3...7	lgA <sub>1</sub> QA <sub>2</sub>	$\lg [A_1] \theta \Rightarrow A_2$	Soovitatav on võtta $A_1 = A_2$ kui $\theta = H$
ex	$e^x$ arvutamine	3...8	exA <sub>1</sub> QA <sub>2</sub>	$e^{[A_1] \theta} \Rightarrow A_2$	Kui $[A_1] \theta > 43$ trükitab masin $\infty$ ja peatub.

Järg	1	2	3	4	5	6
cn	sin arvutamine	3...9	sn <sub>A<sub>1</sub></sub> QA <sub>2</sub>	sin [A <sub>1</sub> ] θ ⇒ A <sub>2</sub>	Kui [A <sub>1</sub> ] θ järk > 27, trükkib masin n="järg" ja peatub.	
tg	tg arvutamine	3...9	tga <sub>A<sub>1</sub></sub> QA <sub>2</sub>	tg [A <sub>1</sub> ] θ ⇒ A <sub>2</sub>	Kui [A <sub>1</sub> ] θ = ± 27 trükkib masin ∞ ja peatub.	
as	arcsin arvutamine	3...7	asa <sub>A<sub>1</sub></sub> QA <sub>2</sub>	arcsin [A <sub>1</sub> ] θ ⇒ A <sub>2</sub>	Kui [A <sub>1</sub> ] θ > 1 trükkib masin x > 1 ja peatub.	
ec	arccos arvutamine	3...7	aca <sub>A<sub>1</sub></sub> QA <sub>2</sub>	arccos [A <sub>1</sub> ] θ ⇒ A <sub>2</sub>		
at	arctg arvutamine	3...7	ata <sub>A<sub>1</sub></sub> QA <sub>2</sub>	arctg [A <sub>1</sub> ] θ ⇒ A <sub>2</sub>		
gn	liikuva koma arvude normaaliseerimine	1,3...9	gna <sub>A<sub>1</sub></sub> QA <sub>2</sub>			
nn	liikuva koma arvude trükk	3...9, 16, 17, 20...26	nna <sub>A<sub>1</sub></sub> QA <sub>2</sub>			
ym	täisarvude korrutamise	1...9, 16, 17	yma <sub>A<sub>1</sub></sub> QA <sub>2</sub>	(A <sub>2</sub> ) x [A <sub>1</sub> ] θ ⇒ A <sub>2</sub>	Kui tulemus on absooluutväärtuselt suurem kui 235 trükkib masin ∞ ja peatub.	
gm	täisarvude jagamine	3...10	gma <sub>A<sub>1</sub></sub> QA <sub>2</sub>	(A <sub>2</sub> ) : [A <sub>1</sub> ] θ ⇒ A <sub>2</sub>	Kui [A <sub>1</sub> ] θ = 0 trükkib masin ∞ ja peatub.	
nm	täisarvude trükk	nna <sub>A<sub>1</sub></sub> Q				

1	2	3	4	5	6
cg	pikkade arvu- de liitmine	3...8	cgA <sub>1</sub> QA <sub>2</sub>	$(A_2)^E + (A_1)^E \Rightarrow A_2^E$	
bg	pikkade arvu- de lahutamine	3...8	bgA <sub>1</sub> QA <sub>2</sub>	$(A_2)^E - (A_1)^E \Rightarrow A_2^E$	Ületäitumisel trü- kib masin $\infty$ ja peatub.
vg	pikkade arvu- de korrutamine	3...9, 16,17, 20-23	vgA <sub>1</sub> QA <sub>2</sub>	$(A_2)^E \times (A_1)^E \Rightarrow A_2^E$	
eg	pikkade arvu- de jagamine	2...9, 16,17, 20-23	egA <sub>1</sub> QA <sub>2</sub>	$(A_2)^E : (A_1)^E \Rightarrow A_2^E$	
om	pika arvu saatmine	3...8	omA <sub>1</sub> QA <sub>2</sub>	$(A_1)^E \Rightarrow A_2^E$	
he	pika arvu nor- maliseerimine	3...8	heA <sub>1</sub> QA <sub>2</sub>		
ng	pika arvu trükk	3...9, 16,17, 20,21	ngA <sub>1</sub> Qq		
ck	kompleksarvu- de liitmine	3...8, 16,17, 20-23	ckA <sub>1</sub> QA <sub>2</sub>	$(A_1)^K + (A_2)^K \Rightarrow A_2^K$	
bk	kompleksarvu- de lahutamine	3...8, 16,17, 20-23	bkA <sub>1</sub> QA <sub>2</sub>	$(A_1)^K - (A_2)^K \Rightarrow A_2^K$	
yk	kompleksarvu- de korrutamine	3...8, 16,17, 20...23	ykA <sub>1</sub> QA <sub>2</sub>	$(A_1)^K \times (A_2)^K \Rightarrow A_2^K$	
ek	kompleksarvu- de jagamine	25,26	ekA <sub>1</sub> QA <sub>2</sub>	$(A_1)^K : (A_2)^K \Rightarrow A_2^K$	

Järg	1	2	3	4	5	6
------	---	---	---	---	---	---

nr	murdarvu trükk	3...9, 16,17	nrA <sub>1</sub> Q
nk	käsu trükk	3...9	nkA <sub>1</sub> Q
nc	sisu trükk	3...7	ncA <sub>1</sub> Q
nu	arvuliste in- deksite trükk	3...8	nuA <sub>1</sub> Q

[A<sub>1</sub>] trükitakse  
kättekoodina.

Kui [A<sub>1</sub>] < 0  
või [A<sub>1</sub>] > 1000  
trükitakse masin \* nu  
ja peatub.

Trükimasina "Consul" sümboleite sisend-  
koodide aadressid (käsu "O" jaoks).

Arvud		Astmenäita- jad		Indeksid	
0	2176	0	2186	0	2201
1	2177	1	2187	1	2202
2	2178	2	2188	2	2203
3	2179	3	2189	3	2204
4	2180	4	2190	4	2205
5	2181	5	2191	5	2206
6	2182	6	2192	6	2207
7	2183	7	2193	7	2208
8	2184	8	2194	8	2209
9	2185	9	2195	9	2210

## T ä h e d

π	2211	δ	2222	t	2233	ю	2244
c	2212	Γ	2223	и	2234	э	2245
b	2213	ч	2224	к	2235	i	2246
y	2214	u	2225	ъ	2236	j	2247
g	2215	o	2226	ñ	2237	k	2248
л	2216	e	2227	b	2238	n	2249
M	2217	l	2228	φ	2239	indeksid	
z	2218	p	2229	ц	2240		
H	2219	m	2230	ш	2241	i	2198
x	2220	ω	2231	f	2242	j	2199
a	2221	ς	2232	я	2243	k	2200

## M ä r g i d

=	2250	l	2256	√	2262	rea-	
≠	2251	∞	2257	(	2263	vahetus	2268
>	2252	+	2258	)	2264	tühik	2270
»	2253	-	2259	,	2265	rea	
<	2254	x	2260	×	2266	algus	2274
≤	2255	/	2261	!	2267		

AP-programmi mälujaotus

1) Muutujatele reserveeritud pesad AP-s.

П	с	β	у	г	л	м	з	н	х	а	δ	г	ч	у	о
32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
е	l	p	m	ш	s	t	Ы	i	j	k	n	i	j	k	(indeksid)
48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	

- 2) Operaatori programmide algusaadresside massiiv 69...200.
- 3) Liikuva koma konstandid pesades 201...325. Selle massiivi esimese vaba pesa aadress on pesas 19.
- 4) Tööprogramm asub alates pesast 370. Esimese vaba pesa aadress on pesas 64.
- 5) Kohe peale tööprogrammi on operaatoriga "ХРАНИМ" hoitava funktsiooniväärtuste massiiv. Selle algusaadress on pesas 1006 ja pikkus pesas 1007.
- 6) Arvu massiivid asuvad pesades 980 - M ...980. Pesas 67 on arv 981 - M; siin M on massiivide üldpikkus.
- 7) AP-programm kasutab tööpesi 13 - 18, 20...31, 63...67, 326...329, 999...1003, 1006...1007. AP-programm kasutab tööpesi. 20, 21, 326...369.

Programmide poolt kasutatavad tööpesad

Programmi nimetus	Tööpesad
BB (mälu väljatrükk)	2, 3, 18, 1018 + 1020
an	4, 5, 13, + 18, 20 + 369
DII (informatsiooni sisseviimine ja töötlemine)	1, 6, 7, 16 + 28, 1000, 1001
Pöördmaatriksi ja determinandi arvutamine	14, 32 + 60, 61 + 61 + 5n <sup>2</sup> + 3n, 980
CY (lineaarne võrrandisüsteem)	32 + 63, 64 + 64 + n <sup>2</sup> + 3n, 980, 993, 1022
DY (diferentsiaalvõrrandite süsteem)	15, 20, 33 + 54, 65 + 980
Maatriksite korrutamine	15, 30 + 43, 45 + 47
AY (polünoomi juurte arvutamine)	835 + 836 + 3n + 4, 954 + 980, 993, 1017
Integraal	15, 20, 41, 60, 61, 65, 67 + 370, 980
cc (arvutusrežiim)	2, 9, 16 + 18, 23, 25, 981 + 990, 998 + 1003, 1005, 1007 + 1016, 1018

## S i s u k o r d

Sissejuhatus	3
I Ülevaade arvutist "Nairii-C"	3
§ 1 Arvuti struktuur	3
1.1. Aritmeetiline seade	3
1.2. Juhtseade	6
1.3. Mäluseade	6
1.4. Sisend- ja väljundseadmed	6
1.5. Juhtpult ja signalisatsioonipaneel	7
§ 2 Andmete sisestamine arvutisse "Nairii-c"	7
§ 3 Andmete väljatrükkimine mälust	9
II Programmeerimine arvutil "Nairii-C"	11
§ 1 Arvutusrežiim	11
1.1. Elementaarfunktsioonide arvutamine	12
1.2. Aritmeetilised tehted	13
1.3. Tehted tähtede arväärtustega	14
§ 2 Tüüpülesannete lahendamine	17
1.2. Algebraalse võrrandi lahendamine	18
2.2. Määratud integraali arvutamine	19
2.3. Lineaarse algebraise võrrandisüsteemi lahendamine	20
2.4. Maatrikseid käsitlevad programmid	20
2.5. Maatriksite korrutamine	23
2.6. Esimest järku diferentsiaalvõrrandite süsteemi lahendamine	23
§ 3 Automaatprogrammeerimine	26
3.1. Üldkirjeldus	26
3.2. Avaldised	28
3.3. Operaatorid	30
3.4. Operaatorite kirjeldus	31
3.5. Näited	37

3.6.	Indeksitega muutujad	40
3.7.	Indeksiga muutujad operaatorites	42
3.8.	Näited	43
3.9.	Andmeid translaatori töö kohta	45
3.10.	Vigade leidmine ja parandamine	48
3.11.	AP-programmi sisseviimine ja ülesande lahendamine	48
3.12.	Näited	49
§ 4	Arvud masinas. Käskude süsteem	51
4.1.	Arvude kujutamine	51
4.2.	Käskude süsteem	53
4.3.	Käskude sise- ja väliskoodid	53
4.4.	Tingimusteta käskude modifikatsioonid	54
4.5.	Tingimuslikud käsud	56
4.6.	Tingimuslike käskude modifikatsioonid	56
4.7.	Käskude formeerimine	57
4.8.	Tingimuslikud tähistused	57
4.9.	Masinaoperatsioonid	58
4.10.	Aritmeetilised operatsioonid	59
4.11.	Nihked	59
4.12.	Loogilised operatsioonid	60
4.13.	Operatsioonid koodidega	61
4.14.	Juhtimisoperatsioonid	61
4.15.	Välisseadmete poole pöördumise käsud	61
4.16.	Pseudooperatsioonid	61
4.17.	Mõningate pseudooperatsioonide selgitus	64
4.18.	Trükkimise pseudokäsud	65
4.19.	Näited	66
§ 5	Üleminek ühest arvuti režiimist teise	70
§ 6	Töö perforaatori ja transmitteriga	71
6.1.	AP-programmi perforeerimine	71
6.2.	Programmi sisseviimine transmitteriga	72
6.3.	Ülesande lahendamine ja algandmete perforeerimine	73
6.4.	Väljaperforeerimine režiimis "bb"	73
6.4.	Transmitter ja perforaator univeraalrežiimis	74

Lisad	76
Lisa 1	
Arvuti "Nairii-C" käskude mofidikatsioonid	77
Lisa 2	
Masinaoperatsioonid	78
Lisa 3	
Pseudooperatsioonid	81
Lisa 4	
Trükimasina "Consul" sümbolite sisendkoodide aadressid	86
Lisa 5	
AP-programmi mälujaotus	87
Lisa 6	
Programmide poolt kasutatavad tööpesad	88





Hind 16 kop.

A

33431

TÜ RAAMATUKOGU



1 0300 01129629 2