

**ioRīgas Tehniskā Universitāte**

**Datorzinātnes un Informācijas Tehnoloģijas fakultāte**

**Automātika un datortehnika**

**Risinājumu algoritmizācija un programmēšana (1. daļa)**

**Laboratorijas darbs#6(a) Divdimensiju masīva apstrāde**

**DITF**

**RDBF0 1. kurss 9. grupa**

**Viktorija Ovčinnikova**

**studenta apl. nr. 101RDB131**

Darba izpildes grafiks			
	Protokola sagatave	Darbs ar datoru	Ieskaite
Pēc plāna (nod.)			
Faktiski (nod.)			

## 1. Darba uzdevums

Izstrādāt programmu, kas divdimensiju masīvā aizpilda norādātos elementus (skat. 1. attēlu) ar vērtībām 1, 2, 3, 4, 5. . . . pa vertikāli.

{	.	.	.	.	.	.	.	.	.	.	}
{	.	.	.	.	.	.	.	.	.	.	}
{	1	4	8	13	18	24	31	39	47	54	}
{	2	5	9	14	19	25	32	40	48	55	}
{	3	6	10	15	20	26	33	41	49	56	}
{	.	7	11	16	21	27	34	42	50	57	}
{	.	.	12	17	22	28	35	43	51	58	}
{	.	.	.	.	23	29	36	44	52	59	}
{	.	.	.	.	.	30	37	45	53	.	}
{	.	.	.	.	.	.	38	46	.	.	}

1. att. Masīva aizpildījums

## 3. Algoritma izstrāde

Lai aizpildītu norādīto masīva daļu ar uzdotām vērtībām vispirms ir nepieciešams noteikt aizpildīšanas shēmu.

Izvēlēsimies izstrādātajā fragmentā izmantojamo mainīgo (identifikatoru) vārdus:

- is** – masīva aizpildāmās rindas sākuma indekss;
- ib** – masīva aizpildāmās rindas beigu indekss;
- js** – masīva aizpildāmās kolonnas sākuma indekss;
- jb** – masīva aizpildāmās kolonnas beigu indekss;
- j** – analizējamā elementa numurs;
- k** – masīva aizpildāmās rindas indekss;

Tā kā aizpildīšana notiek pa vertikāli, tad loģiski ir rīkoties pēc sekojoša algoritma (2. att.):

- 1) nosakām no kuras (**js**) un līdz kurai (**jb**) kolonnai notiks masīva aizpildīšana;
- 2) katrai izvēlētai kolonnai (**j**) noteiksim no kura (**is**) līdz un līdz kuram (**ib**) elementam notiek kolonnas aizpildīšana;

No 2. attēla analīzes redzams, ka :

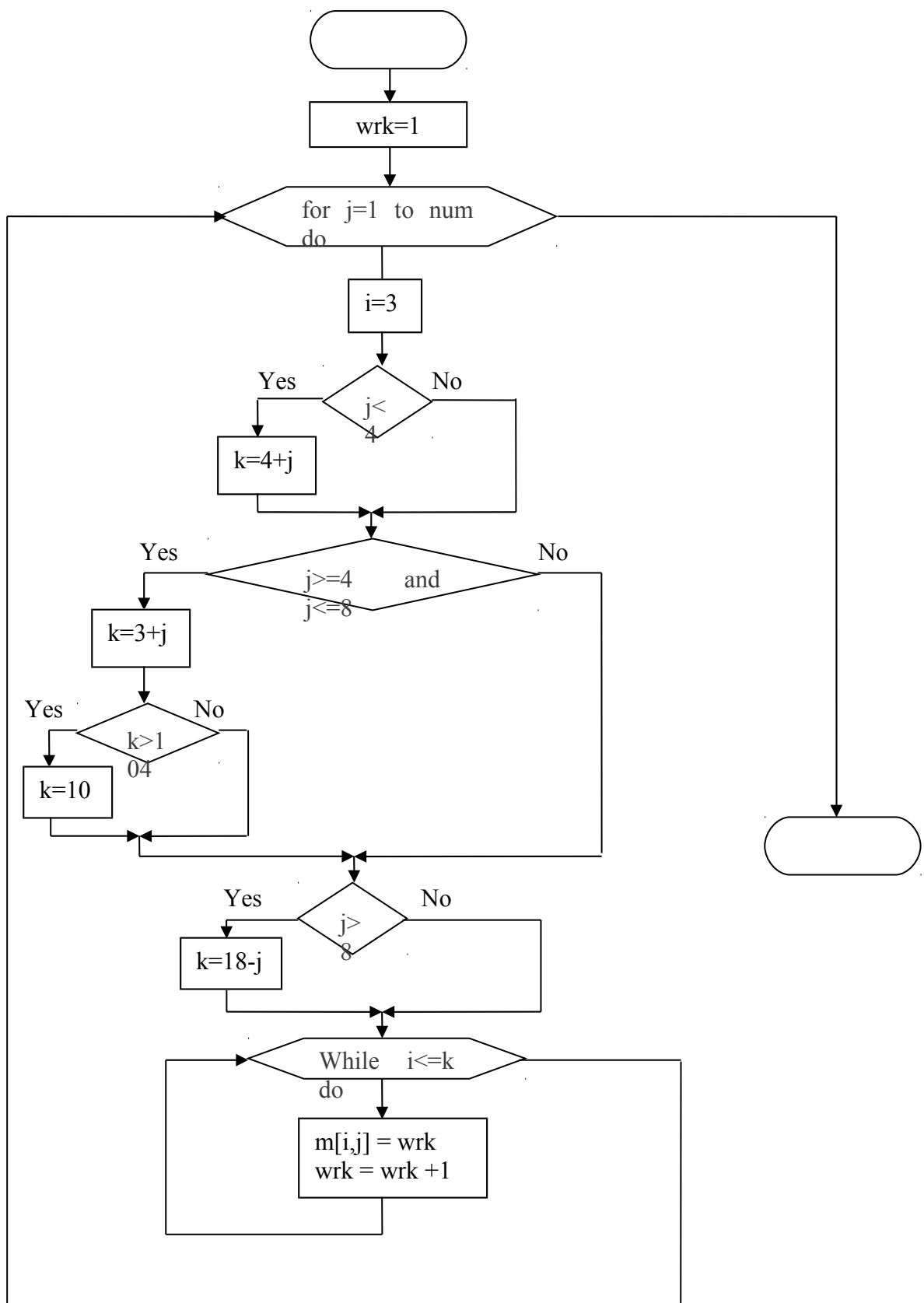
- 1) **is** vērtība, uzdotai kolonnai **j** ir aprēķināma sekojoši:
  - a) **is = 3**;
  - b) ja **js < 4**, tad **k = 4 + j**;
  - c) ja **js >= 4** un **js <= 8**, tad **k = 3 + j**; ja **k > 10**, tad **k = 10**;
  - d) ja **js > 8**, tad **k = 18 - j**;
- 2) **ib <= k**.

		j →									
		1	2	3	4	5	6	7	8	9	10
i ↓	1										
	2										
	3	3, 1	3, 2	3, 3	3, 4	3, 5	3, 6	3,7	3,8	3, 9	3,10
	4	4, 1	4, 2	4, 3	4, 4	4, 5	4, 6	4,7	4,8	4, 9	4,10
	5	5, 1	5, 2	5, 3	5, 4	5, 5	5, 6	5,7	5,8	5, 9	5,10
	6		6, 2	6, 3	6, 4	6, 5	6, 6	6,7	6,8	6, 9	6,10
	7			7, 3	7, 4	7, 5	7, 6	7,7	7,8	7, 9	7,10
	8					8 5	8 6	8,7	8,8	8 9	8,10
	9						9, 6	9,7	9,8	9, 9	
	10							10, 7	10, 8		
		is=3; ib=5	is=3; ib=6	is=3; ib=7	is=3; ib=7	is=3; ib=8	is=3; ib=9	is=3; ib=10	is=3; ib=10	is=3; ib=9	is=3; ib=8

2. att. Masīva aizpildīšanas shēma

Sagatave satur mums līdzīga uzdevuma risinājumu un līdzekļus programmas testēšanai. Tāpēc ir nepieciešams izstrādāt tikai augstāk apskatīto masīva aizpildīšanu.

#### 4. Algoritma blokshēma



3. att. Izstrādātā algoritma shēma

## **5. Testpiemēru kopa**

Uzdevuma izpildes rezultāta masīva aizpildījuma izvadam uz ekrāna jāatbilst 1. attēlā dotajam masīva attēlojumam.

## **6. Programmas fragmenta pirmteksts**

```
wrk:=1;
for j:=1 to num do
begin
  i:=3;
  If (j<4) then k:=4+j;
  If ((j>=4) and (j<=8)) then begin
    k:=3+j; If (k>10) then k:=10;
  End;
  If (j>8) then k:=18-j;
  while(i <= k) do
  begin
    ms[i,j]:=wrk;
    wrk:=wrk+1;
    i:=i+1
  end;
end;
```

## **7. Secinājumi**

Tika izstrādāta programma, kas nodrošina divdimensiju masīva uzdotā apgabala aizpildīšanu ar skaitliskām vērtībām, izveidota darba atskaite. Dotā laboratorijas darba sagatavošanai ir patērētas 4 stundas laika. Daudz laika aizņēma sagatavē dotā programmas pirmteksta izpratnes iegūšana. Izprasto sagataves uzdevumu pielāgot individuālā uzdevuma vajadzībām sevišķas grūtības nesagādāja.