

Kursa „DIP107 Risinājumu algoritmizēšana un programmēšana”

3. MĀJAS DARBA IZPILDES PIEMĒRS

1. Uzdevuma nostādne.

1. Izveidot masīvu A no 40 tipa float skaitļiem.

Ievadīt veselo skaitli K.

- Ja skaitlis K ir vienāds ar nulli vai negatīvs, tad aizpildīt masīvu A ar patvaļīgiem reāliem skaitļiem no diapazona no -100 līdz 100.
- Ja skaitlis K ir lielāks nekā nulle, tad masīva A elementu vērtības aprēķināt izmantojot izteiksmi: $A[i] = (i+1) \cdot K / (K+1)$

2. Izvadīt masīva A elementus ekrānā (pa 20 elementiem vienā rindā). Masīva A elementus izvadīt formātā ar vienu ciparu aiz komata.

3. Apgriezt masīvu otrādāk, lai pirmais elements kļūtu par pēdējo, pēdējais par pirmo un t.t.

4. Izvadīt ekrānā masīva A elementus (pa 20 elementiem vienā rindā). Masīva A elementus izvadīt formātā ar vienu ciparu aiz komata.

Piemēram, ja masīvā A atrodas skaitļi:

1.2 5.4 45.7 74.1 60.0

Tad masīvam A pēc pārveidošanas ir jāizskatās sekojošā veidā:

60.0 74.1 45.7 5.4 1.2

Prasības cikla operatoriem:

<i>Cikla operators, kurš ir jāizmanto norādītā uzdevuma punkta realizācijā</i>			
1. masīva inicializācija	2. masīva izvade	3. masīva apstrāde	4. masīva izvade
for	for	for	for

2. Uzdevuma risināšanas analīze

Ievaddati:

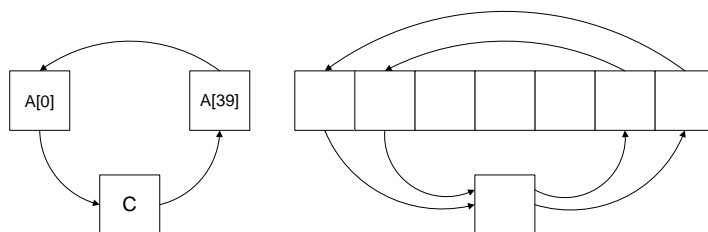
Vesels skaitlis K.

Izvaddati:

- Masīva A nosaukums un masīvs A pirms pārveidošanas (pa 20 elementiem vienā rindā).
- Masīva A nosaukums un masīvs A pēc pārveidošanas (pa 20 elementiem vienā rindā).

Risinājuma algoritma analīze:

Lai apmainītu masīva divu elementu vērtības ir nepieciešams trešais mainīgais. Lietojot vienu mainīgo, var apgriezt visu masīvu otrādi:



Mainīgajam C var piešķirt pirmā masīva elementa vērtību, pēc tam pirmajam masīva elementam var piešķirt pēdējā elementa vērtību. Pēc tam pēdējam masīva elementam var piešķirt pirmā elementa vērtību, kas tika saglabāts mainīgajā C.

Tādējādi, lai apmainītu vietām A[0] ar A[39] ir jāizpilda sekojošas darbības:

```
C = A[0];    A[0] = A[39];    A[39] = C;
```

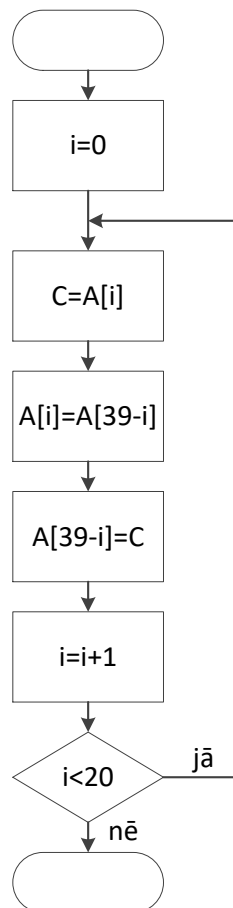
Pēc tam ir jāapmaina vietām elements ar indeksu 1 un elements ar indeksu 38. Nākamā solī elementus ar indeksiem 2 un 37, pēc tam ar indeksu 3 un 36. Tās nozīmē, ka katra cikla iterācijā ir jāapmaina vietām elementi ar indeksiem i un 39-i. No tā secinām, kā vispārējā veidā divu elementu apmaiņu var aprakstīt tā:

```
C = A[i];    A[i] = A[39-i];    A[39-i] = C;
```

Mainīgo apraksts:

Mainīga nosaukums	Tips	Komentārs
A	float []	masīvs
K	int	ievadāmā vērtība
C	float	papildus mainīgais tekošā masīva elementa glabāšanai
i	int	masīva elementa indekss

3. Uzdevuma risināšanas algoritms (masīva apgriešana otrādāk)



4. Izstrādātās programmas pirmkods

```
package dip107;

import java.util.Random;
import java.util.Scanner;

public class Md3_000rdb111 {

    public static void main(String[] args) {

        float A[] = new float[40];
        int K, i;
        float C;
        Scanner sc = new Scanner(System.in);

        System.out.println("000RDB111 Jānis Programmētājs 1");
        System.out.print("K=");
        if (sc.hasNextInt())
            K = sc.nextInt();
        else {
            System.out.println("input-output error");
            sc.close();
            return;
        }
        sc.close();
```

```

System.out.println("result:");

if (K<=0) {
    Random r = new Random();
    for (i=0; i<40; i++)
        A[i] = r.nextFloat()*200-100;
}
else
    for (i=0; i<40; i++)
        A[i] = (i+1)*K/(K+1.0f);

System.out.println("A:");
for (i=0; i<40; i++) {
    System.out.printf("%.1f\t", A[i]);
    if (i==19) System.out.println();
}

for (i=0; i<20; i++) {
    C = A[i];
    A[i] = A[39-i];
    A[39-i] = C;
}

System.out.println("\nA:");
for (i=0; i<40; i++) {
    System.out.printf("%.1f\t", A[i]);
    if (i==19) System.out.println();
}
}
}

```

5. Programmas testēšana

Programmas darbība tika pārbaudīta izmantojot sekojošu kontroles piemēru:

K vērtība	gaidāms rezultāts										
1	A:										
	0.5	1.0	1.5	2.0	2.5	3.0	3.5	4.0	4.5	5.0	5.5
		6.0	6.5	7.0	7.5	8.0	8.5	9.0	9.5	10.0	
	10.5	11.0	11.5	12.0	12.5	13.0	13.5	14.0	14.5	15.0	15.5
		16.0	16.5	17.0	17.5	18.0	18.5	19.0	19.5	20.0	
	A:										
a	20.0	19.5	19.0	18.5	18.0	17.5	17.0	16.5	16.0	15.5	15.0
		14.5	14.0	13.5	13.0	12.5	12.0	11.5	11.0	10.5	
	10.0	9.5	9.0	8.5	8.0	7.5	7.0	6.5	6.0	5.5	5.0
		4.5	4.0	3.5	3.0	2.5	2.0	1.5	1.0	0.5	
	input-output error										
0	masīvs tiek aizpildīts ar patvaļīgām vērtībām										