

## Lucrarea de laborator nr. 6.

**Tema:** Utilizarea funcțiilor create de utilizator la prelucrarea datelor în limbajul C

**Scopul lucrării:** Studierea posibilităților și mijloacelor limbajului C pentru programarea algoritmilor cu structură ramificată și ciclică la prelucrarea tablourilor unidimensionale și bidimensionale, utilizând funcții create de utilizator.

### Programarea algoritmilor cu structură ciclică.

**Varianta 1.** Se dă un număr natural  $n$ . Să se scrie două funcții, apelate ulterior în funcția **main()**, care va calcula: *funcția 1* – valoarea expresiei  $\frac{\pi}{4}$ , pentru valoarea lui  $\pi$  se va folosi constanta respectivă din fișierul antet *math.h*, iar *funcția 2* – valoarea expresiei ce urmează:  $1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \frac{1}{9} - \frac{1}{11} + \dots + \frac{(-1)^n}{2n+1}$ . Să se compare rezultatele expresiilor în funcția **main()** și să se afișeze mesaj corespunzător cercetării.

**Varianta 2.** Se dă un număr natural  $n$ . Să se scrie două funcții, apelate ulterior în funcția **main()**, care va calcula: *funcția 1* – valoarea expresiei  $\frac{\pi}{8}$ , pentru valoarea lui  $\pi$  se va folosi constanta respectivă din fișierul antet *math.h*, iar *funcția 2* – valoarea expresiei ce urmează:  $\frac{1}{1 \times 3} + \frac{1}{5 \times 7} + \dots + \frac{1}{(4n+1)(4n+3)}$ . Să se compare rezultatele expresiilor în funcția **main()** și să se afișeze mesaj corespunzător cercetării.

**Varianta 3.** Se dă un număr natural  $n$ . Să se scrie două funcții, apelate ulterior în funcția **main()**, care va calcula: *funcția 1* – valoarea expresiei  $\frac{\pi}{2}$ , pentru valoarea lui  $\pi$  se va folosi constanta respectivă din fișierul antet *math.h*, iar *funcția 2* – valoarea expresiei ce urmează:  $\frac{2 \times 2}{1 \times 1} \times \frac{4 \times 4}{3 \times 3} \times \frac{6 \times 6}{5 \times 5} \times \dots \times \frac{2n \times 2n}{(2n-1)(2n-1)}$ . Să se compare rezultatele expresiilor în funcția **main()** și să se afișeze mesaj corespunzător cercetării.

**Varianta 4.** Se dă un număr natural  $n$ . Să se scrie două funcții, apelate ulterior în funcția **main()**, care va calcula: *funcția 1* – valoarea expresiei  $\frac{\pi^2}{6}$ , pentru valoarea lui  $\pi$  se va folosi constanta respectivă din fișierul antet *math.h*, iar *funcția 2* – valoarea expresiei ce urmează:  $\frac{2^2}{2^2-1} \times \frac{3^2}{3^2-1} \times \frac{5^2}{5^2-1} \times \frac{7^2}{7^2-1} \times \dots$ . Să se compare rezultatele expresiilor în funcția **main()** și să se afișeze mesaj corespunzător cercetării.

**Varianta 5.** Se dă un număr natural  $n$ . Să se scrie două funcții, apelate ulterior în funcția **main()**, care va calcula: *funcția 1* – va returna valoarea lui  $\pi$  extrasă din fișierul antet *math.h*, iar *funcția 2* – valoarea expresiei ce urmează:  $2 \left[ 1 + \frac{1}{3} + \frac{1 \times 2}{3 \times 5} + \frac{1 \times 2 \times 3}{3 \times 5 \times 7} + \dots \right]$ . Să se compare rezultatele expresiilor în funcția **main()** și să se afișeze mesaj corespunzător cercetării.

**Varianta 6.** Se dă un număr natural  $n$ . Să se scrie două funcții, apelate ulterior în funcția **main()**, care va calcula: *funcția 1* – valoarea expresiei  $\frac{\pi^3}{32}$ , pentru valoarea lui  $\pi$  se va folosi constanta respectivă din fișierul antet *math.h*, iar *funcția 2* – valoarea expresiei ce urmează:  $\frac{1}{1^3} - \frac{1}{3^3} + \frac{1}{5^3} - \frac{1}{7^3} + \dots$ . Să se compare rezultatele expresiilor în funcția **main()** și să se afișeze mesaj corespunzător cercetării.

**Varianta 7.** Se dă un număr natural  $n$ . Să se scrie două funcții, apelate ulterior în funcția **main()**, care va calcula: *funcția 1* – valoarea expresiei  $\frac{\pi^2}{6}$ , pentru valoarea lui  $\pi$  se va folosi constanta respectivă din fișierul antet *math.h*, iar *funcția 2* – valoarea expresiei ce urmează:  $\frac{1}{1^2} + \frac{1}{2^2} + \frac{1}{3^2} + \frac{1}{4^2} + \dots + \frac{1}{n^2}$ . Să se compare rezultatele expresiilor în funcția **main()** și să se afișeze mesaj corespunzător cercetării.

**Varianta 8.** Se dă un număr natural  $n$ . Să se scrie două funcții, apelate ulterior în funcția **main()**, care va calcula: *funcția 1* – valoarea expresiei  $\frac{\pi^2}{8}$ , pentru valoarea lui  $\pi$  se va folosi constanta respectivă din fișierul antet *math.h*, iar *funcția 2* – valoarea expresiei ce urmează:  $\frac{1}{1^2} + \frac{1}{3^2} + \frac{1}{5^2} + \frac{1}{7^2} + \dots$ . Să se compare rezultatele expresiilor în funcția **main()** și să se afișeze mesaj corespunzător cercetării.

**Varianta 9.** Se dă un număr natural  $n$ . Să se scrie două funcții, apelate ulterior în funcția **main()**, care va calcula: *funcția 1* – valoarea expresiei  $\frac{\pi}{4}$ , pentru valoarea lui  $\pi$  se va folosi constanta respectivă din fișierul

antet *math.h*, iar funcția 2 – valoarea expresiei ce urmează: 
$$\frac{1}{1 + \frac{1^2}{2 + \frac{3^2}{2 + \frac{5^2}{2 + \frac{7^2}{2 + \dots}}}}}$$
. Să se compare rezultatele expresiilor

în funcția **main()** și să se afișeze mesaj corespunzător cercetării.

**Varianta 10.** Se dă un număr natural  $n$ . Să se scrie două funcții, apelate ulterior în funcția **main()**, care va calcula: funcția 1 – va returna valoarea lui  $\pi$  extrasă din fișierul antet *math.h*, iar funcția 2 – valoarea expresiei ce urmează:  $2 \times \frac{2}{\sqrt{2}} \times \frac{2}{\sqrt{2+\sqrt{2}}} \times \frac{2}{\sqrt{2+\sqrt{2+\sqrt{2}}}} \times \dots \times \frac{2}{\sqrt{2+\sqrt{2+\sqrt{2+\sqrt{2+\dots}}}}}$ . Să se compare rezultatele expresiilor în

funcția **main()** și să se afișeze mesaj corespunzător cercetării.

**Varianta 11.** Se dă un număr natural  $n$ . Să se scrie două funcții, apelate ulterior în funcția **main()**, care va calcula: funcția 1 – valoarea expresiei  $\frac{4}{\pi}$ , pentru valoarea lui  $\pi$  se va folosi constanta respectivă din fișierul antet *math.h*, iar funcția 2 – valoarea expresiei ce urmează:  $1 + \frac{1^2}{2 + \frac{3^2}{2 + \frac{5^2}{2 + \frac{7^2}{2 + \dots}}}}$ . Să se compare rezultatele

expresiilor în funcția **main()** și să se afișeze mesaj corespunzător cercetării.

**Varianta 12.** Se dă un număr natural  $n$ . Să se scrie două funcții, apelate ulterior în funcția **main()**, care va calcula: funcția 1 – va returna valoarea lui  $\pi$  extrasă din fișierul antet *math.h*, iar funcția 2 – valoarea expresiei ce urmează:  $\frac{4}{1 + \frac{1^2}{2 + \frac{3^2}{2 + \frac{5^2}{2 + \frac{7^2}{2 + \dots}}}}}$ . Să se compare rezultatele expresiilor în funcția **main()** și să se afișeze mesaj

corespunzător cercetării.

**Varianta 13.** Se dă un număr natural  $n$ . Să se scrie două funcții, apelate ulterior în funcția **main()**, care va calcula: funcția 1 – va returna valoarea expresiei  $\sqrt{2}$ , iar funcția 2 – valoarea expresiei ce urmează:  $1 + \frac{1}{2 + \frac{1}{2 + \frac{1}{2 + \dots}}}$ . Să se compare rezultatele expresiilor în funcția **main()** și să se afișeze mesaj corespunzător cercetării.

**Varianta 14.** Se dă un număr natural  $n$  și un număr întreg  $a$ . Să se scrie două funcții, apelate ulterior în funcția **main()**, care va calcula: funcția 1 – va returna valoarea expresiei  ${}^{n-1}\sqrt{a}$ , iar funcția 2 – valoarea expresiei ce urmează:  $\sqrt[n]{a \times \sqrt[n]{a \times \sqrt[n]{a \times \dots}}}$ . Să se compare rezultatele expresiilor în funcția **main()** și să se afișeze mesaj corespunzător cercetării.

**Varianta 15.** Se dă un număr natural  $n$  și un număr întreg  $a$ . Să se scrie două funcții, apelate ulterior în funcția **main()**, care va calcula: funcția 1 – va returna valoarea expresiei  ${}^{n+1}\sqrt{a}$ , iar funcția 2 – valoarea expresiei ce urmează:  $\sqrt[n]{a \div \sqrt[n]{a \div \sqrt[n]{a \div \dots}}}$ . Să se compare rezultatele expresiilor în funcția **main()** și să se afișeze mesaj corespunzător cercetării.

**Varianta 16.** Se dă un număr natural  $n$  și un număr întreg  $a$ . Să se scrie două funcții, apelate ulterior în funcția **main()**, care va calcula: funcția 1 – va returna valoarea expresiei  ${}^{m \times n - 1}\sqrt{a^{m+1}}$ , iar funcția 2 – valoarea expresiei ce urmează:  $\sqrt[n]{a \times \sqrt[m]{a \times \sqrt[n]{a \times \sqrt[m]{a \times \dots}}}}$ . Să se compare rezultatele expresiilor în funcția **main()**

și să se afișeze mesaj corespunzător cercetării.

**Varianta 17.** Se dă un număr natural  $n$  și două numere întregi  $a$  și  $b$ . Să se scrie două funcții, apelate ulterior în funcția **main()**, care va calcula: funcția 1 – va returna valoarea expresiei  ${}^{m \times n - 1}\sqrt{a^m \times b}$ , iar funcția 2 –

valoarea expresiei ce urmează:  $\sqrt[n]{a \times \sqrt[m]{b \times \sqrt[n]{a \times \sqrt[m]{b \times \dots}}}}$ . Să se compare rezultatele expresiilor în funcția **main()** și să se afișeze mesaj corespunzător cercetării.

**Varianta 18.** Se dă un număr natural  $n$  și un număr întreg  $a$ . Să se scrie două funcții, apelate ulterior în funcția **main()**, care va calcula: *funcția 1* – va returna valoarea expresiei  $\frac{1+\sqrt{4 \times a+1}}{2}$ , iar *funcția 2* – valoarea expresiei

ce urmează:  $\sqrt{a + \sqrt{a + \sqrt{a + \dots}}}$ . Să se compare rezultatele expresiilor în funcția **main()** și să se afișeze mesaj corespunzător cercetării.

**Varianta 19.** Se dă un număr natural  $n$  și un număr întreg  $a$ . Să se scrie două funcții, apelate ulterior în funcția **main()**, care va calcula: *funcția 1* – va returna valoarea expresiei  $\frac{-1+\sqrt{4 \times a+1}}{2}$ , iar *funcția 2* – valoarea expresiei

ce urmează:  $\sqrt{a - \sqrt{a - \sqrt{a - \dots}}}$ . Să se compare rezultatele expresiilor în funcția **main()** și să se afișeze mesaj corespunzător cercetării.

**Varianta 20.** Se dă un număr natural  $n$  și un număr întreg  $a$ . Să se scrie două funcții, apelate ulterior în funcția **main()**, care va calcula: *funcția 1* – va returna valoarea expresiei  $\left(\frac{a+\sqrt{a^2-4}}{2}\right)^{\frac{1}{2^n}} + \left(\frac{a+\sqrt{a^2-4}}{2}\right)^{\frac{-1}{2^n}}$ , iar *funcția 2* –

valoarea expresiei ce urmează:  $\sqrt{2 + \sqrt{2 + \dots + \sqrt{2 + a}}}$ . Să se compare rezultatele expresiilor în funcția **main()** și să se afișeze mesaj corespunzător cercetării.

### Programarea algoritmilor cu structură ramificată și ciclică la prelucrarea tablourilor unidimensionale.

**Varianta 1.** Se dă un tablou unidimensional cu  $n$  elemente de tip întreg, generate aleatoriu. Să se scrie funcții în limbajul C, care vor calcula: *funcția 1* – suma cifrelor unui număr, iar *funcția 2* – va determina elementul cu suma cifrelor cea mai mare. În consolă se va afișa elementele vectorului într-o singură linie despărțite de un spațiu și elementul cu cea mai mare sumă a propriilor cifre, inclusiv, suma obținută.

**Varianta 2.** Se dă un tablou unidimensional cu  $n$  elemente de tip întreg, generate aleatoriu. Să se scrie funcții în limbajul C, care vor calcula: *funcția 1* – numărul cifrelor unui număr, iar *funcția 2* – va determina elementul cu cel mai mare număr de cifre. În consolă se va afișa elementele vectorului într-o singură linie despărțite de un spațiu și elementul cu cel mai mare număr de cifre, inclusiv, numărul propriu zis (câte cifre sunt).

**Varianta 3.** Se dă un tablou unidimensional cu  $n$  elemente de tip întreg, generate aleatoriu. Să se scrie funcții în limbajul C, care vor determina: *funcția 1* – elementul vectorului dacă este număr prim, iar *funcția 2* – va determina cel mai mare număr prim din vector. În consolă se va afișa elementele vectorului într-o singură linie despărțite de un spațiu și cel mai mare număr prim din vector.

**Varianta 4.** Se dă un tablou unidimensional cu  $n$  elemente de tip întreg, generate aleatoriu. Să se scrie funcții în limbajul C, care vor determina: *funcția 1* – numărul de divizori al unui element, iar *funcția 2* – va determina elementul cu cel mai mare număr de divizori. În consolă se va afișa elementele vectorului într-o singură linie despărțite de un spațiu și elementul cu cel mai mare număr de divizori din vector, inclusiv, numărul de divizori.

**Varianta 5.** Se dă un tablou unidimensional cu  $n$  elemente de tip întreg, generate aleatoriu. Să se scrie funcții în limbajul C, care vor calcula: *funcția 1* – numărul cifrelor unui număr, iar *funcția 2* – va determina câte elemente din vector au un număr impar de cifre. În consolă se va afișa elementele vectorului într-o singură linie despărțite de un spațiu, pe o linie nouă, elementele cu numărul impar de cifre și, respectiv, numărul de elemente cu un număr impar de cifre.

**Varianta 6.** Se dă un tablou unidimensional cu  $n$  elemente de tip întreg, generate aleatoriu. Să se scrie funcții în limbajul C, care vor calcula: *funcția 1* – numărul cifrelor unui număr, iar *funcția 2* – va determina câte elemente din vector au un număr impar de cifre. În consolă se va afișa elementele vectorului într-o singură linie despărțite de un spațiu, pe o linie nouă, elementele cu numărul impar de cifre și, respectiv, numărul de elemente cu un număr impar de cifre.

**Varianta 7.** Se dă un tablou unidimensional cu  $n$  elemente de tip întreg, generate aleatoriu. Să se scrie funcții în limbajul C, care vor determina: *funcția 1* – dacă cifrele unui element sunt distincte, iar *funcția 2* – va calcula câte elemente din vector au toate cifrele elementului distincte. În consolă se va afișa elementele vectorului într-o singură linie despărțite de un spațiu și elementul cu cifrele distincte și, inclusiv, numărul elementelor ce sunt formate din cifre distincte.

### Programarea algoritmilor cu structură ramificată și ciclică la prelucrarea tablourilor bidimensionale.

**Varianta 1.** Se dă un tablou bidimensional cu  $n \times m$  numere întregi, citite de la tastatură. Să se scrie funcții în limbajul C, care vor calcula: *funcția 1* – numărul de elemente numere prime pe fiecare linie, iar *funcția 2* – va calcula dacă numărul este prim. În consolă se va afișa elementele matricei despărțite de un spațiu în format  $n \times m$ , numărul liniei cu cele mai multe numere prime și, respectiv, numerele prime din acea linie.

**Varianta 2.** Se dă un tablou bidimensional cu  $n \times m$  numere întregi, citite de la tastatură. Să se scrie funcții în limbajul C, care vor calcula: *funcția 1* – numărul de elemente numere prime pe fiecare coloană, iar *funcția 2* – va calcula dacă numărul este prim. În consolă se va afișa elementele matricei despărțite de un spațiu în format  $n \times m$ , numărul coloanei cu cele mai multe numere prime și, respectiv, numerele prime din acea coloană.

**Varianta 3.** Se dă un tablou bidimensional cu  $n \times m$  numere întregi, citite de la tastatură. Să se scrie funcții în limbajul C, care vor calcula: *funcția 1* – numărul de divizori pentru un element, iar *funcția 2* – va determina linia cu cei mai mulți divizori, se va calcula sumarul tuturor divizorilor elementelor dintr-o linie. În consolă se va afișa elementele matricei despărțite de un spațiu în format  $n \times m$ , numărul liniei cu cei mai mulți divizori.

**Varianta 4.** Se dă un tablou bidimensional cu  $n \times m$  numere întregi, citite de la tastatură. Să se scrie funcții în limbajul C, care vor calcula: *funcția 1* – numărul de divizori pentru un element, iar *funcția 2* – va determina coloana cu cei mai mulți divizori, se va calcula sumarul tuturor divizorilor elementelor dintr-o coloană. În consolă se va afișa elementele matricei despărțite de un spațiu în format  $n \times m$ , numărul coloanei cu cei mai mulți divizori.

**Varianta 5.** Se dă un tablou bidimensional cu  $n \times m$  numere întregi, citite de la tastatură. Să se scrie funcții în limbajul C, care vor calcula: *funcția 1* – numărul de elemente numere prime pe fiecare linie, iar *funcția 2* – va calcula dacă numărul este prim. În consolă se va afișa elementele matricei despărțite de un spațiu în format  $n \times m$ , numărul liniei cu cele mai puține numere prime și, respectiv, numerele prime din acea linie.

**Varianta 6.** Se dă un tablou bidimensional cu  $n \times m$  numere întregi, citite de la tastatură. Să se scrie funcții în limbajul C, care vor calcula: *funcția 1* – numărul de elemente numere prime pe fiecare coloană, iar *funcția 2* – va calcula dacă numărul este prim. În consolă se va afișa elementele matricei despărțite de un spațiu în format  $n \times m$ , numărul coloanei cu cele mai puține numere prime și, respectiv, numerele prime din acea coloană.

**Varianta 7.** Se dă un tablou bidimensional cu  $n \times m$  numere întregi, citite de la tastatură. Să se scrie funcții în limbajul C, care vor calcula: *funcția 1* – numărul de divizori pentru un element, iar *funcția 2* – va determina linia cu cei mai puțini divizori, se va calcula sumarul tuturor divizorilor elementelor dintr-o linie. În consolă se va afișa elementele matricei despărțite de un spațiu în format  $n \times m$ , numărul liniei cu cei mai puțini divizori.

**Varianta 8.** Se dă un tablou bidimensional cu  $n \times m$  numere întregi, citite de la tastatură. Să se scrie funcții în limbajul C, care vor calcula: *funcția 1* – numărul de divizori pentru un element, iar *funcția 2* – va determina coloana cu cei mai puțini divizori, se va calcula sumarul tuturor divizorilor elementelor dintr-o coloană. În consolă se va afișa elementele matricei despărțite de un spațiu în format  $n \times m$ , numărul coloanei cu cei mai puțini divizori.

**Varianta 9.** Se dă un tablou bidimensional cu  $n \times n$  numere reale, citite de la tastatură, unde  $n$  este un număr impar de orice dimensiune. Să se scrie funcții în limbajul C, care vor calcula: *funcția 1* – suma elementelor din domeniul colorat, vezi figura de mai jos, punctul  $a$ , iar *funcția 2* – va calcula numărul de elemente din domeniul colorat. În consolă se va afișa elementele matricei despărțite de un spațiu în format  $n \times n$ , suma și numărul elementelor din domeniul colorat.

**Varianta 10.** Se dă un tablou bidimensional cu  $n \times n$  numere reale, citite de la tastatură, unde  $n$  este un număr impar de orice dimensiune. Să se scrie funcții în limbajul C, care vor calcula: *funcția 1* – produsul elementelor

din domeniul colorat, vezi figura de mai jos, punctul *b*, iar *funcția 2* – va calcula numărul de elemente din domeniul colorat. În consolă se va afișa elementele matricei despărțite de un spațiu în format  $n \times n$ , produsul și numărul elementelor din domeniul colorat.

**Varianta 11.** Se dă un tablou bidimensional cu  $n \times n$  numere reale, citite de la tastatură, unde  $n$  este un număr impar de orice dimensiune. Să se scrie funcții în limbajul C, care vor calcula: *funcția 1* – media aritmetică elementelor din domeniul colorat, vezi figura de mai jos, punctul *c*, iar *funcția 2* – va calcula numărul de elemente din domeniul colorat. În consolă se va afișa elementele matricei despărțite de un spațiu în format  $n \times n$ , media aritmetică și numărul elementelor din domeniul colorat.

**Varianta 12.** Se dă un tablou bidimensional cu  $n \times n$  numere reale, citite de la tastatură, unde  $n$  este un număr impar de orice dimensiune. Să se scrie funcții în limbajul C, care vor calcula: *funcția 1* – suma elementelor din domeniul colorat, vezi figura de mai jos, punctul *d*, iar *funcția 2* – va calcula numărul de elemente din domeniul colorat. În consolă se va afișa elementele matricei despărțite de un spațiu în format  $n \times n$ , suma și numărul elementelor din domeniul colorat.

**Varianta 13.** Se dă un tablou bidimensional cu  $n \times n$  numere reale, citite de la tastatură, unde  $n$  este un număr impar de orice dimensiune. Să se scrie funcții în limbajul C, care vor calcula: *funcția 1* – produsul elementelor din domeniul colorat, vezi figura de mai jos, punctul *e*, iar *funcția 2* – va calcula numărul de elemente din domeniul colorat. În consolă se va afișa elementele matricei despărțite de un spațiu în format  $n \times n$ , produsul și numărul elementelor din domeniul colorat.

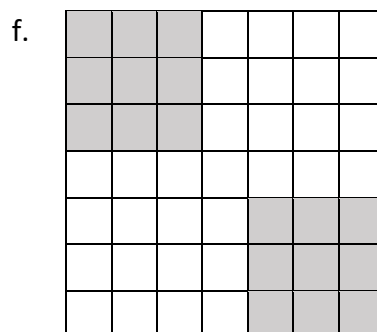
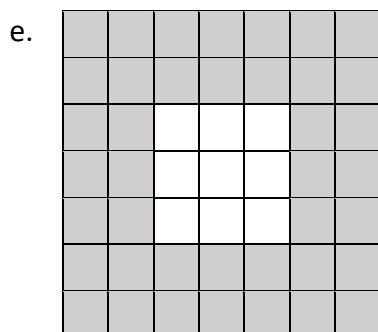
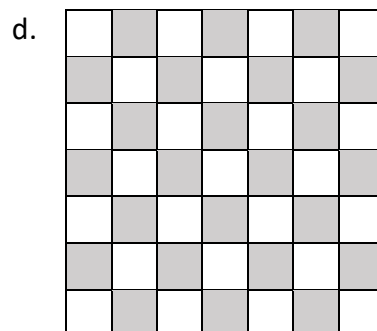
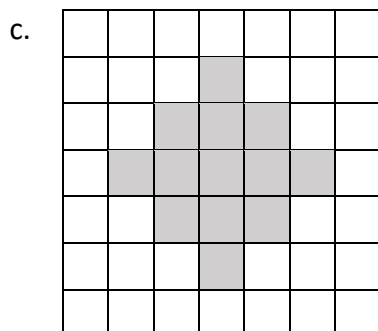
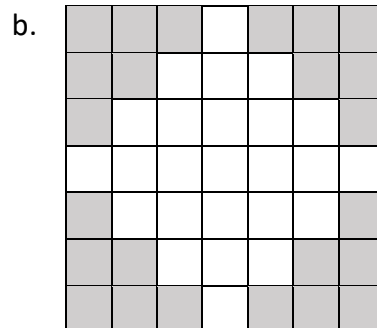
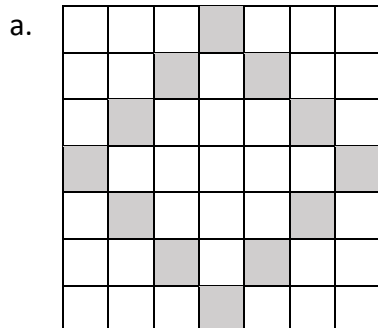
**Varianta 14.** Se dă un tablou bidimensional cu  $n \times n$  numere reale, citite de la tastatură, unde  $n$  este un număr impar de orice dimensiune. Să se scrie funcții în limbajul C, care vor calcula: *funcția 1* – media aritmetică elementelor din domeniul colorat, vezi figura de mai jos, punctul *f*, iar *funcția 2* – va calcula numărul de elemente din domeniul colorat. În consolă se va afișa elementele matricei despărțite de un spațiu în format  $n \times n$ , media aritmetică și numărul elementelor din domeniul colorat.

**Varianta 15.** Se dă un tablou bidimensional cu  $n \times n$  numere reale, citite de la tastatură, unde  $n$  este un număr impar de orice dimensiune. Să se scrie funcții în limbajul C, care vor calcula: *funcția 1* – suma elementelor din domeniul colorat, vezi figura de mai jos, punctul *g*, iar *funcția 2* – va calcula numărul de elemente din domeniul colorat. În consolă se va afișa elementele matricei despărțite de un spațiu în format  $n \times n$ , suma și numărul elementelor din domeniul colorat.

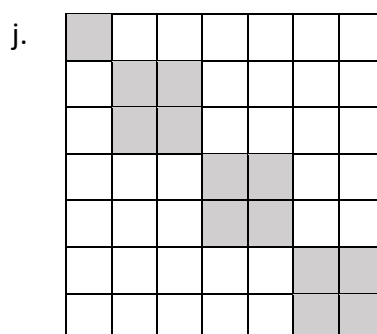
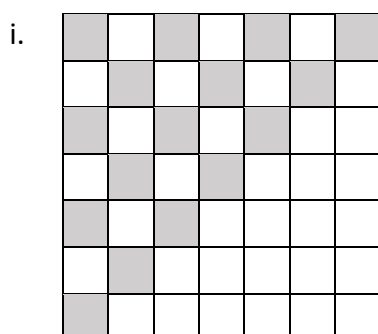
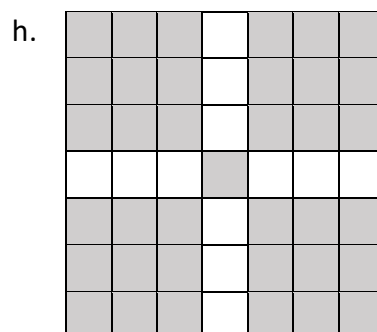
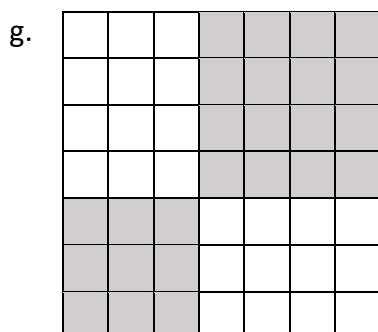
**Varianta 16.** Se dă un tablou bidimensional cu  $n \times n$  numere reale, citite de la tastatură, unde  $n$  este un număr impar de orice dimensiune. Să se scrie funcții în limbajul C, care vor calcula: *funcția 1* – produsul elementelor din domeniul colorat, vezi figura de mai jos, punctul *h*, iar *funcția 2* – va calcula numărul de elemente din domeniul colorat. În consolă se va afișa elementele matricei despărțite de un spațiu în format  $n \times n$ , produsul și numărul elementelor din domeniul colorat.

**Varianta 17.** Se dă un tablou bidimensional cu  $n \times n$  numere reale, citite de la tastatură, unde  $n$  este un număr impar de orice dimensiune. Să se scrie funcții în limbajul C, care vor calcula: *funcția 1* – suma elementelor din domeniul colorat, vezi figura de mai jos, punctul *i*, iar *funcția 2* – va calcula numărul de elemente din domeniul colorat. În consolă se va afișa elementele matricei despărțite de un spațiu în format  $n \times n$ , suma și numărul elementelor din domeniul colorat.

**Varianta 18.** Se dă un tablou bidimensional cu  $n \times n$  numere reale, citite de la tastatură, unde  $n$  este un număr impar de orice dimensiune. Să se scrie funcții în limbajul C, care vor calcula: *funcția 1* – produsul elementelor din domeniul colorat, vezi figura de mai jos, punctul *j*, iar *funcția 2* – va calcula numărul de elemente din domeniul colorat. În consolă se va afișa elementele matricei despărțite de un spațiu în format  $n \times n$ , produsul și numărul elementelor din domeniul colorat.



Elementele chenarului a 2 rânduri, indiferent de valoarea lui  $n$ .



**Sarcină (conform variantelor) + schema bloc pentru o singură sarcină, la alegere:**

Student 1:    Structură ciclică: 1, 9; Tablourilor unidimensionale: 7; Tablourilor bidimensionale: 4, 12.	Student 13:   Structură ciclică: 5, 9; Tablourilor unidimensionale: 2; Tablourilor bidimensionale: 8, 14.
Student 2:    Structură ciclică: 2, 10; Tablourilor unidimensionale: 6; Tablourilor bidimensionale: 5, 13.	Student 14:   Structură ciclică: 6, 10; Tablourilor unidimensionale: 1; Tablourilor bidimensionale: 2, 15.
Student 3:    Structură ciclică: 3, 11; Tablourilor unidimensionale: 5; Tablourilor bidimensionale: 6, 14.	Student 15:   Structură ciclică: 7, 11; Tablourilor unidimensionale: 7; Tablourilor bidimensionale: 1, 16.
Student 4:    Structură ciclică: 4, 12; Tablourilor unidimensionale: 4; Tablourilor bidimensionale: 7, 15.	Student 16:   Structură ciclică: 8, 12; Tablourilor unidimensionale: 6; Tablourilor bidimensionale: 3, 17.
Student 5:    Structură ciclică: 5, 13; Tablourilor unidimensionale: 3; Tablourilor bidimensionale: 8, 16.	Student 17:   Structură ciclică: 1, 13; Tablourilor unidimensionale: 3; Tablourilor bidimensionale: 4, 18.
Student 6:    Structură ciclică: 6, 14; Tablourilor unidimensionale: 2; Tablourilor bidimensionale: 1, 17.	Student 18:   Structură ciclică: 2, 14; Tablourilor unidimensionale: 4; Tablourilor bidimensionale: 5, 9.
Student 7:    Structură ciclică: 7, 15; Tablourilor unidimensionale: 1; Tablourilor bidimensionale: 2, 18.	Student 19:   Structură ciclică: 3, 15; Tablourilor unidimensionale: 5; Tablourilor bidimensionale: 6, 10.
Student 8:    Structură ciclică: 8, 16; Tablourilor unidimensionale: 7; Tablourilor bidimensionale: 3, 9.	Student 20:   Structură ciclică: 4, 16; Tablourilor unidimensionale: 2; Tablourilor bidimensionale: 7, 11.
Student 9:    Structură ciclică: 1, 17; Tablourilor unidimensionale: 6; Tablourilor bidimensionale: 4, 10.	Student 21:   Structură ciclică: 5, 17; Tablourilor unidimensionale: 1; Tablourilor bidimensionale: 8, 12.
Student 10:   Structură ciclică: 2, 18; Tablourilor unidimensionale: 5; Tablourilor bidimensionale: 6, 11.	Student 22:   Structură ciclică: 6, 18; Tablourilor unidimensionale: 7; Tablourilor bidimensionale: 1, 13.
Student 11:   Structură ciclică: 3, 19; Tablourilor unidimensionale: 4; Tablourilor bidimensionale: 5, 12.	Student 23:   Structură ciclică: 7, 19; Tablourilor unidimensionale: 6; Tablourilor bidimensionale: 2, 14.
Student 12:   Structură ciclică: 4, 20; Tablourilor unidimensionale: 3; Tablourilor bidimensionale: 7, 13.	Student 24:   Structură ciclică: 8, 20; Tablourilor unidimensionale: 3; Tablourilor bidimensionale: 4, 15.