

ATESTAT 2016  
– ALGORITMI –

1. Se citește un șir de  $n$  numere naturale de maxim 9 cifre fiecare, să se afișeze pentru fiecare număr din șir numărul cifrelor pare și numărul cifrelor impare.
2. Să se citească un număr natural  $x$ . Afișați **cifrele numărului în ordine crescătoare**.
3. Să se citească un număr natural de maxim 4 cifre. Afișați un mesaj corespunzător dacă numărul citit este scris în într-o bază de numerație dată,  $b$ , ( $b < 10$ ).
4. Se citește un număr natural  $n$ . Introduceți semnul "\*" (înmulțire) în număr astfel încât **produsul** obținut să fie **maxim**.

Exemplu:  $n=4322$ , produsul maxim se obține astfel:

$$4*322=1288$$

(celelalte variante:  $43*22=946$  sau  $432*2=864$  dau produs mai mic)

5. Se citește un șir de  $n$  numere naturale de maxim 9 cifre fiecare, să se afișeze pentru fiecare număr din șir cifra maximă.
6. Se citește un șir de  $n$  numere naturale de maxim 9 cifre fiecare, să se afișeze numărul cu suma cifrelor maximă/minimă.
7. Se citește un șir de  $n$  numere naturale, să se afișeze pentru fiecare număr din șir cifra de vaoare maximă și cifra de valoare minimă.
8. Se citește un număr natural "n" de maxim 9 cifre. Să se determine dacă el are toate **cifrele ordonate strict descrescător** de la cifra cea mai semnificativă spre cifra unităților. Exemplu: Pentru 54321 se va afișa "DA", iar pentru 543234 se va afișa "NU".
9. Se citește un număr natural cu cel mult 9 cifre. Să se determine de câte ori apare **ultima cifră** în numărul citit.
10. Se citește un număr natural "n" cu cel mult 9 cifre. Să se determine **cea mai mare cifră** a lui "n" și de câte ori apare ea în "n".
11. Se citește un număr natural "n" de maxim 9 cifre. Să se elimine toate cifrele impare din număr.
12. Se citesc numere naturale până la întâlnirea valorii 0. Să se realizeze un program care afișează pentru fiecare număr citit **numărul de apariții** al unei cifre specificate de la tastatură.
13. Se citește un număr natural "n" de maxim 9 cifre. Să se verifice și să se afișeze un mesaj corespunzător dacă este **palindrom** (citit de la dreapta spre stânga are aceeași valoare). Exemplu: 123321 este palindrom, iar 12322 nu este.
14. Se citește un șir de  $n$  numere naturale de maxim 9 cifre fiecare, să se afișeze pentru fiecare număr din șir inversul/oglinditul acestuia.
15. Se citește un număr natural "n" de maxim 9 cifre. Afișați cele 2 numere obținute prin **împărțirea "la mijloc"** a numărului "n".

Exemple:  $n=12345$  se afișează 12 și 345

$n=12345678$  se afișează 1234 5678

16. Se citește un număr natural "n" de maxim 9 cifre. Afișați **cifrele distincte** ale lui n (în orice ordine).  
Exemplu: Cifrele distincte ale lui 234542 sunt 2, 3, 4 și 5.
17. Fie  $n$  un număr natural nenul par ( $n < 100$ ) și un șir cu  $n$  numere reale distincte. Să se ordoneze crescător elementele din prima jumătate a vectorului și descrescător celelalte.  
Exemplu: pentru  $n=6$  și șirul 5, 8, 2, 1, 7, 9 se va obține 2, 5, 8, 9, 7, 1
18. Să se determine toate **numerele întregi de 3 cifre** de forma abc cu proprietatea că numărul este egal cu suma cuburilor cifrelor.
19. Să se creeze un fișier text care conține toate numerele care aparțin intervalului determinat de a și b care sunt formate din cifre identice. Valorile lui a și b se citesc de la tastatură, numere naturale de maxim 9 cifre.  
Exemplu : a=10, b=150 fișierul de ieșire va conține numerele: 11, 22, 33, ..., 99, 111 .
20. Se citește de la tastatură un număr natural n ( $100 \leq n \leq 999$ ). Dintre numerele naturale mai mici sau egale cu n, să se afișeze acelea care sunt **divizibile cu suma cifrelor** lor.
21. Să se afișeze primele n numere care au **suma cifrelor**  $\leq m$ , n număr natural nenul de maxim 3 cifre, m număr natural nenul de maxim 2 cifre.  
Exemplu: n=10, m=4 : 1, 2, 3, 4, 10, 11, 12, 13, 20, 21.
22. Să citească un număr natural nenul n și să afișeze toate numerele naturale perfecte mai mici sau egale cu n.
23. Sa se citească n numere naturale și să se afișeze numărul cu cei mai mulți divizori, dacă sunt mai multe numere cu același număr maxim de divizori se va afișa unul dintre ele.
24. Să se scrie un program care citește 2 numere naturale și să se afișeze **divizorii comuni** ai celor două numere.
25. Sa se citească un număr natural n. Să se afișeze mesajul: "**Numărul este aproape prim**" dacă n se descompune într-un produs de 2 numere prime și "Nu este prim", altfel.
26. Scrieți un program care să calculeze și să afișeze **cel mai mic multiplu comun** a trei numere citite de la tastatură: a, b, c.
27. Se citește de la tastatură un număr natural n și apoi se citesc de la tastatură n numere naturale. Scrieți un program care să calculeze și să afișeze **cel mai mare divizor comun** al celor n numere naturale.
28. Se dau două numere întregi a și b, ce reprezintă numărătorul, respectiv numitorul unei fracții. Să se realizeze un program care afișează **fracția ireductibilă** obținută.
29. Se citesc perechi de numere până la întâlnirea unei perechi în care cel puțin un număr din pereche este zero. Să se afișeze doar acele **perechi alcătuite din numere prime între ele**. Să se precizeze câte astfel de perechi sunt în total.  
Exemplu: pentru perechile (3 27), (5 12), (8 13), (2 46), (13 17), (0 0)  
Se vor afișa valorile/perechile: (5 12), (8 13), (13 17) și 3 perechi
30. Să se citească un număr natural nenul de maxim 9 cifre și să se testeze dacă numărul și oglinditul său sunt numere prime sau nu.
31. Să se scrie un program care determină și afișează toate **numerele prime** mai mici sau egale cu o valoare n naturală, citită de la tastatură.
32. Scrieți un program care afișează toate **numerele prime** de trei cifre.

33. Se dă un număr natural par  $n$ ,  $n > 4$ . Să se afișeze toate scrierile acestuia în sumă de **două numere prime**.
34. Să se citească numere naturale până la citirea numărului 0, care nu face parte din șir. Să se construiască un vector care să conțină doar numerele **prime**.
35. Fișierul "sir. in" conține pe o singură linie, separate prin caracterul blank/spațiu un șir de numere naturale. Să se scrie în fișierul "sir. out" **elementele prime** din acest șir.
36. Să se determine câte **elemente prime** se află deasupra diagonalei principale a unei matrice cu  $n \times n$  elemente naturale,  $n < 15$ .
37. Să se determine câte **perfecte** (un număr este perfect dacă este egal cu suma divizorilor săi mai mici strict ca numărul) se află sub diagonala principală a unei matrice cu  $n \times n$  elemente naturale,  $n < 15$ .
38. Să se determine câte elemente **cu suma cifrelor număr par** se află deasupra diagonalei secundare a unei matrice cu  $n \times n$  elemente naturale,  $n < 15$ .
39. Să se determine câte **formate doar din cifre impare** se află sub diagonala secundară a unei matrice cu  $n \times n$  elemente naturale,  $n < 15$ .
40. Se citește de la tastatură o matrice  $n \times m$  numere naturale nenule, să se calculeze pentru fiecare coloană numărul de **numere neprime**.  
Exemplu:  
 $n=3$   $m=4$   
 2 5 7 5  
 1 4 7 0  
 2 4 5 6  
 se va afișa      1 2 0 2
41. Se citește de la tastatură numărul natural  $n$  de maxim 9 cifre. Generați în fișierul bac.out **numerele prime mai mici sau egale cu  $n$** , câte 10 pe fiecare linie.  
Exemplu:  
 $N=40$   
 Bac.out  
 2 3 5 7 11 13 17 19 23 29  
 31 37
42. Generați toate numerele naturale de 5 cifre care au proprietatea că **suma cifrelor este divizor al produsului cifrelor**.  
Exemplu: Un astfel de număr este 12345, care are suma cifrelor 15 și produsul 120.
43. Pentru un număr natural  $n$  citit de la tastatură (de cel mult 8 cifre) afișați al  $p$ -lea număr **prim** strict mai mare decât numărul citit.  
Exemplu:  $n=20$ ,  $p=4$  se va afișa 37
44. Să se scrie un program care să afișeze **factorul prim** care apare la puterea cea mai mare în descompunerea numărului natural  $x$  (citat) în factori primi.
45. Să se scrie un program care să afișeze **descompunerea** unui număr natural **în factori primi**.
46. Se citesc din fișierul NUMERE.TXT un număr natural  $n$  (de pe primul rând) și  $n$  numere naturale separate prin spații (de pe al doilea rând). Să se afișeze pe ecran **cel mai mare divizor comun al celor  $n$  numere**.

47. Se citesc de la tastatură  $n$  numere întregi. Să se scrie un program care să **partiționeze acest șir în două subșiruri** după un  $x$  dat astfel: primul subșir să conțină elementele mai mici decât  $x$ , iar al doilea – elementele mai mari decât  $x$ .
48. Se citesc de la tastatură  $n$  numere naturale. Să se scrie un program care să **rearanjeze** aceste numere astfel încât elementele să fie în ordine descrescătoare până la jumătatea șirului, apoi să fie în ordine crescătoare.
49. Se citește de la tastatură un  $n$  natural nenul. Să se scrie un program care să **genereze o matrice** cu  $n$  linii și  $n$  coloane care să fie completată pe linii cu primele  $n*n$  numere pare.
50. Se citește de la tastatură un  $n$  natural nenul. Să se scrie un program care să **genereze o matrice** cu  $n$  linii și  $n$  coloane care să fie completată pe linii cu primele  $n*n$  numere impare, în ordine descrescătoare.
51. Să se scrie un program care să permită citirea de la tastatură a  $n$  valori întregi și să afișeze aceste valori **în ordine crescătoare**. Să se afișeze un mesaj corespunzător dacă cele  $n$  numere citite sunt distincte.
52. Să se verifice dacă o valoare întreagă  $x$ , citită de la tastatură, **se găsește printre cele  $n$  elemente** întregi ale unui vector. Elementele vectorului se vor citi de la tastatură în ordine crescătoare.
53. Fie un vector cu  $n$  componente întregi distincte. Să se determine **elementul maxim din vector** și să se afișeze vectorul după ce elementele din stânga celui maxim au fost sortate crescător, iar cele din dreapta maximumului au fost sortate descrescător.
54. Să citească un vector cu  $n$  elemente naturale, să se calculeze media aritmetică a elementelor pozitive din șir și să se afișeze elementele pare de pe poziții impare.
55. Să se ordoneze crescător un vector de  $n$  elemente numere reale folosind **metoda bubble sort**. Să se afișeze în ordine crescătoare elementele pozitive ale unui vector cu  $n$  numere reale, fără a modifica pozițiile elementelor negative
56. Să se ordoneze crescător un vector de  $n$  elemente numere reale folosind **metoda selecției**. Să se afișeze în ordine descrescătoare elementele impare ale unui vector cu  $n$  numere întregi, fără a modifica pozițiile numerelor pare.
57. Se citește din fișierul MAT.IN o matrice cu  $n$  linii și  $m$  coloane conținând numai elemente de 0 și 1. Să se determine **numărul de zerouri** din matrice. Și să se afișeze numărul liniei cu suma elementelor maximă, dacă matricea conține mai multe linii cu suma elementelor maximă, se va afișa numărul primei linii.
58. Se citesc din fișierul BAC.IN pe mai multe rânduri mai multe numere naturale de cel mult 3 cifre despărțite prin spațiu (pe o linie cel mult 100 numere) Scrieți în fișierul BAC.OUT (pe fiecare linie numerele din fișierul inițial dar **sortate crescător**.) numerele din fișierul BAC.IN sortate crescător, câte  $n$  numere pe linie, unde  $n$  este un număr natural nenul citit de la tastatură,  $n < 100$ .

Exemplu:

BAC.IN

3 8 5

8 3 9

4 1

și  $n=3$

BAC.OUT

3 5 8            1 3 3

3 8 9            4 5 8

1 4              8 9

59. Realizați un program care afișează **maximul** a  $n$  numere citite de la tastatură. ( $n < 50$ , numerele sunt naturale de cel mult 5 cifre) și numărul de apariții ale acestuia în șir.
60. Să se realizeze un program care afișează numărul de apariții al **elementului minim** dintr-un șir dat cu  $n$  elemente numere reale.
61. Din fișierul "numere.in" se citesc mai multe numere întregi. Ele sunt scrise în fișier pe mai multe linii. Numerele de pe aceeași linie sunt separate prin spațiu. Să se scrie un program care construiește fișierul text "numere.out" conținând numărul **cel mai mare** și numărul **cel mai mic** din (fiecare linie a) fișierul "numere.in", urmate, fiecare, de numărul de apariții. (Cele două numere se vor scrie pe aceeași linie separate prin spațiu.)  
 Exemplu: "numere. in" :  
                   10 9 3 57 3  
                   32 53 9 53 12 15 53  
                   6  
 "numere. out" :  
                   3 57                   3 2     (minimul este 3 și apare de 2 ori)  
                   9 53                   53 3    (maximul din fișier este 53 și apare de 3 ori)  
                   6 6
62. Se citește de la tastatură un text cu cel mult 250 de caractere. Afișați **numărul de vocale** al textului.
63. Scrieți un program care citește un text. Să se determine **numărul de cifre** din text.
64. Se citește din fișierul BAC.IN un text pe mai multe linii, fiecare linie conține maxim 150 caractere. Scrieți în fișierul BAC.OUT același text dar în care să **schimbe literele** mici cu litere mari și literele mari cu litere mici.  
 Exemplu:  
 Bac.in  
 Tata Este la  
 PiatA?  
 Bac. out  
 TATA eSTE LA  
 pIATa?
65. Să se calculeze produsul primilor  $n$  ( $n < 11$ ) termeni ai șirului lui **Fibonacci** definit astfel:  $f_1=1, f_2=1, f_n=f_{n-1}+f_{n-2}$  pentru  $n \geq 3$ ;  
 Exemplu:  $n=6$ , produsul este 240,
66. Să se determine dacă 2 numere naturale nenule sunt **termeni consecutivi ai șirului lui Fibonacci**.  
 Exemplu: 55 și 89 sunt termeni consecutivi ai șirului lui Fibonacci.
67. Pentru un număr natural nenul  $n$  să se determine suma primelor  $n$  numere naturale prime.  
 Exemplu:  $n=5$  se va afișa valoarea 28 ( $2+3+5+7+11$ )
68. Să se scrie un program în care să se calculeze expresia:  $E = \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \dots + \frac{1}{n}$ , unde  $n$  este un număr natural dat. Valoarea expresiei se va afișa sub formă de fracție.  
 Exemplu:  $n=5$ , se va afișa  $\frac{77}{60}$
69. Calculați suma  $S = x + x^2 + x^3 + \dots + x^n$  cu  $x$  și  $n$  citite de la tastatură,  $x$  număr întreg de maxim 2 cifre,  $n$  număr natural,  $n < 10$ .
70. Să se scrie un program în care să se calculeze expresia  $E = 1 + 1*2 + 1*2*3 + \dots + 1*2*3* \dots *n$ , unde  $n$  este un număr natural nenul dat.

71. Să se scrie un program în care să se calculeze expresia  $E=1*(1+2)*(1+2+3)*\dots*(1+2+3+\dots+n)$ , unde  $n$  este un număr natural nenul dat,  $n < 10$ .
72. Să se scrie un program în care să se calculeze expresia  $E=1*3-2*5+3*7-\dots\pm n*(2*n+1)$ , unde  $n$  este un număr natural dat.
73. Se dau doi vectori  $u$  și  $v$  cu  $n$  componente numere întregi. Să se calculeze produsul scalar al acestora,  $u[1]*v[1]+u[2]*v[2]+\dots+u[n]*v[n]$ .
74. Sa se calculeze **produsul a două numere naturale/întregi** prin adunări repetate și câtul și restul împărțirii celor două numere prin scăderi repetate. \_\_\_\_\_
75. Se citește din fișierul MATRICE.TXT o matrice pătratică de ordin  $n$ . Pe prima linie din fișier se găsește numărul de linii, iar pe liniile următoare elementele matricei. Realizați un program care verifică dacă **matricea este simetrică față de diagonala principală**.
76. Să se realizeze un program care să testeze dacă un vector cu  $n$  elemente întregi poate reprezenta o **mulțime** ( în sens matematic).
77. Se dau doi vectori  $A$  și  $B$  cu  $n$ , respectiv  $m$  elemente ce reprezintă două mulțimi. Să se determine mulțimea ce reprezintă intersecția celor două mulțimi date.
78. Să se scrie un program care citește două numerele naturale nenule  $n$  și  $m$  și elementele întregi ale unei matrice cu  $n$  linii și  $m$  coloane. Să se interschimbe elementele liniei  $x$  cu elementele liniei  $y$ ,  $x$  și  $y$  două numerele naturale nenule mai mici decât  $n$ . Să se afișeze matricea obținută.
79. Să se realizeze un program care calculează **produsul elementelor** de sub diagonala secundară a unei matrice pătratice de ordin  $n$  ( $n < 10$ ).
80. Să se realizeze un program care calculează **suma elementelor** de deasupra diagonalei secundare a unei matrice pătratice de ordin  $n$  ( $n < 15$ ).
81. În fișierul NUMERE.TXT se află mai multe numere naturale scrise cu spațiu între ele. Să se creeze fișierul PARE.TXT care sa conțină, cate una pe linie doar acele valori din fișierul NUMERE.TXT care sunt **numere pare**, iar pe ultimul rând numărul de numere impare din fișierul de intrare.
82. Se citește un număr natural nenul  $n$  și  $n$  valori numere naturale. Să se realizeze un program care construiește un vector cu **elemente pare**.
83. Se citește un număr natural nenul  $n$  și  $n$  valori numere naturale. Să se realizeze un program care construiește un vector cu **elemente impare**.
84. Fiind dat un vector  $v$  cu  $n$  componente numere întregi, să se afișeze de câte ori găsim două **elemente consecutive** egale.
85. Fiind dat un vector  $v$  cu  $n$  componente numere întregi, să se **insereze** la mijlocul vectorului media aritmetică a elementelor sale.
86. Fiind dat un vector  $v$  cu  $n$  componente numere întregi, să se afișeze **elementele pare** de pe pozițiile impare din vector și numărul acestora.
87. Fiind dat un vector  $v$  cu  $n$  componente numere întregi, să se afișeze toate perechile de **elemente consecutive** cu proprietatea că al doilea element al perechii este egal cu suma cifrelor primului.

88. Fiind dat un vector  $v$  cu  $n$  componente numere întregi și un număr natural  $p$  mai mic sau egal cu  $n$ , să se **șteargă** elementul aflat pe poziția  $p$  a vectorului și să se afișeze elementele vectorului după eliminare.
89. Fiind dat un vector  $v$  cu  $n$  componente numere întregi și două numere naturale  $p$  și  $q$  mai mici sau egal cu  $n$ , să se **interschimbe** valoarea aflată pe pozițiile  $p$  cu valoarea aflată pe poziția  $q$ .
90. Fiind dat un vector  $v$  cu  $n$  componente numere întregi. Să se afișeze un mesaj corespunzător dacă elementele vectorului sunt sau nu **ordonate** crescător.