**SUBIECTE ATESTAT PROGRAMARE – PROBLEME PROPUSE**

**An școlar 2020 – 2021**

1. Se citesc de la tastatură n numere naturale nenule (n≥2). Calculați și afișați cel mai mare divizor comun al celor n numere.

**Exemplu**: pentru n=4, dacă se citesc numerele 16 18 400 128 se va afișa 2

1. Se citesc n triplete de numere întregi (n≥2). Să se numere câte triplete de numere pot reprezenta laturile unui triunghi (echilateral, isoscel, dreptunghic sau oarecare). Dacă nu există astfel de triplete se afișează mesajul “NU EXISTĂ TRIUNGHIURI”.

**Exemplu**: pentru n=3, dacă se citesc tripletele -3 4 5 ; 3 3 4; 3 4 5 se va afișa

1 triunghi isoscel

1 triunghi dreptunghic

1. Se citesc numere întregi până la citirea valorii zero. Să se afișeze media aritmetică a numerelor prime și media aritmetică a numerelor neprime.

**Exemplu**: pentru numerele 115 45 11 23 2 0 se va afișa

media numerelor neprime este 80

media numerelor prime este 12

1. Se citesc a și b numere naturale nenule, reprezentând capetele unui interval deschis (a≠b). Se citesc apoi n numere naturale cu cel mult 8 cifre. Să se afișeze și să se numere câte dintre acestea au suma cifrelor în intervalul respectiv. Dacă nu există astfel de numere se afișează mesajul “NU EXISTĂ”.

**Exemplu**: pentru a=5, b=15, n=5 dacă se citesc numerele 32 1000 308 1234 99 se va afișa 2

1. Se citesc numere întregi până la citirea numărului zero. Să se afișeze media aritmetică a divizorilor pari pentru fiecare număr citit.

**Exemplu**: pentru numerele 115 16 20 0 se va afișa 0 7.5 9

1. Se citește un șir de n numere naturale. Afișați media aritmetică a numerelor care sunt perfecte(numărul perfect este egal cu suma divizorilor săi, în afară de el însuși). Dacă nu există astfel de numere se afișează mesajul “NU EXISTĂ”.

**Exemplu**: pentru n=3, dacă se citesc 28 14 6 se va afișa 17

1. Se citesc n numere naturale de maxim 9 cifre. Pentru fiecare număr citit, să se calculeze cifra maximă și cifra minimă, precum și numărul lor de apariții. Dacă toate cifrele sunt identice, se va afișa mesajul “CIFRE EGALE “.

**Exemplu**: pentru n=3, dacă se citesc numerele

5347372 se va afișa *cifra maxima 7 numar de aparitii 2, cifra minima 2 numar de aparitii 1*

22222 se va afișa *cifre egale*

646464 se va afișa *cifra maxima 6 numar de aparitii 3, cifra minima 4 numar de aparitii 3*

1. Se citește un șir de n numere naturale. Să se afișeze câte cifre au fiecare dintre numerele citite în reprezentarea în baza de numerație 2.

**Exemplu**: pentru n=2, pentru numerele citite:

45 se afișează 6 cifre binare

14 se afișează 4 cifre binare

1. Din fișierul Atestat.in se citesc cel mult 100 de numere naturale. Pentru fiecare număr din fișier să se calculeze factorul prim minim și puterea la care apare acesta în descompunerea numărului în factori primi. Rezultatele se vor afișa în fisierul Factor.out pe linii diferite.

**Exemplu:**

**Atestat.in Factor.out**

49 45 16 7 2

3 2

2 4

1. În fișierul Atestat.in se află pe prima linie valoarea n, număr natural nenul, apoi, pe următoarea linie, n numere întregi. Afișați în fișierul Palindrom.out media aritmetică a numerelor citite din fișier care sunt numere palindroame, sau, mesajul “NU EXISTĂ” dacă niciun număr nu este palindrom.

**Exemplu:**

**Atestat.in Palindrom.out**

4

789 1221 14 6 613.5

11. Se dă un vector cu **n** elemente (n<=100), numere întregi. Să se afișeze elementele vectorului în ordinea următoare: primul, ultimul, al doilea, penultimul, al treilea, antepenultimul etc.

**Exemplu.**

Pentru n=7 și vectorul 10 -2 -81 15 3 47 -41 se va afișa 10 -41 -2 47 -81 3 15.

12. Se citește un vector cu **n** elemente (n<=100), numere întregi. Să se determine câte elemente ale vectorului sunt egale cu suma dintre cea mai mare și cea mai mică valoare din vector. În cazul în care nu există astfel de elemente se va afișa mesajul N**u există.**

**Exemple.**

Pentru n=7 și vectorul -3 10 3 7 -2 7 4 se va afișa 2, iar pentru n=7 și vectorul -3 10 6 -2 5 4 -1 se va afișa **Nu există**.

13. Din fișierul **vector.in** se citește un vector cu **n** elemente (n<=100), numere naturale. Să se verifice dacă există numere prime printre cele citite. Să se afișeze mesajul **DA** în fișierul **vector.out** dacă există numere prime sau **NU** dacă nu există.

**Exemple.**

Pentru n=7 și vectorul 24 25 17 6 3 4 9 se va afișa **DA**, iar pentru n=5 și vectorul 6 21 8 15 32 se va afișa **NU**.

14. Se citește un vector cu **n** elemente (n<=100), numere întregi. Să se șteargă din vector toate elementele pare apoi să se afișeze vectorul obținut.

**Exemplu.**

Pentru n=10 și vectorul 10 -2 -2 15 -3 4 4 7 8 1 se va afișa 15 -3 7 1.

15. Se dă un vector cu **n** elemente (n<=100), numere naturale și un număr **k** natural (k<n). Să se ordoneze crescător primele **k** elemente ale vectorului și descrescător ultimele **n-k** elemente. Vectorul obținut se va afișa în fișierul **vector.out**.

**Exemplu.**

Pentru n=9, k=4 și vectorul 11 5 27 2 12 30 3 5 5 se va afișa 2 5 11 27 30 12 5 5 3.

16. Se dă un vector cu **n** elemente (n<=100), numere întregi. Să se insereze după fiecare element par din vector un număr **x** întreg dat de la tastatură și să se afișeze vectorul obținut.

**Exemplu.**

Pentru n=9, x=13 și vectorul 10 -2 3 15 -3 4 4 7 8 se va afișa 10 13 -2 13 3 15 -3 4 13 4 13 7 8 13.

17. Se dă un vector cu **n** elemente (n<=100), numere întregi. Să se calculeze suma elementelor din secvența determinată de primul și ultimul element par. Vectorul va conține cel puțin un element par.

**Exemplu.**

Pentru n=7 și vectorul 13 2 3 4 5 8 11 se va afișa 22. Explicație 2+3+4+5+8=22

18. Se dă un număr natural **n** cu maxim 9 cifre. Dacă numărul are toate cifrele distincte se va afișa **DA**, dacă cel puțin două cifre sunt egale se va afișa **NU**. Rezolvați problema astfel încât să fie eficientă din punct de vedere al timpului de execuție.

**Exemple.**

Pentru n=987123450 se va afișa DA; pentru n=122997321 se va afișa NU.

19. Se dau doi vectori **x** și **y**, fiecare având **n** elemente (n<=100), numere întregi. Să se construiască și să se afișeze vectorul **z** după formula z=2x+y2 în fișierul **vector.out**.

**Exemplu.**

Pentru n=3, x=(1,2,3), y=(4,5,6) se va afișa 18 29 42.

20. Se dă un vector cu **n** elemente (n<=100), numere naturale distincte. Să se ordoneze crescător elementele situate înaintea valorii maxime din vector și descrescător elementele situate după această valoare.

**Exemplu.**

Pentru n=9 și vectorul 3 2 11 14 17 8 4 5 7 se va afișa 2 3 11 14 17 8 7 5 4.

21. Se dă o matrice cu **n** linii și **m** coloane (n, m numere naturale <=100), elemente numere întregi. Să se ordoneze crescător prima linie a matricei, apoi să se afișeze în fișierul **matrice.out**.

**Exemplu.**

Pentru n=3, m=4 si matricea se va afișa matricea

22. Se dă o matrice pătratică de dimensiune **n** (n număr natural<=100), elemente numere întregi. Să se afișeze ce diagonală are suma elementelor mai mare sau dacă suma este egală.

**Exemplu.**

Pentru n=4 și matricea se va afișa mesajul ***Diagonala principală*** (Suma pe diagonala principală=113>Suma pe diagonala secundară=37)

23. Din fișierul **matrice.in** se citeșe o matrice cu **n** linii și **m** coloane (n, m numere naturale <=100, m impar), elemente numere întregi. Să se verifice dacă toate elementele coloanei din mijloc sunt impare și se va afișa un mesaj corespunzator.

**Exemplu.**

Pentru n=3, m=5 si matricea se va afișa mesajul **NU**.

24. Scrieți un program care construiește o matrice cu primele **n**\***n** (n<=100) numere naturale impare astfel: pe prima coloană elementele sunt ordonate crescător de sus în jos, pe a doua coloană descrescător de sus în jos, pe a treia coloană crescător de sus în jos ș.a.m.d. Matricea se va afișa pe ecran.

**Exemplu.**

Pentru n=4 se va genera maticea:

25. Scrieți un program care construiește o matrice cu **n** linii și **m** coloane (n,m numere naturale<=100) formată din primii n\*m termeni impari din șirul lui Fibonacci. Matricea se va afișa pe ecran.

**Exemplu.**

Pentru n=4, m=3 se va genera maticea:

26. Se dă un șir cu cel mult 100 caractere format din spații, litere mari și mici ale alfabetului englez. Cuvintele sunt separate printr-unul sau mai multe spații. Pentru cuvintele de lungime impară să se transforme litera din mijloc în literă mare.

**Exemplu.** Pentru textul: ”Azi avem examinarea pentru atestat”

Se va afișa: ”AZi avem examinarea pentru ateStat”

27. Scrieţi un program care citeşte de la tastatură un şir de cel mult 50 de caractere (litere mici şi mari ale alfabetului englez, cifre, puncte, virgule şi spaţii) şi afişează pe ecran cifra care apare de cele mai multe ori în şirul citit. Dacă textul nu conține cifre se va afișa mesajul **Nu exista**. Dacă există mai multe cifre cu număr maxim de apariții se va afișa oricare.

**Exemplu.** Pentru textul”Azi voi lua 9,5 la Baze de date 10 la Programare si 10 la Operare” Se poate afișa 1 sau 0.

28. Din fișierul **text.in** se citesc mai multe cuvinte. Cuvintele sunt scrise pe un rând și separate prin spații. Să se afișeze în ordine alfabetică cuvintele cu lungime impară. Dacă nu există astfel de cuvinte se va afișa **Nu exista**.

**Exemplu.** Dacă în fișier se găsesc cuvintele:  **”**litere mici sau mari din alfabetul englez cifre puncte virgule si spatii” Se va afișa șirul de cuvinte: ***alfabetul cifre din sau virgule***

29. Se dă un șir cu cel mult 100 caractere format din spații, litere mari și mici ale alfabetului englez. Cuvintele sunt separate printr-unul sau mai multe spații. Câte cuvinte au aceeași literă pe poziții consecutive?

**Exemplu.** Pentru textul: ”Testul pentru **alcool**  a fost **ambiguu**  **fiind** insuficient pentru **succesul** **operatiunii**” se va afișa 5.

30. Se dă un șir cu cel mult 100 caractere format din spații, litere mari și mici ale alfabetului englez. Cuvintele sunt separate printr-unul sau mai multe spații. Să se elimine cuvintele care incep și se termină cu aceeași literă. Dacă nu există astfel de cuvinte se va afișa **Nu există**.

**Exemplu.** Pentru textul ”La **examene** **avea** trac” se va afișa textul ”La trac”