1. ***(Sumă elemente impare pe poziţii pare)*** Se citeşte un număr natural nenul n (1<=n<=100). Scrieţi un program care citeşte un şir de n elemente numere întregi, apoi calculează şi afişează suma tuturor elementelor pare situate pe poziţii impare în şirul citit iniţial.
2. ***(Elemente consecutive egale)*** Se citeşte un număr natural nenul n (1<=n<=100). Scrieţi un program care citeşte un şir de n elemente numere întregi, apoi determină şi afişează de câte ori găsim două elemente pe poziţii consecutive egale între ele.
3. ***(Elemente din intervalul [a,b])*** *S*e citeşte un număr natural nenul n (1<=n<=100), a şi b (a < b, a, b numere întregi). Scrieţi un program care citeşte un şir de n elemente numere reale, apoi determină şi afişează câte elemente din vectorul iniţial se găsesc în intervalul [a,b]. Dacă nu există nici un astfel de eleent se sa afişa mesajul „NU EXISTĂ”.
4. ***(Răsturnatul unui şir)*** Se citeşte un număr natural nenul n (1<=n<=100). Scrieţi un program care citeşte un şir de n elemente numere întregi, apoi construieşte un alt şir (fără a folosi un vector suplimentar) cu elementele şirului dat citite invers.
5. ***(Vector de apariţii)*** Se citeşte un număr natural nenul n, de maxim 9 cifre. Scrieţi un program care afişează cifrele distincte ale numărului n şi numărul de apariţii a fiecărei cifre din n.
6. ***(Cel mai mic nr. construit din cifra dominantă)*** Se citeşte un număr natural nenul (n<=9). Scrieţi un program care citeşte un şir de n elemente numere naturale, apoi construieşte şi afişează cel mai mic număr natural care se poate alcătui luând prima cifră a fiecărui element al şirului.
7. ***(Sortarea elementelor nenule)*** Se citeşte un număr natural nenul n (1<=n<=100). Scrieţi un program care citeşte un şir de n elemente numere întregi, apoi sortează (ordonează) descrescător elementele nenule ale vectorului citit iniţial. La final se va afişa vectorul cu elementele nenule sortate descrescător. Elementele nule sunt lăsate pe loc. Se poate construi un vector suplimentar.
8. ***(Inserarea valorii 0 după fiecare element)*** Se citeşte un număr natural nenul n (1<=n<=100). Scrieţi un program care citeşte un şir de n elemente numere naturale, apoi inserează după fiecare element al vectorului initial valoare 0. La final se va afişa vectorul modificat prin inserarea valorii 0.

9. ***(Şergere aşa încât vectorul să aibă val. în ordinea strict crescătoare)*** Se citeşte un număr natural nenul n (1<=n<=100). Scrieţi un program care citeşte un şir de n elemente numere naturale, apoi (fără a şterge prima valoarea din vector) să se şteargă un număr minim de valori din vectorul iniţial aşa încât la final să se obţină un şir strict crescător de elemente. La final se va afişa vectorul obţinut.
10. ***(Permutări circulară)*** Se citeşte un număr natural nenul n (1<=n<=100). Scrieţi un program care citeşte un şir de n elemente numere naturale, apoi afişează permutările circulare ale vectorului. O permutare circulară se obţine prin rotirea elementelor vectorului cu i poziţii (i<n)

.

PRELUCRĂRI VECTORI cu/din FIŞIERE

1. Fişierul **date.in** conţine două linii. Pe prima linie este scris un număr natural nenul **n,** (**5<n<10**). Pe cea de-a doua linie a fişierului sunt scrise **n** numere întregi separate prin câte un spaţiu, formate fiecare din cel mult **4** cifre, reprezentând un şir de **n** întregi .

Să se scrie un program în limbajul **Pascal/C/C++**, care:

a) să afişeze pe ecran, în linie, valorile absolute ale numerelor din şir, separate prin câte un spaţiu;
b) să afişeze pe ecran, în linie, numărul de divizori pozitivi proprii pentru fiecare număr din şir, separaţi prin câte un spaţiu și pe câte un rând pentru fiecare număr din șir
c) să scrie în fişierul **date.out**, pe prima linie, toate numerele impare din şir, în ordine crescătoare, separate prin câte un spaţiu.

2. Fişierul **date.in** conţine două linii. Pe prima linie este scris un număr natural nenul **n,** (**5<n<30**). Pe cea de-a doua linie a fişierului sunt scrise **n** numere naturale separate prin câte un spaţiu, formate fiecare din cel mult **9** cifre, reprezentând un şir de **n** numere naturale.

Să se scrie un program în limbajul **Pascal/C/C++**, care:

a) să afişeze pe ecran, în linie, numerele din şir, separate prin câte un spaţiu;
b) să afişeze pe ecran, pe linii diferite, cel mai mic număr **a** şi cel mai mare număr **b** din şirul dat;
c) să scrie în fişierul **date.out** cel mai mare divizor comun al numerelor **a** şi **b,** determinate la punctul b).

Exemplu:

date.in		**Date de ieşire:**
6	**a)**	**123 55 372 3465 242 44**
123 55 372 3465 242 44	**b)**	**44** **3465**
		Fişierul **date.out** conţine**:**
	c)	**11**

3. Fişierul **date.in** conţine două linii. Pe prima linie este scris un număr natural nenul **n,** (**5<n<30**). Pe cea de-a doua linie a fişierului sunt scrise **n** numere naturale separate prin câte un spaţiu, formate fiecare din cel mult **9** cifre, reprezentând un şir de **n** numere naturale.

Să se scrie un program în limbajul **Pascal/C/C++**, care:

a) să afişeze pe ecran, în linie, toate numerele din şir, separate prin câte un spaţiu;
b) să afişeze pe ecran, în linie, toate numerele din şir formate numai din cifre pare (dacă nu există astfel de numere în şir se va afişa mesajul **"NU EXISTĂ NUMERE NUMAI CU CIFRE PARE"**);
c) să citească de la tastatură două numere naturale nenule **p1** şi **p2** (**1<p1<p2<n**), să ordoneze descrescător numerele din şir situate între poziţiile **p1** şi **p2**, inclusiv, şi să scrie noul şir în fişierul **date.out**, pe o linie, numerele separându-se prin câte un spaţiu.

Exemplu: de la tastatură se citesc: **p1=2** şi **p2=4**

date.in		**Date de ieşire:**
6	a)	1233 22 1785 56 15657 457
1233 22 1785 56 15657 457	b)	22
		Fişierul **date.out** conţine:
	c)	1233 1785 56 22 15657 457

4. Fişierul **date.in** conţine două linii. Pe prima linie este scris un număr natural nenul **n,** (**5<n<30**). Pe cea de-a doua linie a fişierului sunt scrise **n** numere reale separate prin câte un spaţiu, reprezentând un şir de **n** numere reale.
Să se scrie un program în limbajul **Pascal/C/C++**, care:

a) să afişeze pe ecran, în linie, toate numerele din şir, separate prin câte un spaţiu;
b) să afişeze pe următoarea linie a ecranului, media aritmetică a numerelor negative din şir, cu o precizie de 2 zecimale (dacă şirul nu conţine numere negative se va afişa **0**);
c) să citească de la tastatură două numere naturale nenule **p1** şi **p2** (**1<p1<p2<n**), să ordoneze crescător numerele din şir situate între poziţiile **p1** şi **p2**, inclusiv, şi să scrie noul şir în fişierul **date.out**, pe o linie, numerele separându-se prin câte un spaţiu.

Exemplu: de la tastatură se citesc: **p1=2** şi **p2=4**

date.in		**Date de ieşire:**
6	a)	-56.765 2.3 4.56 -1.2 -1.8 3
-56.765 2.3 4.56 -1.2 -1.8 3	b)	-19.92
		Fişierul **date.out** conţine:
	c)	-56.765 -1.2 2.3 4.56 -1.8 3

5. Fişierul **date.in** conţine două linii. Pe prima linie este scris un număr natural nenul **n,** (**5<n<30**). Pe cea de-a doua linie a fişierului sunt scrise **n** numere naturale separate prin câte un spaţiu, formate fiecare din cel mult **4** cifre, reprezentând un şir de **n** numere naturale distincte.
Să se scrie un program în limbajul **Pascal/C/C++**, care:

a) să afişeze pe ecran, în linie, toate numerele din şir, separate prin câte un spaţiu;
b) să afişeze pe ecran, pe linii diferite, cel mai mic număr din şir şi poziţia acestuia;
c) să scrie în fişierul **date.out**, pe o linie, separate prin câte un spaţiu, toate numerele *perfecte* din şirul dat (dacă nu există astfel de numere, se va se va afişa mesajul "**NU EXISTĂ NUMERE PERFECTE**"). Un număr este *perfect* dacă este egal cu suma divizorilor lui pozitivi, exceptându-l pe el însuşi, de exemplu: **6 = 1+2+3**.

Exemplu:

date.in		**Date de ieşire:**
6	a)	28 11 81 496 6 100
28 11 81 496 6 100	b)	6
		5
		Fişierul **date.out** conţine:
	c)	28 496 6

6. Fişierul **date.in** conţine două linii. Pe prima linie este scris un număr natural nenul **n,** (**5<n<30**). Pe cea de-a doua linie a fişierului sunt scrise **n** numere naturale separate prin câte un spaţiu, formate fiecare din cel mult **4** cifre, reprezentând un şir de **n** numere naturale distincte.

Să se scrie un program în limbajul **Pascal**/**C**/**C++**, care:

a) să afişeze pe ecran, în linie, toate numerele din şir, separate prin câte un spaţiu;
b) să afişeze pe ecran, pe linii diferite, cel mai mare număr din şir şi poziţia acestuia;
c) să scrie în fişierul **date.out**, pe o linie, separate prin câte un spaţiu, numerele *supraperfecte* din şirul dat (dacă nu există astfel de numere, se va se va afişa mesajul "**NU EXISTĂ NUMERE SUPRAPERFECTE**"). Un număr este *supraperfect* dacă este mai mic decât suma divizorilor lui pozitivi, exceptându-l pe el însuşi, de exemplu: **12<1+2+3+4+6**.

Exemplu:

date.in							**Date de ieşire:**					
6						**a)**	**22**	**12**	**121**	**20**	**18**	**13**
22	**12**	**121**	**20**	**18**	**13**	**b)**	**121**					
							3					
						c)	Fişierul **date.out** conţine**:**					
							12	**20**	**18**			

7. Fişierul **date.in** conţine două linii. Pe prima linie este scris un număr natural nenul **n,** (**5<n<30**). Pe cea de-a doua linie a fişierului sunt scrise **n** numere naturale separate prin câte un spaţiu, formate fiecare din cel mult **4** cifre, reprezentând un şir de **n** numere naturale. Şirul conţine cel puţin două numere pare.

Să se scrie un program în limbajul **Pascal**/**C**/**C++**, care:

a) să afişeze pe ecran, în linie, toate numerele din şir, separate prin câte un spaţiu;
b) să afişeze pe următoarea linie a ecranului, media aritmetică a tuturor numerelor pare din şir ;
c) să scrie în fişierul **date.out**, pe o linie, separate prin câte un spaţiu, numerele de tip palindrom din şirul dat (dacă nu există astfel de numere, se va afişa mesajul "**NU EXISTĂ NUMERE PALINDROM**"). Un număr este palindrom dacă numărul citit de la stânga la dreapta este egal cu numărul citit de la dreapta la stânga, de exemplu: **33**, **141**, **2552**.

Exemplu:

date.in							**Date de ieşire:**					
6						**a)**	**2552**	**56**	**32**	**444**	**46**	**1221**
2552	**56**	**32**	**444**	**46**	**1221**	**b)**	**626**					
							Fişierul **date.out** conţine**:**					
						c)	**2552**	**444**	**1221**			

8. Fişierul **date.in** conţine două linii. Pe prima linie este scris un număr natural nenul **n,** (**5<n<30**). Pe cea de-a doua linie a fişierului sunt scrise **n** numere naturale separate prin câte un spaţiu, formate fiecare din cel mult **4** cifre, reprezentând un şir de **n** numere naturale. Şirul conţine cel puţin două numere impare.

Să se scrie un program în limbajul **Pascal**/**C**/**C++**, care:

a) să afişeze pe ecran, în linie, în ordinea inversă citirii, toate numerele din şir, separate prin câte un spaţiu;
b) să afişeze pe ecran, în linie, numărul de cifre din care este format fiecare număr din şirul iniţial, numerele din linie separându-se prin câte un spaţiu;
c) să scrie în fişierul **date.out**, pe prima linie, suma tuturor numerelor impare din şir.

Exemplu:

date.in		Date de ieşire:
6	a)	1001 242 2 71 555 13
13 555 71 2 242 1001	b)	2 3 2 1 3 4
		Fişierul date.out conţine:
	C)	1640

9. Fişierul **date.in** conţine două linii. Pe prima linie este scris un număr natural nenul **n,** (**5<n<30**). Pe cea de-a doua linie a fişierului sunt scrise **n** numere întregi separate prin câte un spaţiu, formate fiecare din cel mult **4** cifre, reprezentând un şir de **n** numere întregi.

Să se scrie un program în limbajul **Pascal**/**C**/**C++**, care:

a) să afişeze pe ecran, în linie, în ordinea inversă citirii, toate numerele din şir, separate prin câte un spaţiu;
b) să afişeze pe ecran, în linie, separate prin câte un spaţiu, toate numerele prime din şir (dacă nu există numere prime în şir, se va afişa pe ecran un răspuns corespunzător);
c) să scrie în fişierul **date.out**, pe prima linie, suma tuturor numerelor pozitive din şir (dacă nu există numere pozitive în şir se va scrie în fişierul **date.out** un mesaj corespunzător) .

Exemplu:

date.in		Date de ieşire:
6	a)	11 -242 -2 41 -555 1234
1234 -555 41 -2 -242 11	b)	41 -2 11
		Fişierul date.out conţine:
	c)	1286