**Subiecte programare în Pascal/C/C++,**

**pentru clasele de matematică-informatică, intensiv informatică**

1. În fişierul *“vector.txt”* pe primul rând se află un număr natural n(n<1000), iar pe a doua linie se găsesc n numere naturale despărţite printr-un spaţiu, cu cel mult 5 cifre fiecare. Se consideră că n reprezintă dimensiunea unui vector iar cele n elemente de pe al doilea rând sunt elementele vectorului. Să se determine suma elementelor vectorului care sunt palindrom. (Se numeşte palindrom un număr care citit de la stânga la dreapta şi de la dreapta la stânga este acelaşi). Suma rezultată (sau un mesaj corespunzător, dacă nu există niciun palindrom) se va afişa în fişierul *“suma.txt”*.
2. În fişierul *“vector.txt”*, pe primul rând se află un număr natural n(n<1000), iar pe a doua linie se găsesc n numere naturale, cu cel mult 6 cifre fiecare, despărţite printr-un spaţiu. Se consideră că n reprezintă dimensiunea unui vector iar cele n elemente de pe al doilea rând sunt elementele vectorului; citirea lor se face folosind un subprogram. Afişaţi pe ecran câte elemente din vector au exact p divizori proprii, unde p este un număr natural citit de la tastatură.
3. Se citeşte din fişierul *“vector.txt”* un vector de numere naturale, n se află pe primul rând(n<1000), iar cele n elemente ale vectorului se află pe al doilea rând. Afişaţi câte numere perfecte sunt în vector. Un număr este *perfect* dacă este egal cu suma divizorilor săi mai mici decât el (ex: 6=1+2+3) – se va folosi un subprogram care verifică această proprietate. Afişarea se face în fişierul *“solutie.out”*.
4. Se citeşte din fişierul *“vector.txt”* un vector de numere întregi (cu cel mult 5 cifre fiecare) astfel: n(n<1000)se află pe primul rând iar cele n elemente ale vectorului se află pe al doilea rând. Citirea se face folosind un subprogram. Să se scrie un program care calculează şi afişează maximul dintre elementele pare ale vectorului sau afişează un mesaj în cazul în care nu există numere pare în vector. Afişarea se face în fişierul *“solutie.out”*.
5. Se consideră o matrice cu n linii şi m coloane cu elemente de tip real (n, m şi matricea se citesc de la tastatură, 1≤n≤50, 1≤m≤50). Afişaţi pe ecran: matricea(fiecare linie pe câte un rând, iar elementele de pe fiecare rând despărţite de câte un spaţiu) şi maximul de pe fiecare coloană a matricei, maximele se scriu pe acelaşi rând, despărţite de căte un spaţiu.

Exemplu:

|  |  |
| --- | --- |
| n=3, m=4 şi matricea:  -1.6 18 2.1 0.3  141 3 0.9 0.5  200 0.22 14 1000 | Se va afişa pe ecran:  -1.6 18 2.1 0.3  141 3 0.9 0.5  200 0.22 14 1000  200 18 14 1000 |

1. În fişierul „Matrice.in” pe primul rând se găsesc două numere naturale n şi m (1≤n≤50, 1≤m≤50), despărţite printr-un spaţiu, care reprezintă numărul de linii, respectiv de coloane ale unei matrici de numere reale. Apoi pe următoarele n linii se găsesc, pe fiecare linie câte m numere reale, despărţite printr-un spaţiu, care reprezintă elementele matricei. Să se construiască o matrice B cu 2 coloane care să conţină indicii elementelor subunitare pozitive ale matricei A. Afişaţi pe ecran cele 2 matrici: cea iniţială şi matricea nou formată B, în caz că se formează o astfel de matrice, sau tipăriţi un mesaj în caz contrar.

Exemplu:

|  |  |
| --- | --- |
| Matrice.in  3 4  -1.6 18 2.1 0.3  141 3 0.9 0.5  200 0.22 14 1000 | Matricea B va fi:  1 4  2 3  2 4  3 2 |

1. Se consideră o matrice cu n linii şi m coloane de numere întregi (n, m şi matricea se citesc de la tastatură, 1≤n≤50, 1≤m≤50). Să se afişeze pe ecran suma elementelor pare de pe fiecare linie, fiecare dintre cele n sume, se scrie pe câte un rând nou, astfel

”Pe linia …. suma este…..”/sau mesajul ”Pe linia ….nu exista numere pare”

1. În fişierul „Matrice.in” pe primul rând se găsesc două numere naturale n şi m (1≤n≤50, 1≤m≤50), despărţite printr-un spaţiu, care reprezintă numărul de linii, respectiv de coloane a unei matrici de numere întregi (cu cel mult 9 cifre fiecare). Apoi, pe următoarele n linii se găsesc, pe fiecare linie câte m numere întregi, despărţite printr-un spaţiu, care reprezintă elementele matricei. Să se afişeze numărul elementelor impare de pe fiecare coloană. Tipărirea se face în fişierul „Impar.out”, pe fiecare rând câte un număr.

Exemplu:

|  |  |
| --- | --- |
| Matrice.in  3 4  16 18 21 3  141 3 19 5  200 22 14 10 | Impar.out  1  1  2  2 |

1. Să se construiască un tablou pătratic de dimensiune n x n (1≤ n≤50) cu primele n2 numere pare începând cu 2.(n se citeşte de la tastatură). Tabloul se va afişa pe ecran, fiecare linie pe câte un rând nou, iar elementele de pe fiecare rând se scriu separate de câte un spaţiu

Exemplu: Pentru n=4 se va afişa pe ecran:

2 4 6 8

10 12 14 16

18 20 22 24

26 28 30 32

1. Se citeşte de la tastatură un vector de n numere reale (se dă n natural, 1≤n≤100). Scrieţi programul care afişează vectorul citit, valoarea elementului minim din vector precum şi poziţiile pe care acesta apare în vector. Pentru citirea, afişarea elementelor vectorului precum şi pentru determinarea minimului din vector se vor folosi subprograme.
2. În fişierul *„numere.in”* pe prima linie este un număr natural n (1≤n≤50), iar pe următoarea linie se găsesc n numere naturale de maxim 9 cifre fiecare, despărţite printr-un spaţiu. Să se scrie un program care să calculeze cel mai mare divizor comun al celor n numere de pe linia a doua a fişierului. Rezultatul se va afişa în fişierul *„cmmdc.out”*.

Exemplu:

|  |  |
| --- | --- |
| numere.in  5  16 18 210 14 30 | cmmdc.out  2 |

1. Creaţi fişierul *„trei.out"* care să conţină pe prima linie toate numerele de trei cifre divizibile cu suma cifrelor lor.
2. În fişierul *„graf.in"* pe primul rând se găseşte numărul de vârfuri n (1≤n≤50) al unui graf neorientat, apoi pe următoarele n rânduri se află matricea de adiacenţă a acestui graf neorientat. Să se determine gradul fiecărui vârf. Afişarea se face pe ecran, fiecare grad se scrie pe câte un rând nou.
3. În fişierul *„graf.in"* pe primul rând se găseşte n(n≤50), numărul de vârfuri al unui graf neorientat, apoi pe următoarele n rânduri matricea de adiacenţă a acestui graf neorientat. Să se scrie muchiile grafului în fişierul *“muchii.txt”* (pe fiecare rând se vor scrie extremităţile unei muchii separate printr-un spaţiu).
4. În fişierul *„graf.in"* se găseşte n, (n≤50), numărul de vârfuri al unui graf neorientat, apoi pe următoarele n rânduri matricea de adiacenţă a acestui graf neorientat. Să se afişeze maximul gradelor vârfurilor din graf. Afişarea se face pe ecran.
5. Să se verifice dacă un graf neorientat cu n vârfuri este un graf complet (n≤50). Numărul de vârfuri şi matricea de adiacenţă a grafului se găsesc în fişierul text *„adiacent.txt”.* Mesajul se va afişa în fişierul text *„complet.txt”*.
6. În fişierul *„matrice.in”* se găseşte pe primul rând un număr natural n, (n≤50), iar pe fiecare din următoarele n rânduri, câte n numere întregi(cu cel mult 9 cifre fiecare) despărţite de câte un spaţiu, care reprezintă dimensiunea, respectiv elementele unei matrici pătratice de dimensiune n\*n. Să se verifice dacă matricea este simetrică faţă de diagonala principală, tipărindu-se pe ecran un mesaj corespunzător.
7. Se citeşte de la tastatură un şir de caractere s, de lungime cel mult 100 de caractere, ce reprezintă un cuvânt din dicţionarul limbii române. Să se scrie pe ecran acest cuvânt în limba “păsărească”, adică șirul obținut prin inserarea în şirul inițial după fiecare vocală subşirul format din litera “p” şi vocala respectivă.

Exemplu: dacă şirul s este “acoperire” va rezultă noul şir s: ”a**pa**co**po**pe**pe**ri**pi**re**pe**”.

1. Se citesc de la tastatură n, (n≤50), număr natural şi apoi n şiruri de caractere, cu cel mult 30 de caractere fiecare, ce reprezintă nume de persoane. Să se scrie în fişierul „litera.txt” toate numele de persoane, fiecare pe un rând, care încep cu o anumită literă, citită de la tastatură.
2. Se citeşte de la tastatura un număr natural n, (n≤50), apoi se citesc de la tastatură n şiruri de caractere, cu cel mult 30 de caractere fiecare. Să se scrie în fişierul *„litere.txt”* toate şirurile, fiecare pe un rând, care încep şi se termină cu acelaşi caracter.
3. Se citeşte de la tastatură un şir de caractere s ce reprezintă un cuvânt din dicţionarul limbii române, cu cel mult 30 de caractere, litere mici ale alfabetului englez. Afişaţi pe ecran toate sufixele şirului s.

Ex: dacă şirul s este “vara” se vor afişa *a ra ara vara*

1. Se citeşte de la tastatură un şir de caractere s ce reprezintă un cuvânt din dicţionarul limbii române, cu cel mult 30 de caractere, litere mici ale alfabetului englez. Afişaţi pe ecran toate prefixele şirului s.

Ex: dacă şirul s este “vara” se vor afişa *v va var vara*

1. Se citeşte de la tastatură un şir de caractere, cu cel mult 30 de caractere, litere mici ale alfabetului englez, ce reprezintă un cuvânt din dicţionarul limbii române, care are cel puţin o vocală şi cel puţin o consoană. Afişaţi pe ecran şirul obţinut prin ştergerea vocalelor din şir.
2. Se citeşte de la tastatură un şir de caractere s ce reprezintă un cuvânt, cu cel mult 30 de caractere din dicţionarul limbii române. Afişaţi pe ecran şirul obţinut prin transformarea literelor mari în mici şi a celor mici în mari.
3. Se citeşte de la tastatură un număr natural n(n≤20), apoi se citesc n cuvinte de la tastatură, cu cel mult 30 de caractere fiecare. Să se afişeze pe ecran cuvintele care sunt palindrom (un cuvânt care citit de la sfârşit la început este acelaşi cu cuvântul iniţial; exemplu: “cojoc”).
4. Scrieţi definiţia completă a subprogramului *ordonare*, care primeşte prin intermediul parametrului n un număr natural nenul (**n**≤**1000**), ce reprezintă numărul efectiv de elemente ale unui vector **v** şi prin intermediul parametrului v un tablou unidimensional cu cel mult **1000** de numere întregi care pot avea maxim 9 cifre şi ordonează *crescător* elementele tabloului **v**.

În fişierul *„Numere.in”* pe prima linie se găseşte n, iar pe următoarele n rânduri se găsesc n numere întregi cu cel mult 9 cifre fiecare. Se cere să se tipărească pe ecran numerele în ordine *descrescătoare*, pe acelaşi rând, despărţite printr-un spaţiu.

Exemplu

|  |  |
| --- | --- |
| Numere.in  6  15  100  4  3  -32  18 | Se va tipări pe ecran:  100 18 15 4 3 -32 |

1. Se citesc de la tastatură două numere naturale n şi p (1≤p≤n≤10). Să se afişeze în fişierul „combin.out”, câte una pe linie, toate combinările numerelor de la 1 la n, luate câte p.

Exemplu

|  |  |
| --- | --- |
| pentru n=3, p=2 | combin.out  1 2  1 3  2 3 |

1. Se citeşte un vector cu n (n≤1000) numere întregi cu cel mult 4 cifre fiecare. Citirea se face de la tastatură. Se ştie că vectorul este sortat. Se mai citeşte o valoare întreagă x (cu cel mult 4 cifre). Să se determine, folosind un subprogram recursiv dacă x se găseşte sau nu printre elementele vectorului. Afişarea unui mesaj sugestiv se va face pe ecran.
2. Să se scrie un subprogram recursiv pentru afişarea cifrelor unui număr natural cu maxim 9 cifre (Cifrele se vor afişa de la cea mai semnificativă, la cifra unităţilor, fiecare cifră se va afişa pe câte o linie, pe ecran). În programul principal se citeşte un număr natural, căruia i se vor afişa pe ecran cifrele, folosind subprogramul creat.

Exemplu:

|  |  |
| --- | --- |
| pentru n=30127 | Se va afişa:  3  0  1  2  7 |

1. Scrieţi o funcţie recursivă care ridică un număr A la puterea N (A şi N numere întregi).Afişarea rezultatului se face pe ecran.

Exemplu

|  |  |
| --- | --- |
| pentru A= 2 N=3  pentru A=2 N=-3 | Se va afişa:  8  0.125 |

1. Să se genereze şi să se afişeze pe ecran toate numerele formate din n cifre distincte cu proprietatea că suma cifrelor este S (n şi S se citesc de la tastatură).

Ex: n=3, S=22 => 589, 598, 679, 697, 769, 796, 859, 895, 958, 967, 976, 985.

1. Să se genereze toate şirurile de lungime n, (1≤n≤10) formate doar din literele “A” şi “M”, şiruri care să nu aibă două litere “A” alăturate. Valoarea lui n se citeşte de la tastatură. Fiecare şir se va afişa pe câte un rând al ecranului, fără spaţii între ele. Exemplu: pentru n=3 şirurile generate sunt: AMA, AMM, MAM, MMA, MMM
2. În fişierul „atestat.in” se găsesc scrise pe acelaşi rând numere reale, despărţite prin câte un spaţiu. Cu aceste numere se formează o listă liniară simplu înlănţuită alocată dinamic. Tipăriţi pe ecran lista creată. Inseraţi între oricare două elemente din listă iniţială media aritmetică a vecinilor săi. Vizualizaţi din nou lista.

Exemplu:

|  |  |
| --- | --- |
| atestat.in  2 10 7 11 | Se va tipari, după inserare:  2 6 10 8.5 7 9 11 |

1. Se citesc de la tastatură pe rând numere întregi(cu cel mult 9 cifre fiecare) până la întâlnirea numărului 13 cu care se formează o listă liniară simplu înlănţuită (numărul 13 nu face parte din listă). Tipăriţi pe ecran lista creată, apoi ştergeţi prima apariţie a primului număr negativ din listă. Vizualizaţi din nou lista sau daţi mesajul „Nu există numere negative” în caz că lista nu are numere negative.
2. Să se creeze o listă liniară dublu înlănţuită cu “n” cuvinte cu cel mult 30 de litere fiecare, care se citesc de la tastatură (n dat, citit de la tastatură). Vizualizaţi lista creată, determinaţi câte cuvinte în listă sunt de lungime minimă.

Ex: n=6 şi cuvintele“notă”, “la”, “atestat”, “în”, “luna”, “aprilie” => “2”.

1. Se citesc din fişierul *„atestat.in”* numere reale cu care se formează o listă liniară dublu înlănţuită alocată dinamic. Tipăriţi lista pe ecran de la dreapta la stânga. Adăugaţi în faţa primului element suma numerelor din listă. Vizualizaţi din nou lista, dar de la stânga la dreapta.
2. Se citesc din fişierul *„atestat.in”* numere reale cu care se formează o listă circulară simplă înlănţuită, alocată dinamic. Tipăriţi lista pe ecran. Găsiţi valoarea minimă a elementelor listei, care se afişează pe ecran.
3. Din fişierul *„atestat.in”* se citeşte de pe primul rând n,(n≤1000) apoi de pe următoarele n rânduri, n numere întregi (cu cel mult 9 cifre fiecare). Creaţi o coadă alocată dinamic. Vizualizaţi coada creată. Ştergeţi din coadă 3 numere, dacă se poate, sau tipăriţi un mesaj corespunzător, dacă acest lucru nu se poate. Tipăriţi apoi din nou coada pe ecran.
4. Fişierul *„atestat.in”* conţine numere naturale cu cel mult 4 cifre, cu care se formează o listă liniară dublu înlănţuită alocată dinamic. Tipăriţi pe ecran lista. Ştergeţi prima apariţie a unui număr par din listă. Vizualizaţi din nou.
5. Se citesc din fişierul *„atestat.in”* numere reale cu care se formează o listă liniară tip stivă, alocată dinamic. Tipăriţi pe ecran stiva. Adăugaţi în stivă un nou element care va fi egal cu media aritmetică a elementelor din stivă. Vizualizaţi din nou stiva.