

Laborator 1

1. Prezentarea generală a limbajului
2. Medii de dezvoltare

Laborator 2

1. Să se scrie un program pentru calculul lui $n! = 1 * 2 * 3 * \dots * n$ pentru $0 < n < 13$, folosind o instrucțiune de ciclare.
2. Să se calculeze suma primelor n numere naturale.
3. Să se scrie un program ce calculează valoarea aproximativă a funcției e^x , utilizând următoarea formulă:

$$y = e^x = 1 + \frac{x}{1!} + \frac{x^2}{2!} + \dots + \frac{x^n}{n!}.$$

4. Să se scrie un program care citește n valori pozitive și calculează mediile

$$\text{- aritmetică : } M_a = \frac{\sum_{i=1}^n x_i}{n}$$

$$(*) \text{- geometrică : } M_g = \sqrt[n]{x_1 * x_2 * \dots * x_n}$$

$$\text{- armonică : } M_h = \frac{n}{\sum_{i=1}^n \frac{1}{x_i}}$$

5. Să se scrie un program ce citește două valori reale (a, b) și un caracter reprezentând un operator (+, -, *) și efectuează calculul indicat de acest operator cu valorile citite în variabilele a și b. Se afișează rezultatul.
6. Să se scrie un program pentru citirea și însumarea unor numere pozitive, sfârșitul introducerii datelor fiind marcat de un număr negativ.
7. Să se scrie un program pentru calculul câtului și restului împărțirii a două numere întregi, prin scăderi succesive.
8. Să se calculeze și să se afișeze cel mai mare divizor comun a două numere întregi, obținut prin împărțiri succesive (algoritmul lui Euclid).
9. Să se scrie un program ce stabilește dacă un număr n întreg este sau nu prim.
10. Să se scrie programul de calcul al combinațiilor de n luate câte m , folosind formula de recurență:

$$C_n^k = C_{n-1}^k + C_{n-1}^{k-1} \text{ pentru } k = 2, 3, \dots, n \text{ unde } C_1^0 = C_1^1 = 1.$$

Laborator 3

1. Să se scrie un program pentru extragerea rădăcinii pătrate dintr-un număr real x , printr-o metodă de aproximații succesive (Newton), cu o precizie epsilon.

Indicație:

Se repetă următoarea relație de recurență, între două aproximații succesive:

$$x_{i+1} = \frac{1}{2} \left(x_i + \frac{x}{x_i} \right), |x_{i+1} - x_i| \leq \epsilon$$

Aproximația inițială este: $x_1 = 1/2$

2. Să se scrie un program care determină toți divizorii unui număr natural.
3. Să se scrie un program care descompune în factori primi un număr natural.
4. Se citesc n elemente numere naturale. Să se determine numărul elementelor divizibile cu 3.
5. Să se scrie un program care să verifice dacă un număr natural este simetric.
6. Un număr natural este perfect dacă este egal cu suma tuturor divizorilor săi proprii + 1. Să se determine toate numerele perfecte mai mici ca n .
7. Să se scrie un program care să determine toate modurile de a scrie un număr n ca sumă de k numere naturale consecutive (dacă este posibil).

Laborator 4

1. Să se scrie un program care citește doi vectori de aceeași lungime și calculează cosinusul unghiului format de cei doi vectori. Se afișează această valoare.

Indicații:

- se citesc n și componentele celor doi vectori (A, B)

- se calculează produsul scalar P :

$$P = \overline{A} * \overline{B} = A * B * \cos(\alpha)$$

$$\cos(\alpha) = \frac{P}{|\overline{A}| * |\overline{B}|} = \frac{\sum_{i=1}^n a_i * b_i}{\sum_{i=1}^n a_i^2 * \sum_{i=1}^n b_i^2}$$

2. Să se scrie un program care ordonează un vector de n numere întregi citite de la consolă și afișează vectorul astfel ordonat.

3. Să se scrie programul de căutare a unui număr dat, într-o listă (vector) de n numere citite de la consolă.

4. Să se scrie un program ce determină elementele dintr-un șir (vector) care se află într-un interval dat (a, b), toate valorile fiind citite de la consolă.

5. Să se scrie un program ce calculează produsul a două matrici $A \in M_{n \times m}, B \in M_{m \times p}$.

6. Să se scrie un program ce calculează minimul și maximul dintr-o matrice cu n linii și m coloane utilizând funcții.

7. Să se scrie un program pentru calculul și afișarea matricei B dată de formula următoare:

$$B = I_n + A + A^2 + \dots + A^n$$

unde:

I_n este matricea unitate

A este o matrice de $n \times n$ componente ce se citește de la consolă.

Indicații:

- se citește n și matricea A
- se definește B ca fiind matricea unitate I_n
- se repetă de n ori secvența:

$$C < -I_n$$

$$C < -C * A + I_n$$

$$B < -C$$

Se vor obține, pe rând, următoarele valori:

$$A + I_n, A^2 + A + I_n, A^3 + A^2 + A + I_n, \dots$$

- se afișează B

8. Se dă o matrice de $n \times n$ componente, $n \leq 10$. Să se afișeze numărul elementelor negative, suma elementelor de pe linia pe care se găsește elementul maxim al matricei și numărul elementelor zero de pe diagonala principală.

9. Să se realizeze un program care să conțină funcții pentru:

- eliminarea unei linii dintr-o matrice
- eliminarea unei coloane dintr-o matrice

Să se utilizeze aceste funcții pentru a elimina linia și coloana pe care se află cel mai mic număr dintr-o matrice.

Laborator 5

1. Să se scrie funcții pentru concatenarea a două șiruri de caractere, aflarea lungimii unui șir de caractere, inversarea unui șir de caractere.

2. Să se scrie un program de căutare și înlocuire a unui șir dintr-un text, printr-un alt șir specificat.

Indicații:

- se citește șirul căutat (de înlocuit)
- se determină lungimea acestui șir
- se citește șirul nou (înlocuitor)
- se introduc linii de text terminate cu '!'
- cât timp primul caracter din linia de text $<> '!$

execută repetat

- citește o linie
- repetă secvența
 - determină poziția primului șir din text (linie)
 - dacă valoarea poziției este >0 se înlocuiește șirul (șterge șirul din linie și inserează șirul în poziția indicată)
- până când valoarea poziției este egală cu 0
- scrie linia

3. Să se scrie un program care numără caracterele și literele dintr-un text citit de la consolă și afișează aceste numere.

4. Se dau n numere reale ce reprezintă rădăcinile unui polinom. Să se determine coeficienții acestui polinom.

5. Se consideră n persoane situate în jurul unei mese. Începând cu persoana p se elimină una câte una fiecare a p -a persoană. Procedul se oprește atunci când a mai rămas o singură persoană. Trebuie să aflăm a câta persoană din cele inițiale rămâne.

Observație. Persoanele eliminate nu se mai numără.

6. Fiind dat un polinom

$$f = a_n * x^n + a_{n-1} * x^{n-1} + \dots + a_1 * x^1 + a_0 * x^0, \text{ să se calculeze valoarea } f(b), \text{ unde } a_i \in R, b \in R.$$

7. Fiind date două polinoame de grad n respectiv m să se calculeze produsul și suma acestora.

8. Să se calculeze câtul împărțirii unui polinom de gradul n la $(x-a)$.

9. Fiind date două polinoame de grade n și respectiv m să se calculeze câtul și restul împărțirii primului polinom la cel de-al doilea.

10. Se dă un număr format din n cifre (maxim 20). Să se determine cel mai mare număr obținut prin eliminarea a k cifre din cele n .

11. Se dă un șir de n numere întregi. Să se determine cel mai lung subșir strict crescător format din elemente situate pe poziții consecutive.

12. Un dreptunghi cu laturile de dimensiuni m și n este împărțit în pătrate de dimensiune 1×1 . Să se scrie un program care determină numărul de pătrate traversate de diagonala dreptunghiului.

Laborator 6

1. Să se calculeze cel mai mare divizor comun a două numere naturale.

Indicație:

$$cmmdc(m, n) = \begin{cases} m & , m = n \\ cmmdc(n - m, m) & , n > m \\ cmmdc(m - n, n) & , n \leq m \end{cases}$$

sau

$$cmmdc(m, n) = \begin{cases} m, & n = 0 \\ n, & m = 0 \\ cmmdc(n, m \bmod n), & m > n \\ cmmdc(m, n \bmod m), & n > m \end{cases}$$

2. Să se scrie o funcție recursivă pentru a calcula suma cifrelor unui număr natural.

Indicație:

$$S(n) = \begin{cases} 0 & , n = 0 \\ n \bmod 10 + S(n \text{ div } 10) & , \text{ altfel} \end{cases}$$

3. Calculați recursiv suma a n numere naturale.

4. Calculați recursiv C_n^k utilizând formula:

$$C_n^k = C_{n-1}^k + C_{n-1}^{k-1}$$

5. Scrieți o procedură recursivă care afișează un șir de caractere citit de la tastatură în ordine inversă.

6. Scrieți o procedură recursivă care să verifice egalitatea a două șiruri de caractere.

7. Scrieți o funcție recursivă care să determine elementul maxim al unui vector dat.

Indicație: Maximul unui șir de n elemente se definește ca fiind maximul dintre ultima valoare și maximul șirului de $n-1$ elemente rămas.

$$Max_n = Max(a_n, Max_{n-1}) \text{ unde prin } Max_n \text{ înțelegem valoarea maximă a șirului } a_1, \dots, a_n$$

sau:

$$Max(1, n) = \max(Max(1, p), Max(p+1, n)) \text{ unde } p = (1+n) \text{ div } 2.$$

8. Să se compare alfabetic (lexicografic) două șiruri de caractere cu ajutorul unei funcții recursive. Funcția trebuie să returneze:

- -1 dacă $s_1 < s_2$

- 0 dacă $s_1 = s_2$

- 1 dacă $s_1 > s_2$

9. Fie M un cuvânt și P o frază. Să se verifice dacă fraza P conține, în ordine, literele cuvântului M .

10. Se dă ecuația $x^2 - S * x + P = 0$. Să se realizeze o funcție recursivă pentru a calcula $S = x_1^n + x_2^n$, unde x_1 și x_2 sunt rădăcinile ecuației.

Indicație:

Avem formula de recurență:

$$S_n - S * S_{n-1} + p * S_{n-2} = 0, \quad S_0 = 2, \quad S_1 = S$$

Laborator 7

1. Fiind dată o mulțime cu n elemente, numărul total de partiții în k clase (k submulțimi) este dat de numerele lui Stirling de speța a doua $S(n,k)$.

Să se calculeze numărul partițiilor unei mulțimi de n elemente.

Indicație:

$$S(n,1) = S(n,n) = 1$$

$$S(n+1,k) = S(n,k-1) + k * S(n,k)$$

2. Un număr natural n, se poate descompune ca suma unică de numere naturale. De exemplu, pentru numărul 4 se scrie descompunerea $2+1+1$ (secvență descrescătoare). Să se calculeze numărul total de descompuneri ale numărului n.

Exemplu $4=3+1=2+2 \Rightarrow 2$ descompuneri.

Indicație:

Notăm cu $P(n,m)$ numărul de descompuneri ale lui n ca sumă de m numere. Acesta verifică următoarele relații:

$$P(n,1) + P(n,2) + \dots + P(n,k) = P(n+k, k)$$

$$P(n,1) = P(n,n) = 1$$

Laborator 8

Test de Laborator

Laborator 9

Implementarea exemplilor de la curs (accesul la date prin intermediul pointerilor)

Laborator 10

Operații cu liste

Laborator 11

Operații asupra fișierelor. Funcții de intrare/ieșire de nivel superior

Laborator 12

Operații asupra fișierelor. Operații la nivel de înregistrare

Laborator 13

Exemplificarea claselor de alocare

Legătura dintre pointeri și tablouri

Laborator 14

Prezentarea proiectelor de laborator