



MINISTERUL EDUCAȚIEI ȘI CERCETĂRII
ROMÂNIA

UNIVERSITATEA DE MEDICINĂ,
FARMACIE, ȘTIINȚE ȘI TEHNOLOGIE
„GEORGE EMIL PALADE”
DIN TÂRGU MUREȘ

FACULTATEA DE INGINERIE ȘI TEHNOLOGIA INFORMAȚIEI

Algoritmi fundamentali

Laboratoare

Pseudocod și Bazele logicii binare, ...

...

Dr. ing. Kiss István

istvan.kiss@umfst.ro

Cuprins

1. Bazele logicii binare
2. Operații de bază
3. Propoziții logice
4. Pseudocod. Date intrare-ieșire
5. Noțiunea de variabilă
6. Expresii
7. Atribuire
8. Structuri de algoritmi
 1. Liniară (secvențială)
 2. Alternativă
 3. Repetitivă
9. Algoritmi liniari
10. Algoritmi cu ramificații
11. Algoritmi ciclici

2. Operații de bază

- "1" – un bit
- "1001" – cuvânt (număr) binar de 4 biți
- 2 tipuri
 - $1 \&\& 0 = ?$
 - $10_{10} \& 2_{10} = ?$

Operații logice binare de bază						
Operație	Simbol logic	Simbol programare (logică)	Simbol programare (binar)	Operand 1	Operand 2	Rezultat
AND (și)	∧	&&	&	0	0	0
				0	1	0
				1	0	0
				1	1	1
OR (sau)	∨			0	0	0
				0	1	1
				1	0	1
				1	1	1
NOT (negație)	¬	!	~	-	1	0
				-	0	1

3. Propoziții logice

- Se formează pornind de la operații de bază
- Dacă P , Q sunt doi operatori logici:
- $P \vee Q$ este echivalent cu propoziția: P sau Q , având valoarea de adevăr 1 dacă măcar unul dintre elementele P sau Q este evaluat ca 1 sau 0 (fals) altfel.
- De ex.: $(P \vee Q) \wedge \neg(P \wedge Q)$ este 1 dacă și numai dacă una dintre P și Q este 1 și cealaltă 0.
- Ex.: Să spunem că **P** înseamnă "ai 14 ani sau mai mult" iar **Q** înseamnă "nu ai buletin". Neștiind nici dacă aveți 14 ani și nici dacă v-ați făcut deja buletin, propoziția logică " $R = P \wedge Q$ ", unde **R** înseamnă "ar trebui să vă faceți buletinul" este validă.

3. Propoziții logice

$a > b$ $a \leq b$

$a < b$ $a \geq b$

$a == b$ $a != b$

$(a > b) \ \&\& \ (c > a)$, atunci $(c > b)$?

3. Propoziții logice - probleme

- **Argumente:**

a) Dacă autobuzul pleacă la ora fixată și nu are întârzieri pe traseu, înseamnă că va ajunge la timp. Întrucât autobuzul nu a ajuns la timp, rezultă că el nu a plecat la ora fixată sau că a avut întârzieri pe traseu.

b) Dacă populația crește în progresie geometrică, în timp ce resursele cresc în progresie aritmetică, sărăcia generalizată este inevitabilă. Populația nu crește în progresie geometrică. Deci, sărăcia generalizată nu este inevitabilă.

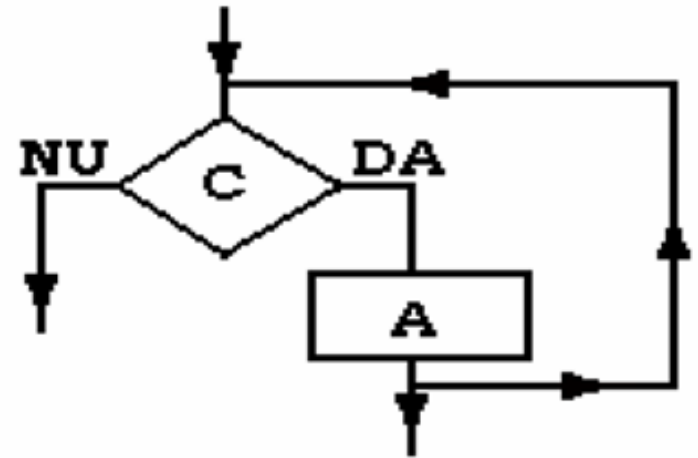
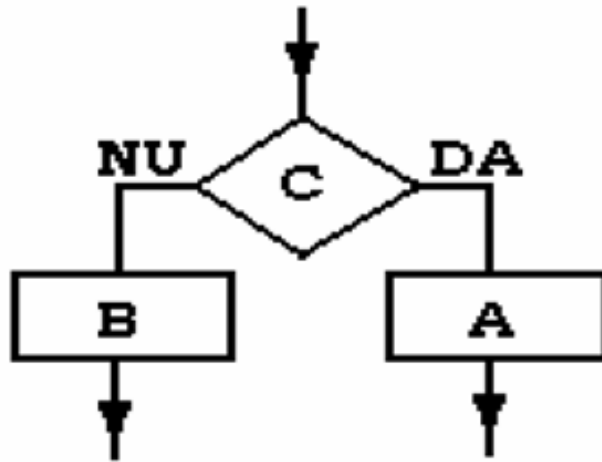
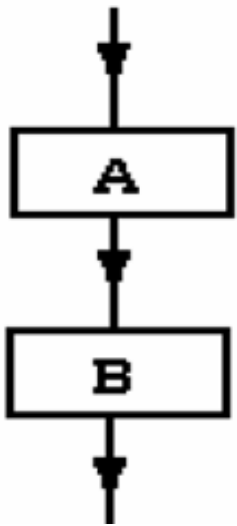
c) Dacă primarul ales este un bun gospodar sau dispune de consilieri pricepuți, atunci fondurile vor fi direcționate spre modernizarea utilitatilor publice. Cum fondurile sunt destinate modernizării utilitatilor publice, înseamnă că primarul ales este un bun gospodar sau dispune de consilieri pricepuți și onesti.

- **Cerinte:**

Identificați propozițiile componente și verificați egalitățile.

8. Structuri de algoritmi

1. Liniară (secvențială)
2. Alternativă
3. Repetitivă



10. Structură alternativă (decizia)

ALGORITMUL *ECGRDOI ESTE:* {Rezolvarea ecuației de gradul doi }

CITEȘTE a, b, c ; { a, b, c = Coeficienții ecuației }

$\text{delta} := b * b - 4 * a * c$;

DACĂ $\text{delta} < 0$ **ATUNCI** $\text{ind} := 0$; { rădăcini complexe }

$r := \text{radical din } (-\text{delta})$;

$x1 := -b / (a + a)$;

$x2 := r / (a + a)$;

ALTFEL $\text{ind} := 1$; { rădăcini reale }

$r := \text{radical din delta}$;

$x1 := (-b - r) / (a + a)$;

$x2 := (-b + r) / (a + a)$;

SFDACĂ

TIPĂREȘTE $\text{ind}, x1, x2$;

SFALGORITM

11.1. cât timp ... Exec.

- **CÂTTIMP** cond **EXECUTĂ** A **SF.CÂT**

```
întreg numar;  
citire numar;  
cat timp numar<>0 executa  
    citire numar;  
sf. cat timp
```

```
întreg numar;  
natural contor:=1;  
cat timp contor<=10 executa  
    citire numar;  
    contor:=contor+1;  
sf. cat timp
```

11.1. cât timp ... Exec.

- **CÂTTIMP** cond **EXECUTĂ** A **SF.CÂT**

ALGORITMUL Euclid **ESTE:** {A3: Cel mai mare divizor comun}

CITEȘTE $n1, n2$; {Cele două numere a căror divizor se cere}

$d := n1; i := n2$;

CÂTTIMP $i \neq 0$ **EXECUTĂ**

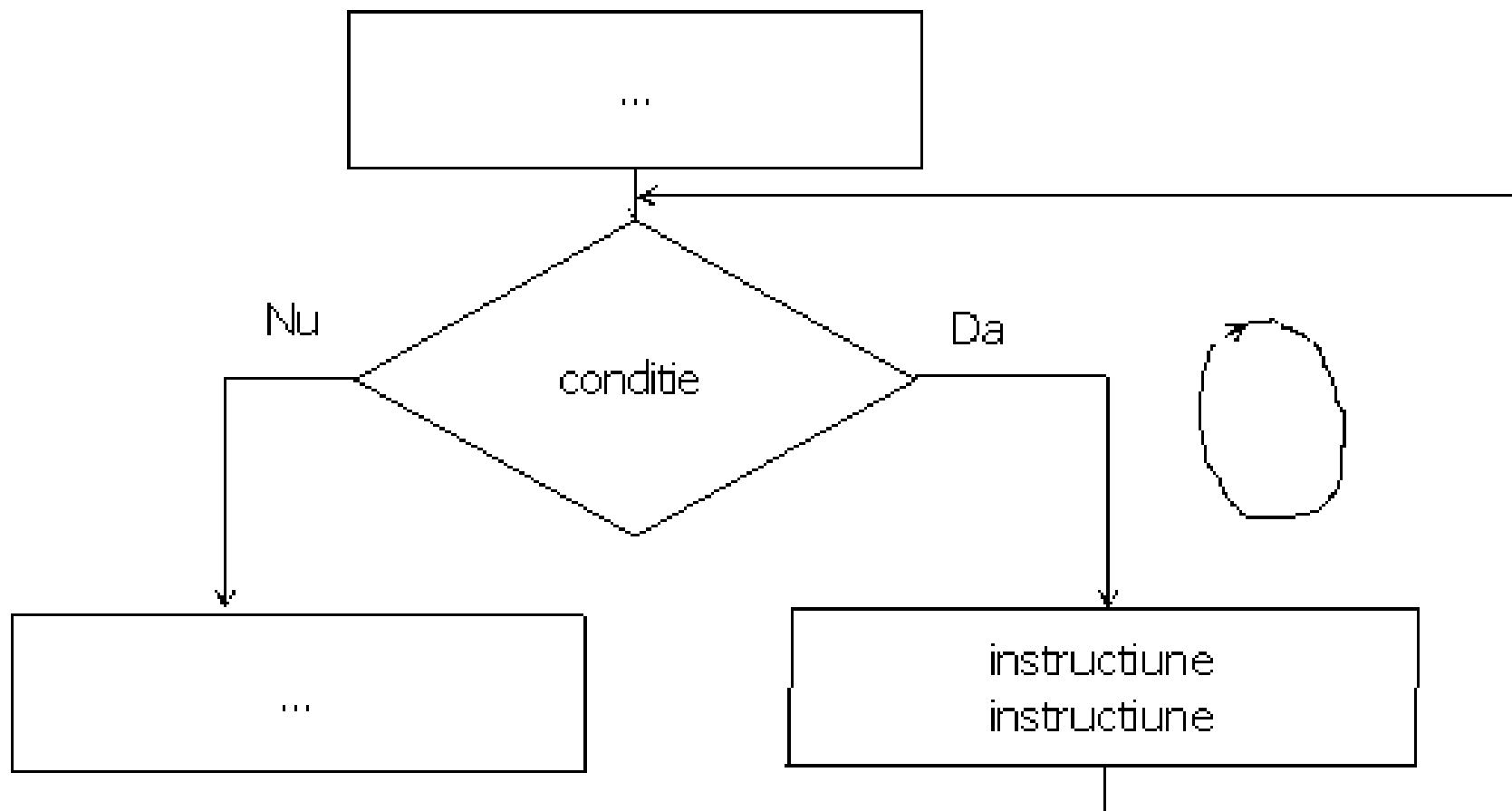
$r := d \text{ modulo } i; d := i; i := r$

SFCÂT

TIPĂREȘTE d ; { $d =$ cel mai mare divizor comun al }

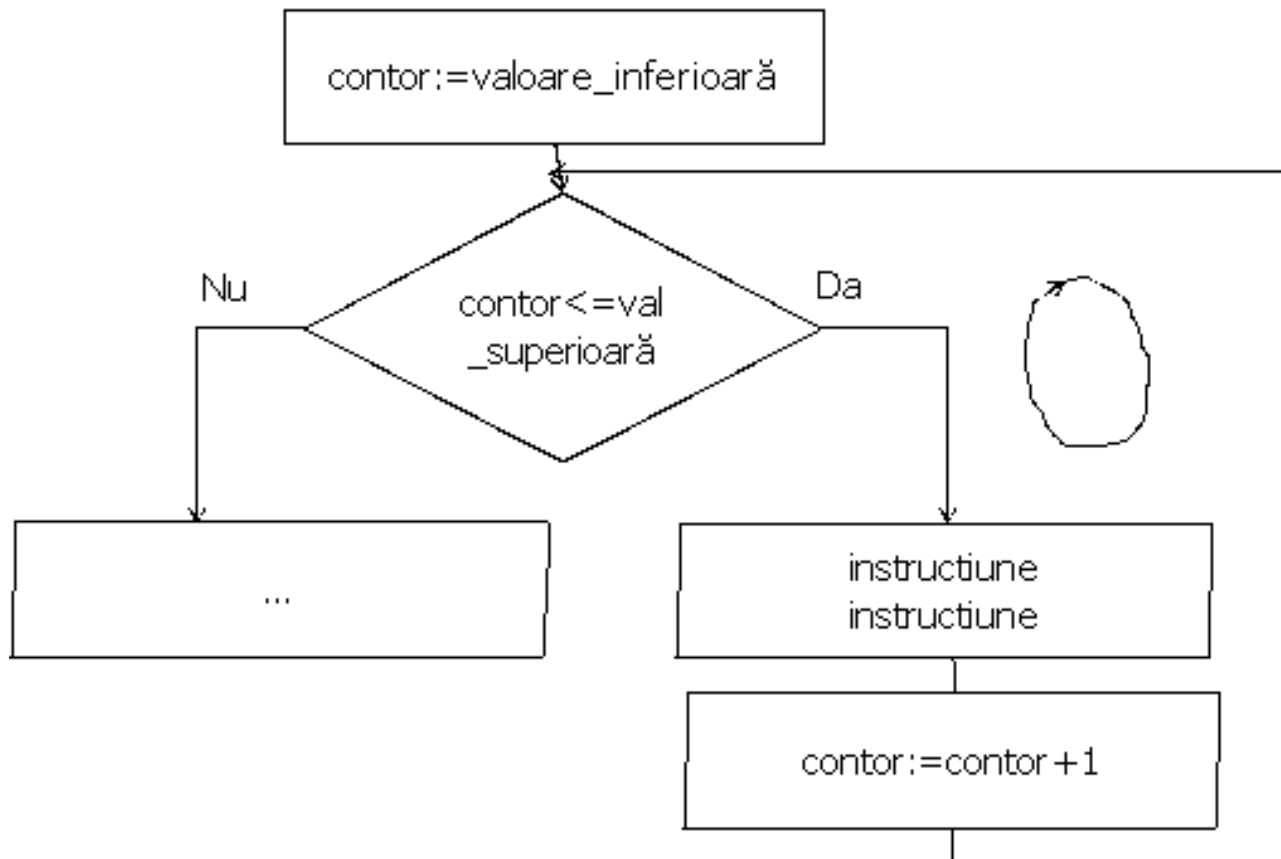
SFALGORITM { numerelor $n1$ și $n2$ }

11.1. Schemă logică cât timp



11.2. Pentru ... Exec.

- **PENTRU** $c:=l_i;l_f[;p]$ **EXECUTĂ A SF.PENTRU**



11.2. Pentru ... Exec.

- **PENTRU** $c:=l_i;l_f[;p]$ **EXECUTĂ A SF.PENTRU**

ALGORITMUL MAXMIN ESTE { Algoritmul 5: Calculul } { valorii minime și maxime }

CITEȘTE $n, (x_i, i=1, n);$

$valmin:=x_1; valmax:=x_1;$

PENTRU $i:=2, n$ **EXECUTĂ**

DACĂ $x_i < valmin$ **ATUNCI** $valmin:=x_i$ **SFDACĂ**

DACĂ $x_i > valmax$ **ATUNCI** $valmax:=x_i$ **SFDACĂ**

SFPENTRU

TIPĂREȘTE $valmin, valmax;$

SFALGORITM

11.2. Pentru ... Exec.

- ***PENTRU*** $c:=l_i;l_f[;p]$ ***EXECUTĂ A SF.PENTRU***

întreg numar;

natural contor;

pentru contor:=1..10 execută

 citire numar;

sf. pentru

11.2. Pentru ... Exec.

- **PENTRU** $c:=l_i;l_f[;p]$ **EXECUTĂ A SF.PENTRU**

Structura echivalentă este:

$c:=l_i$; $final:=l_f$;

REPETĂ

A

$c:=c+p$

PÂNĂCÂND ($c>final$ și $p>0$) sau ($c<final$ și $p<0$)

SFREP

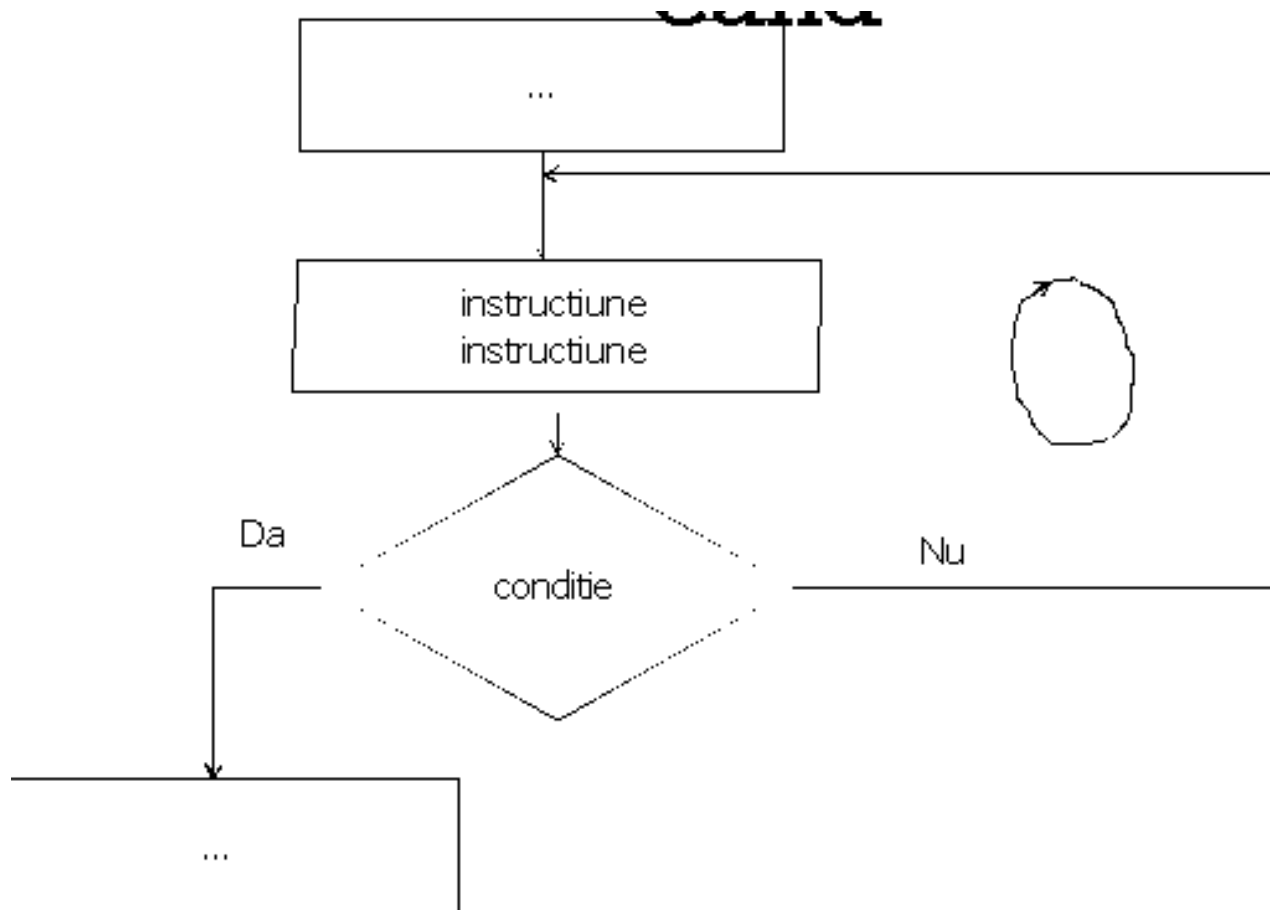
11.3. repetă ... Până când

- ***REPETĂ A PÂNĂ CÂND*** *cond SFREP*
- ***Structură echivalentă:***
- ***CÂTTIMP*** *not(cond)* ***EXECUTĂ A SFCÂT***

```
întreg numar;  
repetă  
    citire numar;  
până când numar=0;
```


11.3. repetă ... Până când

- **REPETĂ A PÂNĂ CÂND** cond **SFREP**



12. Probleme – logică binară

1. Care este operatorul logic cu care putem transforma un bit al unui număr binar în 1? Notând numărul bitului cerut cu b , cum se efectuează transformarea?
 2. Care este operatorul logic cu care putem transforma un bit al unui număr binar în 0? Notând numărul bitului cerut cu b , cum se efectuează transformarea?
 3. Care este operatorul logic cu care putem testa valoarea unui bit dintr-un număr binar? Cum se efectuează testarea?
 4. Realizați un algoritm care determină numărul de cifre de 1 din reprezentarea binară a unui număr natural nenul n .
- **Obs.: la problemele 1-2 numărul binar rezultat trebuie să fie identic cu numărul binar de la intrarea în operație, mai puțin bitul b !**

13. Probleme – alg. Elementari

Pseudocod si Schema logica

1. Maximul a două numere naturale
2. Soluția ecuației de gradul I
3. Suma numerelor naturale de la 1 la 10
4. Afișarea factorialului unui număr dat
5. Algoritm pentru interschimbare a două numere
6. Algoritm pentru interschimbare a două numere fără variabilă intermediară
7. Determinarea maximului (minimului) dintr-un șir de numere introduse de la tastatură
8. Determinarea maximului a n numere (introduse de la tastatură până introducerea valorii 0)
9. Extragerea cifrelor unui număr
10. Compunerea unui număr din cifrele sale
11. Determinarea inversului unui număr
12. Suma cifrelor unui număr natural
13. Algoritmul lui Euclid (prin scădere repetată)
14. Algoritm pentru determinare cel mai mare divizor comun pentru 3 numere întregi
15. Testare număr prim
16. Să se determine primele n numere prime
17. Conversii între sisteme de numerotație (din baza 10 într-o baza b)

14. Probleme – alg. Elementari

Pseudocod sau schema logica

B1. Se citește numărul natural n . Verificați dacă numărul n este *palindrom*. Un *palindrom* este un număr natural egal cu inversatul său. Programul va afișa mesajul PALINDROM sau NEPALINDROM, corespunzător. **Exemplu:** Dacă $n=123321 \Rightarrow$ PALINDROM, iar pentru $n=12605 \Rightarrow$ NEPALINDROM

B2. Se citesc două numere a și b , cu cel mult 4 cifre fiecare. Determinați numărul obținut prin concatenarea cifrelor lui a cu cifrele lui b . Exemplu: Pentru $a=1204$ și $b=3027$ se va afișa 12043027.

B3. Se citesc două numere a și b , cu $0 \leq a, b \leq 100.000$. Notăm cu ab numărul obținut prin concatenarea cifrelor numărului a cu b , și cu ba numărul obținut prin concatenarea cifrelor numărului b cu a . Stabiliți care dintre valorile ab sau ba este mai mare și afișați valoarea maximă. **Exemplu:** $a=208$ și $b=375$, se va afișa 375208.

B4. Se citește numărul natural n , cu cel mult 8 cifre. Afișați pe două linii:

- numărul format din cifrele impare ale numărului n , sau 0 dacă n nu are cifre impare

- numărul format din cifrele pare ale numărului n , sau 0 dacă n nu are cifre pare

Exemplu: Pentru $n=1234567$ se va scrie

1357

246

B5. Se citește un număr n și o cifră c , cu $0 \leq n \leq 109$. Eliminați toate aparițiile cifrei c din scrierea numărului n . **Exemplu:** Dacă $n=123527826$ și $c=2$ se va afișa 135786.

B6. Se citește numărul natural n . Determinați numărul format prin complementarea fiecărei cifre c față de 9, adică fiecare cifră c se va înlocui cu $9-c$. **Exemplu:** Dacă $n=20743$ se va afișa 79256.

B7. Se citește numărul natural n . Determinați numărul format prin complementarea fiecărei cifre c față de cifra maximă a numărului n , adică fiecare cifră c se va înlocui cu $c_{\max}-c$. **Exemplu:** Dacă $n=20743$ se va afișa 57034.

Indicații de rezolvare:

B1. Determinăm în variabila m inversatul variabilei n (clona variabilei n) și comparăm cele două valori.

B2. Obținem în $invb$ inversatul numărului b (clona numărului b). Aplicăm algoritmul de construire a numărului ab din cifrele sale, folosind valoarea inițială aa (clona numărului a) și cu cifrele numărului $invb$.

B3. Aplicăm algoritmul de la problema anterioară pentru a obține numerele ab și ba , apoi comparăm valorile obținute.

B4. Formăm două numere din cifrele numărului n , unul din cifrele pare pn și unul din cifrele impare in , apoi construim inversatul fiecărui număr și le afișăm.

B5. Construim un număr nou din cifrele numărului nn (clona numărului n) diferite de cifra c , apoi afișăm valoarea obținută inversată

B6. Construim un număr nou cu cifrele numărului nn (clona variabilei n) complementate, apoi afișăm inversatul numărului obținut.

B7. Determinăm cifra maximă c_{\max} a numărului nn (clona variabilei n) și aplicăm algoritmul de complementare față de c_{\max} .

15. Probleme – funcție matematică

1. Să se rezolve printr-un algoritm descris prin **schemă logică și pseudocod**.

$$\text{Să se afle } G = \begin{cases} x^3 - 1, & \text{dacă } x \geq 1 \\ \frac{x}{a} - 5, & \text{dacă } x < 1 \end{cases}, \text{ unde } a, x \in \mathfrak{R}.$$

16. Probleme – AppInventor, cele fara SMS in Thunkable

Să se proiecteze (descriere algoritm, schemă, pseudocod, variabile...) aplicațiile de mai jos folosind platforma AppInventor:

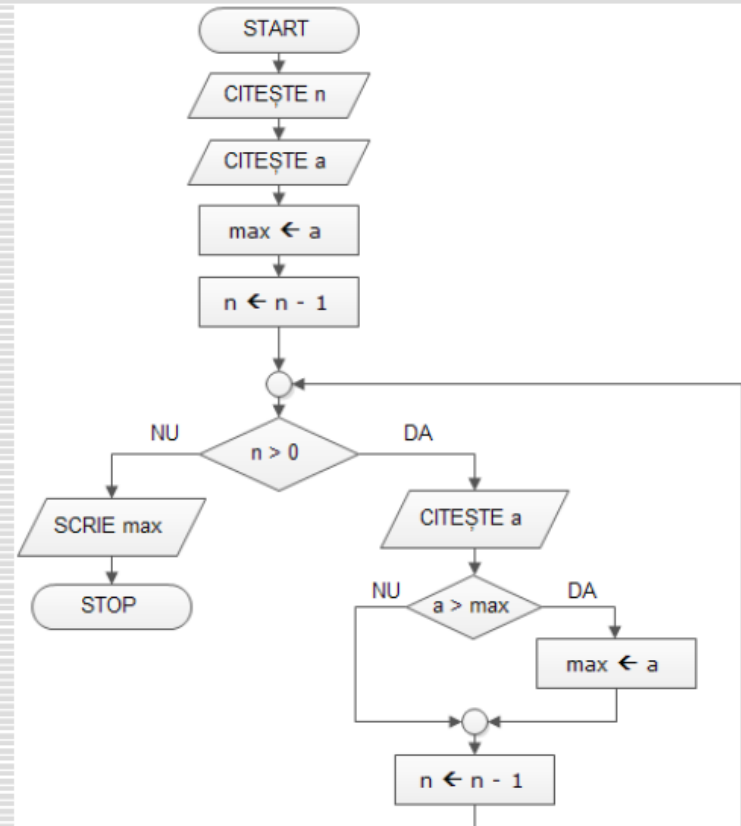
1. Două aplicații care folosește structura alternativă ("dacă").
2. O aplicație care recepționează un SMS și răspunde cu un alt SMS la numărul expeditorului.
3. Aplicație care numără cu un pas și interval de timp specificat.
4. Aplicație care primește ca și comandă prin sms un număr natural și generează un număr de sunete (beep) egale cu numărul recepționat. Între sunete trebuie să introducem o așteptare de 500ms.
5. Aplicație care primește ca și comandă prin SMS un număr natural și pornește vibrații pentru un număr de milisecunde egal cu numărul recepționat prin SMS.
6. Aplicație cu două câmpuri text și trei butoane, astfel: primul câmp text se folosește pentru date de intrare și al doile pentru afișarea rezultatelor; la apăsarea primului buton se afișează minimul și maximul numerelor înșiruite în primul câmp (separate cu spațiu), la apăsarea butonului 2 se afișează suma numerelor și la apăsarea butonului 3 se afișează produsul numerelor introduse.

Exemplu rezolvare

- Determinarea maximului a n numere (introduse de la tastatură până introducerea valorii 0)

□ Algoritmul de determinare a maximului a n numere are în pseudocod forma de mai jos:

```
citește n
citește a
max ← a
n ← n - 1
cât timp n > 0
{
    citește a
    dacă a > max
    {
        max ← a
    }
    n ← n - 1
}
scrie max
```



17. Recursivitate - probleme

1.

Functie fib(intreg n) este:

daca $n \leq 2$ **atunci**

fib:=1;

altfel

fib:=fib(n-1)+fib(n-2);

sf.daca

Sf.fibo

Functie iter_fibo(intreg n) este:

daca $n \leq 2$ **atunci** iter_fibo:=1;

sf.daca

intreg a:=1, b:=1, suma;

pentru i:=1;n-2 **executa**

suma:=a+b;

b:=a;

a:=suma;

sf.pentru

iter_fibo:=a;

Sf.functie

Sa se exemplifice executia pas cu pas
(urmarirea variatiei variabilelor intr-un
tabel) in cazul algoritmilor de mai sus.

17. Recursivitate - probleme

2. Ce înțelegeți prin recursivitate directă?

3. Găsiți greșala în funcția de mai jos.

```
int suma(int i)
{if (i==0)
return 0;
else
if (a[i]%2==0)
return suma(i+1)+a[i];
else return suma(i-1);
}
```

4. Se consideră funcția f definită recursiv. Ce va returna $f(5)$?

```
int f(int i)
{if (i==0) return 0;
if(i%2==0 return f(i-1)+i;
else return f(i-1)-i;
}
```

a.) -4

b.) -3

c.) 0

d.) 3

5. Scrieți o funcție recursivă, care calculează suma elementelor pozitive dintr-un vector și produsul elementelor negative.

17. Recursivitate - probleme

6.) Ce trebuie în mod obligatoriu să conțină o funcție recursivă pentru a nu se reapela la nesfârșit.

7.) Ce înțelegeți prin recursivitate indirectă?

8.) Ce valori vor conține variabilele x și y pentru valorile inițiale a=75, b=12, x=0, y=0, în urma executării funcției f?

```
void f(long a, long b, long &x, long &y)
```

```
{ if (y>=b)
```

```
{ y=a-b;
```

```
x++;
```

```
f((a-b), b, x, y);
```

```
}
```

9.) Se consideră funcția f definită recursiv. Ce va returna f(6)?

```
int f(int x)
```

```
{ if (x<=1) return x;
```

```
else return f(x-2)+x;
```

```
}
```

a.) 6

b.) 12

c.) 8

d.) 15

10.) Scrieți o funcție recursivă care determină maximul elementelor unui șir de numere întregi.

17. Recursivitate - probleme

11. Se citește un șir de cuvinte până la citirea caracterului “*”. Să se afișeze cuvintele în ordinea inversă citirii lor.
12. Scrieți o funcție recursivă care determină maximul elementelor negative ale unui șir de numere întregi.

Pentru puncte bonus:

13. Inversare numar
14. Suma cifrelor, suma unui sir
15. Cifra maxima
16. Conversie $b \rightarrow 10$, $10 \rightarrow b$
17. Testare anagrame – daca literele din x exista si in y ?

17. Recursivitate – probleme (si iterativ)

- **Pentru avansati:**

- Functia recursiva Ackerman
- Generarea permutarilor unei multimi
- Problema reginelor
- Problema rucsacului
- Problema prajiturilor

17. Recursivitate – probleme (si iterativ)

- Pentru avansati si nu numai:

1*. Scrieți o funcție găseste_cifra care returnează valoarea cifrei aflate pe poziția k în cadrul numărului n, începând de la dreapta (n și k vor fi argumentele funcției).

1. Scrieți o funcție recursivă care are ca parametru un număr întreg $n < 30000$ și returnează 1 dacă n este număr prim și 0 dacă n nu este prim.

2. Se citesc de la tastatură două numere naturale a și b. Să se scrie o funcție recursivă care calculează câtul și restul împărțirii întregi a lui a la b, prin scăderi repetate.

3. Scrieți o funcție recursivă care primește ca parametru un număr întreg n și afișează toți divizorii proprii ai numărului.

4. Scrieți o funcție recursivă, care să calculeze suma S, pentru n număr natural, citit de la tastatură.

$$S = 1 + 1/2 + 1/3 + \dots + 1/n;$$

5. Scrieți o funcție recursivă pentru calculul produsului elementelor unui tablou unidimensional cu elemente reale.

6. Să se numere de câte ori apare un număr întreg x într-un șir de n numere întregi.

17. Recursivitate – probleme (si iterativ)

- Pentru avansati si nu numai :

7. Să se calculeze combinaări de n elemente luate câte k unde n și k sunt două numere întregi citite de la tastatură.

8. Să se calculeze $\text{cmmdc}(a,b)$ prin scăderi repetate.

a dacă $a=b$

$\text{cmmdc}(a,b) = \text{cmmdc}(a-b,b)$ dacă $a > b$

$\text{cmmdc}(a,b-a)$ dacă $b > a$

9. Să se citească de la tastatură numere întregi până la întâlnirea lui 0. Să se scrie o funcție recursivă care să primească un parametru și să returneze numărul de numere negative citite. Numerele negative se vor trece într-un tablou declarat global.

10. Scrieți o funcție recursivă care determină elementul minim dintr-un vector cu n numere reale citite de la tastatură.

11. Fiind dat un polinom de grad n , cu o singură nedeterminată, dat prin coeficienții săi, reținuți într-un vector. Să se calculeze valoarea polinomului într-un punct x_0 dat, folosind o funcție recursivă.

12. Se citește de la tastatură un cuvânt și o literă. Scrieți o funcție recursivă care numără de câte ori apare litera respectivă în cuvântul citit.

17. Recursivitate – probleme (si iterativ)

- Pentru avansati **si nu numai** :

13. Se citește de la tastatură un cuvânt și o literă. Să se elimine toate aparițiile literei respective în cuvântul dat.

14. Se citește de la tastatură un șir de cuvinte, apoi se citește un cuvânt. Să se numere de câte ori apare cuvântul respectiv în șirul dat.

15. Se citește de la tastatură un șir de cuvinte, apoi se citește un cuvânt. Să se elimine din propoziție toate aparițiile cuvântului dat.

16. Scrieți un program care citește un număr natural n ($1 < n < 16$) și scrie pe prima linie a fișierului SIR.OUT un șir ce conține numerele între 1 și 2 la puterea n , șir creat astfel:

- se pornește cu șirul inițial format din 1 și 2;
- exact la mijlocul acestuia se inserează secvența formată din 3 și 4;
- exact la mijlocul șirului astfel format se inserează secvența formată din numerele 5,6,7 și 8, obținând astfel șirul 1 3 5 6 7 8 4 2 etc.

Exemplu : Pentru $n=4$ se obține: 1 3 5 6 9 10 11 12 13 14 15 16 7 8 4 2

18. Sortare – probleme vectori

1. Maximul unui vector
2. Inversul unui vector
3. Suma elementelor
4. Intersectia, reuniunea a doua multimi(vectori)
5. Prelucrarea cifrelor elementelor unui vector
6. Frecventa unui element dintr-un vector sau matrice
7. Câte numere din şir (vector) sunt divizibile cu ultimul element al şirului.
8. Gigel s-a documentat şi a făcut o listă cu preţurile ultimelor n smartphone-uri apărute şi altă listă cu preţurile celor mai vandute m gadgeturi. Ajutaţi-l să determine cel mai scump smartphone şi cel mai ieftin gadget. Dacă sunt mai multe, sa afişeze toate poziţiile din listele corespunzătoare.
9. Date fiind un şir cu n numere naturale $n < 1000$ si un k , astfel $0 < k < n$. Afisati produsul ultimelor k numere diferite de 0, daca exista. Daca nu exista k astfel de numere in sir, afisati mesajul "imposibil".
10. Transpusa matricei
11. Elementul maxim si minim dintr-o matrice
12. Sa se stocheze intr-un vector elementele maxime din liniile matricei

18. Sortare – probleme

A. Sa se studieze fiecare algoritm discutat la ore.

- Mecanismul de functionare
- Studiul prin filmulete
- Schitarea functionarii prin numere/siruri concrete (executie pas cu pas)
- **Optional:** Compararea complexitatii

• **In scratch:** ilustrati o metoda de sortare!!!

18. Sortare – probleme vectori

- Pentru avansati

13. Duduța are 2 albume cu poze pe Facebook. In primul album are n poze, în cel de al doilea are m poze. Fiecare poză are un anumit numar de like-uri. Calculați câte poze din primul album au mai puține like-uri decât fiecare poză din cel de-al doilea și care sunt acestea. Pozele sunt numerotate cu numere de la 1 la n respectiv de la 1 la m ($1 < n, m < 50$) .

Exemplu:

```
primul album:      5                //5 poze in primul album (de la 1 la 5)
                   3 6 18 5 9        //numarul de like-uri pentru fiecare
al doilea album:   7                //5 poze in primul album (de la 1 la 7)
                   13 9 10 22 8 7 12 // numarul de like-uri pentru fiecare
```

```
Se va afișa:      3
                  1  2  4
```

adica: sunt 3 poze in primul album care au mai putine like-uri decât orice poză din al doilea iar pozele căutate sunt prima (3 like-uri), a doua (6 like-uri) și a patra (5 like-uri).

18. Sortare – probleme vectori

• Pentru avansati

14. Se citește de la tastatură n , un număr natural nenul, $n < 100$, apoi se citesc cele n elemente ale vectorului a . Să se afișeze pe ecran:
- Pozițiile din vector pe care se află numere divizibile cu 5
 - Elementele vectorului care au ultimele două cifre egale, în ordinea inversă apariției în vector
 - Numărul de valori egale cu 0 din vectorul dat
 - Suma numerelor de exact 3 cifre din vectorul dat
 - Numerele care sunt egale cu media aritmetică a elementelor din vector
 - Perechile de numere alăturate din vector care au aceeași paritate (sunt ambele pare sau ambele impare)
15. Se citește de la tastatură n , un număr natural nenul, $n < 100$, apoi se citesc cele n elemente ale vectorului a . Să se afișeze pe ecran:
- produsul numerelor mai mari decât 100
 - numărul de valori negative
 - numerele care sunt egale cu media aritmetică a vecinilor lor (numărul din stânga + numărul din dreapta lui) / 2
 - mesajul DA, dacă toate valorile sunt în ordine crescătoare și NU altfel

18. Sortare – probleme vectori

- Pentru avansati

16. Scrieți un program **C++** care citește de la tastatură două numere naturale nenule **n** și **m** ($2 \leq m \leq 10$, $2 \leq n \leq 10$) și care construiește în memorie și apoi afișează o matrice **A** cu **n** linii (numerotate de la 1 la **n**) și **m** coloane (numerotate de la 1 la **m**) cu proprietatea că fiecare element **A** *ij* memorează cea mai mică dintre valorile indicilor **i** și **j** ($1 \leq i \leq n$, $1 \leq j \leq m$).

Matricea se va afișa pe ecran, câte o linie a matricei pe câte o linie a ecranului, elementele fiecărei linii fiind separate prin câte un spațiu.

Exemplu: pentru **n=4** și **m=5** se va afișa matricea alăturată.

```
1 1 1 1 1
1 2 2 2 2
1 2 3 3 3
1 2 3 4 4
```

19. Algoritmi cautare - probleme

1. Sa se studieze fiecare algoritm discutat la ore.

- Mecanismul de functionare
- Studiul prin filmulete
- Schitarea functionarii prin numere concrete (Exec. pas cu pas)
- Compararea complexitatii

2. Se se implementeze subalgoritmul de agaugare si stergere element dintr-un tablou.

- Parametri formali sunt: vectorul, indexul, cheia