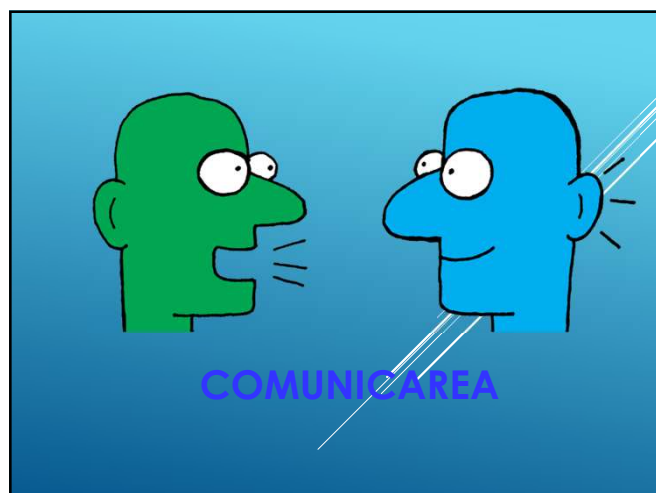


Noțiuni fundamentale utile în
programarea calculatoarelor



- După modul de abordare a rezolvării problemelor cu calculatorul limbajele sunt:
- procedurale - atunci când rezolvarea problemei urmează anumite etape și utilizează structuri fundamentale (Pascal, C, etc.)
 - neprocedurale - ele se bazează pe reguli și sunt mai apropiate de limbajul și modul de raționare natural (limbajele pentru inteligență artificială: Prolog, Lisp).
- ## LIMBAJE DE PROGRAMARE

- ## LIMBAJE DE PROGRAMARE
- orientate pe obiecte (C++, Java)
 - pentru aplicații Web (PHP, Pearl)
 - pentru aplicații distribuite sau mobile
 - pentru aplicații de conducere a roboților și mașinilor unelte

- ## ETAPE DE BAZĂ CARE TREBUIE URMATE PENTRU REZOLVAREA UNEI PROBLEME PE CALCULATOR
- **analiza problemei** - se stabilește exact CE subprobleme trebuie să rezolve programul
 - **programarea** - reprezentarea problemelor într-un mod adecvat pentru rezolvarea asistată de calculator
 - **implementarea** - scrierea programului care rezolvă problema într-un anumit limbaj de programare
 - **evaluarea** – la dezvoltator și la client
-

Muhammad ibn Musa Al-Khwarizmi (1780-1850)

- Islamic mathematician who wrote on Hindu-Arabic numerals
- Among the first to use zero as a place holder in positional base notation
- "Algorithm" is derived from his name
- His algebra treatise *Kitab al-jabr wal-muqabala* gives us the word algebra and can be considered as the first book to be written on algebra

<http://paperkit.net/education/algorithm/index.php?title=al-khwarizmi&id=77709-3025-4443-3023-177709&id=7036>

ALGORITM

matematică, inginerie, psihologie, management, lingvistică, etc.

Programare – principii de bază ale instrumentelor de programare utilizate în prezent, evoluție și probleme

Ansamblu de etape prin care se definește modul în care poate fi dusă la îndeplinire o anumită sarcină.

Un set ordonat de pași executabili, descriși fără echivoc, care definesc un proces finit.

7

Download from Dreamstime.com

4178517

Victor88 | Dreamstime.com

8

Marchează începutul / sfârșitul unui algoritm reprezentat în schemă logică

Se utilizează pentru implementarea operațiilor de intrare / ieșire

Se utilizează pentru scrierea condițiilor în cazul structurilor alternativă sau repetitivă

Se utilizează pentru scrierea operațiilor de atribuire (atunci când o variabilă primește o valoare)

Se utilizează pentru conectarea liniilor care leagă diferitele figuri geometrice și marchează un punct de intersecție.

Legăturile dintre blocurile schemei logice se realizează numai prin intermediul săgeților, pentru a stabili sensul de parcurgere al algoritmului.

9

Algoritmul lui Euclid

- se bazează pe ideea că cel mai mare divizor a două numere divide și restul împărțirii acestora, conform teoremei împărțirii cu rest
- se împarte deîmpărțitul la rest până când împărțitorul este 0, apoi se returnează deîmpărțitul.

Ne permite să calculăm cel mai mare divizor comun, fără să știm care sunt toți divizorii

- ca și cum ne-ar spune direct care este orașul cu cel mai mulți locuitori din Europa fără să le analizeze pe fiecare pe rând.

10

Algoritmul lui Euclid

```

int AlgEuclid(int m, int n) {
    int rest;

    rest = m%n;
    while (rest != 0) {
        m = n;
        n = rest;
        rest = m%n;
    }
    return n;
}

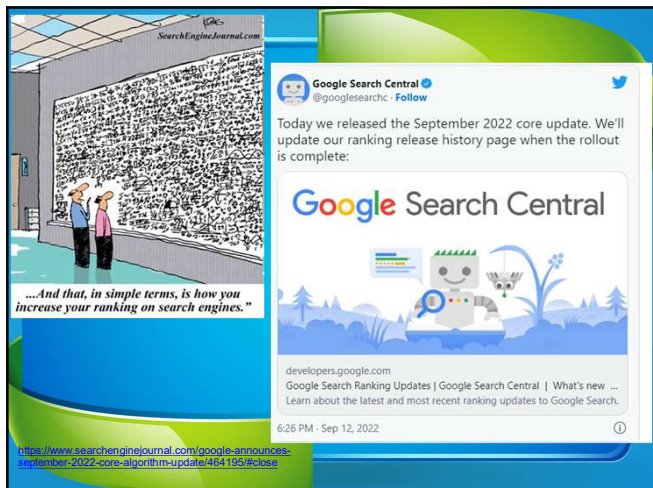
```

euclid.c

11

Algoritmul de supraveghere și screening într-o clinică de pediatrie

12



13

Algoritmi

- rezumat -

- Primește și transmite informație – date de intrare/input, date de ieșire/output
- Rezolvă o problemă

• Sunt în formă **conceptuală** - trebuie să fie reprezentate într-o formă în care pot să fie comunicate unui calculator – prin setul de instrucțiuni (gramatică și limbaj)

- limbaj de programare
- paradigme de programare

Reprezentarea algoritmilor în calculator -> program -> software

14

<https://www.youtube.com/watch?v=Pb9UNhfy1U4>,
How Often Do Software Engineers Write Algorithms?
(12m47) , aprilie 2020

Are algorithms necessary for programming? Do Software Engineers write algorithms as part of their daily job? If yes, how often? In this video, Ex-Google Software Engineer Clement Mihailescu shares three examples of algorithm-style work he's done as a software developer.

<https://tekkieuni.com/blog/algorithms-for-kids>

15

Processing Pipeline

```

byte audio [] = out.toByteArray()
int totalSize = audio.length
int sampledChunkSize =
    totalSize / chunkSize;
ComplexMatrix result =
    new ComplexMatrix[sampledChunkSize][];

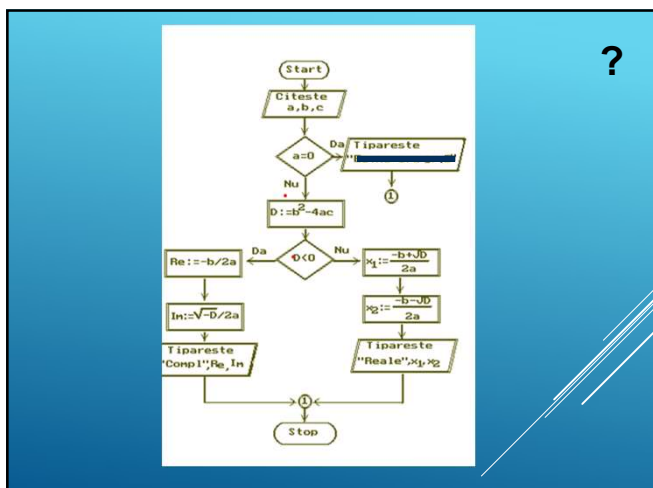
for (int i = 0; i < sampledChunkSize; i++) {
    ComplexArray chunk = new ComplexArray(chunkSize);
    for (int j = 0; j < chunkSize; j++) {
        chunk[j] = new ComplexArray[j] =
            new ComplexArray(audio[(j*chunkSize)+i], 0);
    }
    result[i] = FFT.fft(chunk);
}

```

?

<http://www.tutorial.com/algorithms/shazam-it-music-processing-fingerprinting-and-recognition>

16



17

Pentru IPC3

Realizați schema logică și apoi scrieți programul în limbaj C pentru rezolvarea următoarelor probleme:

Pentru un vector cu k elemente numere întregi citite de la tastatură, calculați suma elementelor vectorului care au valori mai mari decât valoarea minimă a elementelor din tablou pentru opțiunea 1 introdusă de la tastatură, iar pentru opțiunea 2, calculați suma elementelor vectorului care au valori mai mici decât valoarea maximă a elementelor din tablou.

18