PARTEA 1

Perfect 👍
Începem **PARTEA 1 – Lecțiile 1–6**, **pentru ELEVI**, cu **explicații simple + exerciții REZOLVATE**.
Poți copia direct în Word / Google Docs.

📘 CURS COMPLET – INFORMATICĂ CLASA A VII-A

PARTEA 1 – Lecțiile 1–6

🔹 Lecția 1 – Ce este programarea, C++ și Code::Blocks

Ce este programarea?

Programarea înseamnă să îi spui calculatorului **ce să facă**, pas cu pas, folosind un limbaj special.
Calculatorul **nu gândește**, el execută exact instrucțiunile primite.

Ce este C++?

C++ este un limbaj de programare:

- foarte folosit în școli
- folosit la concursuri

- bun pentru a învăța **logica algoritmilor**

Ce este Code::Blocks?

Code::Blocks este un program (IDE) în care:

- scriem cod C++
- compilăm programul
- rulăm programul și vedem rezultatul

Crearea unui proiect nou

```mathematica
File → New → Project
Console Application
C++
Next → Finish
```

Copiază codul

🔹 Lecția 2 – Structura unui program C++

Orice program C++ are această structură:

```cpp
#include <iostream>
using namespace std;

int main() {
    // aici scriem instructiunile
```

```
    return 0;
}
```

Explicații:

Copiază codul

- `#include <iostream>` – permite citirea și afișarea
- `using namespace std;` – nu mai scriem `std::`
- `int main()` – de aici începe programul
- `{ }` – corpul programului
- `return 0;` – programul s-a terminat corect

Exemplu – primul program

```cpp
#include <iostream>
using namespace std;

int main() {
    cout << "Salut!";
    return 0;
}
```

✅ Exercițiu rezolvat

Copiază codul

Afișează mesajul „Invat C++”.

```cpp
```

```cpp
#include <iostream>
using namespace std;

int main() {
    cout << "Invat C++";
    return 0;
}
```

Copiază codul

🔹 Lecția 3 – Variabile și tipuri de date

Ce este o variabilă?

O variabilă este o **cutie** în care memorăm o valoare.

Tipuri de date importante

Tip	Ce stochează
int	numere întregi
double	numere reale
char	un caracter
bool	adevărat / fals

Exemplu

cpp

```cpp
int a = 10;
double b = 2.5;
char c = 'A';
bool ok = true;
```

Operatorii aritmetici

Copiază codul

- `+` adunare
- `-` scădere
- `*` înmulțire
- `/` împărțire
- `%` rest (doar la int)

Exemplu

cpp

```cpp
int x = 17, y = 5;
cout << x + y << "\n";
cout << x - y << "\n";
cout << x * y << "\n";
cout << x / y << "\n";
cout << x % y << "\n";
```

✅ Exercițiu rezolvat

Copiază codul

Citește două numere și afișează suma lor.

cpp

```cpp
#include <iostream>
using namespace std;

int main() {
    int a, b;
    cin >> a >> b;
    cout << a + b;
    return 0;
}
```

Copiază codul

🔹 Lecția 4 – Citire și afișare (cin / cout)

Citire

cpp

```cpp
cin >> x;
```

Copiază codul

Afișare

cpp

```cpp
cout << x;
```

Copiază codul

Citire mai multe valori

cpp

```
cin >> a >> b >> c;
```

Copiază codul

Exemplu complet

```cpp
int a, b;
cin >> a >> b;
cout << "Suma este " << a + b;
```

Copiază codul

✅ Exercițiu rezolvat

Citește un număr și afișează dublul lui.

```cpp
#include <iostream>
using namespace std;

int main() {
    int x;
    cin >> x;
    cout << 2 * x;
    return 0;
}
```

Copiază codul

🔹 Lecția 5 – Instrucțiunea IF (decizii)

Operatorii de comparație

- `==` egal
- `!=` diferit
- `<` mai mic
- `>` mai mare
- `<=` mai mic sau egal
- `>=` mai mare sau egal

IF – ELSE

```cpp
if (conditie) {
    // se executa daca e adevarat
} else {
    // se executa daca e fals
}
```

Exemplu – par sau impar

Copiază codul

```cpp
int x;
cin >> x;

if (x % 2 == 0)
    cout << "Par";
```

```
else
    cout << "Impar";
```

Copiază codul

✅ Exercițiu rezolvat

Citește un număr și spune dacă este pozitiv, negativ sau zero.

cpp

```cpp
#include <iostream>
using namespace std;

int main() {
    int x;
    cin >> x;

    if (x > 0)
        cout << "Pozitiv";
    else if (x < 0)
        cout << "Negativ";
    else
        cout << "Zero";

    return 0;
}
```

🔹 Lecția 6 – Repetiții: FOR și WHILE

Copiază codul

FOR – știm câte repetări

```cpp
for (initializare; conditie; pas) {
    // cod
}
```

Copiază codul

Exemplu – afișează 1..n

```cpp
int n;
cin >> n;

for (int i = 1; i <= n; i++)
    cout << i << " ";
```

Copiază codul

WHILE – nu știm exact câte repetări

```cpp
while (conditie) {
    // cod
}
```

Copiază codul

Exemplu – număr de cifre

```cpp
```

```cpp
int n, cnt = 0;
cin >> n;

if (n == 0) cnt = 1;
while (n != 0) {
    cnt++;
    n /= 10;
}
cout << cnt;
```

Copiază codul

✅ Exercițiu rezolvat

Calculează suma numerelor de la 1 la n.

cpp

```cpp
#include <iostream>
using namespace std;

int main() {
    int n, s = 0;
    cin >> n;

    for (int i = 1; i <= n; i++)
        s += i;

    cout << s;
    return 0;
}
```

Copiază codul

✅ FINAL PARTEA 1

- ✔ Bazele C++
- ✔ Variabile
- ✔ Citire / afișare
- ✔ IF
- ✔ FOR / WHILE
- ✔ Exerciții rezolvate

👉 **Urmează: PARTEA 2 – Lecțiile 7–12**
(numere prime, cmmdc, vectori, sume parțiale, căutare binară, funcții)

Spune doar:
„PARTEA 2” 🚀

PARTEA 2

Excelent 👍
Continuăm cu **PARTEA 2 – Lecțiile 7–12**, în același stil: **pentru elevi + explicații simple + exerciții REZOLVATE**.

📘 CURS COMPLET – INFORMATICĂ CLASA A VII-A

PARTEA 2 – Lecțiile 7–12

🔹 Lecția 7 – Divizibilitate, numere prime, CMMDC (recapitulare clasa a VI-a)

Verificare număr prim

Un număr este **prim** dacă are **exact doi divizori**: 1 și el însuși.

Algoritm

- dacă `n < 2` → NU este prim
- verificăm divizorii de la `2` la `√n`

Exemplu

cpp Copiază codul

```cpp
#include <iostream>
using namespace std;

int main() {
    int n;
    cin >> n;

    if (n < 2) {
        cout << "NU";
        return 0;
```

```
    }

    bool prim = true;
    for (int d = 2; d * d <= n; d++) {
        if (n % d == 0)
            prim = false;
    }

    if (prim) cout << "DA";
    else cout << "NU";
    return 0;
}
```

✅ Exercițiu rezolvat

Citește un număr și afișează „PRIM” sau „NEPRIM”.

👉 Codul de mai sus rezolvă exercițiul.

🔹 Lecția 8 – Vectori (tablouri unidimensionale)

Ce este un vector?

Un vector este o **listă de valori** (ex: notele elevilor).

Declarare

cpp Copiază codul

```cpp
int v[100];
```

Citire și afișare

cpp Copiază codul

```cpp
int n;
cin >> n;
for (int i = 0; i < n; i++)
    cin >> v[i];

for (int i = 0; i < n; i++)
    cout << v[i] << " ";
```

Maxim și minim în vector

cpp Copiază codul

```cpp
int mx = v[0], mn = v[0];
for (int i = 1; i < n; i++) {
    if (v[i] > mx) mx = v[i];
    if (v[i] < mn) mn = v[i];
}
```

✅ Exercițiu rezolvat

Citește un vector și afișează maximul.

cpp Copiază codul

```cpp
#include <iostream>
using namespace std;

int main() {
    int n, v[100];
    cin >> n;

    for (int i = 0; i < n; i++)
        cin >> v[i];

    int mx = v[0];
    for (int i = 1; i < n; i++)
        if (v[i] > mx) mx = v[i];

    cout << mx;
    return 0;
}
```

🔹 Lecția 9 – Sume parțiale (Prefix Sums)

Ideea

Construim un vector `sp`:

powershell — Copiază codul

```
sp[i] = v[0] + v[1] + ... + v[i]
```

Suma pe intervalul `[l..r]`:

```css
sp[r] - sp[l-1]
```

Exemplu complet

```cpp
#include <iostream>
using namespace std;

int main() {
    int n, v[100], sp[100];
    cin >> n;

    for (int i = 0; i < n; i++)
        cin >> v[i];

    sp[0] = v[0];
    for (int i = 1; i < n; i++)
        sp[i] = sp[i - 1] + v[i];

    int l, r;
    cin >> l >> r;

    int suma = sp[r] - (l > 0 ? sp[l - 1] : 0);
    cout << suma;
```

```
    return 0;
}
```

✅ Exercițiu rezolvat

Afișează suma elementelor de pe pozițiile 2–4.

👉 Se folosește exact metoda de mai sus.

🔹 Lecția 10 – Căutare binară (Binary Search)

⚠️ Vectorul trebuie să fie ordonat!

Ideea

- căutăm în jumătatea potrivită
- foarte rapid (logaritmic)

Exemplu

cpp Copiază codul

```cpp
#include <iostream>
using namespace std;

int main() {
    int n, v[100];
    cin >> n;

    for (int i = 0; i < n; i++)
        cin >> v[i];
```

```
    int x;
    cin >> x;

    int st = 0, dr = n - 1;
    bool gasit = false;

    while (st <= dr) {
        int mid = (st + dr) / 2;
        if (v[mid] == x) {
            gasit = true;
            break;
        } else if (v[mid] < x)
            st = mid + 1;
        else
            dr = mid - 1;
    }

    cout << (gasit ? "DA" : "NU");
    return 0;
}
```

🔹 Lecția 11 – Funcții

Ce este o funcție?

O funcție este un **mini-program** care rezolvă o sarcină.

Exemplu

cpp Copiază codul

```cpp
int suma(int a, int b) {
    return a + b;
}
```

cpp Copiază codul

```cpp
int main() {
    cout << suma(3, 5);
}
```

Parametri prin referință

cpp Copiază codul

```cpp
void swap2(int &a, int &b) {
    int t = a;
    a = b;
    b = t;
}
```

✅ Exercițiu rezolvat

Interschimbă două numere.

cpp Copiază codul

```cpp
#include <iostream>
using namespace std;

void swap2(int &a, int &b) {
    int t = a;
    a = b;
    b = t;
}

int main() {
    int a, b;
    cin >> a >> b;
    swap2(a, b);
    cout << a << " " << b;
    return 0;
}
```

🔹 Lecția 12 – Two Pointers (doi indici)

Ideea

Folosim doi indici (`st`, `dr`) pentru a parcurge eficient vectorul.

Exemplu – subsecvențe cu sumă ≤ S

cpp Copiază codul

```cpp
#include <iostream>
using namespace std;

int main() {
    int n;
    long long S;
    cin >> n >> S;
```



```cpp
    for (int i = 0; i < n; i++)
        cin >> v[i];

    int st = 0;
    long long sum = 0, cnt = 0;

    for (int dr = 0; dr < n; dr++) {
        sum += v[dr];
        while (sum > S) {
            sum -= v[st];
            st++;
        }
        cnt += dr - st + 1;
    }

    cout << cnt;
    return 0;
}
```

✅ FINAL PARTEA 2

- ✔ Numere prime
- ✔ Vectori
- ✔ Sume parțiale
- ✔ Căutare binară
- ✔ Funcții
- ✔ Two pointers

👉 **Urmează: PARTEA 3 – Lecțiile 13–18**
(difference arrays, Kadane, element majoritar, struct, STL, Greedy)

Scrie:
„PARTEA 3” 🚀

PARTEA 3

Perfect 👍
Încheiem cursul cu **PARTEA 3 – Lecțiile 13–18**, **pentru elevi**, cu **explicații clare + exerciții REZOLVATE**.

📘 CURS COMPLET – INFORMATICĂ CLASA A VII-A

PARTEA 3 – Lecțiile 13–18

🔹 Lecția 13 – Difference Arrays (tablouri de diferențe)

Problema

Avem multe operații de tip:

> adună `x` pe intervalul `[l..r]`

Dacă adunăm direct, e lent. Folosim **vectorul de diferențe**.

Ideea

- `d[l] += x`
- `d[r+1] -= x`
- reconstruim vectorul final prin sume cumulative

Exemplu complet

cpp Copiază codul

```cpp
#include <iostream>
using namespace std;

int main() {
    int n, q;
    cin >> n >> q;
```

```
    long long d[105] = {0};

    for (int i = 0; i < q; i++) {
        int l, r;
        long long x;
        cin >> l >> r >> x;
        d[l] += x;
        if (r + 1 < n)
            d[r + 1] -= x;
    }

    long long a[105];
    a[0] = d[0];
    for (int i = 1; i < n; i++)
        a[i] = a[i - 1] + d[i];

    for (int i = 0; i < n; i++)
        cout << a[i] << " ";

    return 0;
}
```

🔹 Lecția 14 – Subsecvența de sumă maximă (Algoritmul Kadane)

Problema

Dintr-un vector, găsim **cea mai mare sumă a unei subsecvențe consecutive**.

Ideea

- `cur` = suma curentă
- `best` = cea mai bună sumă găsită

Exemplu

cpp

Copiază codul

```cpp
#include <iostream>
using namespace std;

int main() {
    int n, v[100];
    cin >> n;

    for (int i = 0; i < n; i++)
        cin >> v[i];

    int cur = v[0], best = v[0];

    for (int i = 1; i < n; i++) {
        cur = max(v[i], cur + v[i]);
        best = max(best, cur);
    }

    cout << best;
    return 0;
}
```

✅ Exercițiu rezolvat

Afișează suma maximă a unei subsecvențe.

👉 Codul de mai sus este soluția.

🔹 Lecția 15 – Element majoritar (Boyer–Moore)

Problema

Găsim un element care apare de **mai mult de n/2 ori** (dacă există).

Algoritm

1. Alegem un candidat
2. Verificăm dacă apare de peste `n/2` ori

Exemplu

cpp

Copiază codul

```cpp
#include <iostream>
using namespace std;

int main() {
    int n, v[100];
    cin >> n;

    for (int i = 0; i < n; i++)
        cin >> v[i];
```

```
    int cand = 0, cnt = 0;
    for (int i = 0; i < n; i++) {
        if (cnt == 0) {
            cand = v[i];
            cnt = 1;
        } else if (v[i] == cand)
            cnt++;
        else
            cnt--;
    }

    int apar = 0;
    for (int i = 0; i < n; i++)
        if (v[i] == cand)
            apar++;

    if (apar > n / 2)
        cout << cand;
    else
        cout << "NU EXISTA";

    return 0;
}
```

🔹 Lecția 16 – struct (tipuri neomogene)

Ce este un struct?

Un `struct` grupează **mai multe date diferite**.

Exemplu

cpp Copiază codul

```cpp
#include <iostream>
using namespace std;

struct Elev {
    char nume[50];
    double medie;
};

int main() {
    Elev e;
    cin >> e.nume >> e.medie;
    cout << e.nume << " are media " << e.medie;
    return 0;
}
```

🔹 Lecția 17 – Sortare și căutare cu STL

sort

cpp Copiază codul

```cpp
#include <algorithm>
sort(v, v + n);
```

binary_search

cpp Copiază codul

```cpp
binary_search(v, v + n, x);
```

Exemplu complet

cpp Copiază codul

```cpp
#include <algorithm>
#include <iostream>
using namespace std;

int main() {
    int n, v[100];
    cin >> n;

    for (int i = 0; i < n; i++)
        cin >> v[i];

    sort(v, v + n);

    int x;
    cin >> x;
```

```
    cout << (binary_search(v, v + n, x) ? "DA" : "NU");
    return 0;
}
```

🔹 Lecția 18 – Greedy (alegerea cea mai bună acum)

Ce înseamnă Greedy?

Alegem mereu soluția **cea mai bună pe moment**.

⚠️ Nu funcționează la toate problemele!

Exemplu – activități fără suprapunere

cpp Copiază codul

```cpp
#include <bits/stdc++.h>
using namespace std;

struct Interval {
    int st, dr;
};

bool cmp(const Interval &a, const Interval &b) {
    return a.dr < b.dr;
}

int main() {
    int n;
```

```
    cin >> n;

    vector<Interval> a(n);
    for (int i = 0; i < n; i++)
        cin >> a[i].st >> a[i].dr;

    sort(a.begin(), a.end(), cmp);

    int cnt = 0, lastEnd = -1e9;
    for (int i = 0; i < n; i++) {
        if (a[i].st >= lastEnd) {
            cnt++;
            lastEnd = a[i].dr;
        }
    }

    cout << cnt;
    return 0;
}
```

🎉 FINAL CURS COMPLET

✅ Lecțiile 1–18

✅ Explicații pe înțelesul elevilor

✅ Exemple clare

✅ Exerciții REZOLVATE

✅ Perfect pentru clasă / meditații / studiu individual