FIȘĂ DE LUCRU · PYTHON · CLASA A IX-A · FIȘA I3 (INTENSIV) · LECȚIA PY9-I3

Subprograme în C++ și funcții predefinite din cmath și algorithm

Durată: 100 min Nivel: intensiv · Python + C++

CE VEI ȘTI DUPĂ LECȚIE

- Prezici efectul apelurilor C++ prin valoare și prin referință, al conversiilor la return și al funcțiilor din cmath.
- Repari erorile tipice: prototip lipsă, împărțire întreagă, & lipsă, sort pe un interval greșit, afișarea double.
- Scrii subprograme C++ complete, cu antet dat, și le testezi pe toate cazurile, inclusiv cele limită.

Ce trebuie să știi (o pagină)

Regula	Python	C++
definiția; fără rezultat	`def f(a, b):` ; fără `return`	`double f(int a, int b) { … }` ; `void f(…)`
funcția scrisă după `main`		prototip sus: `double f(int a, int b);`
tipul rezultatului	rămâne cum e	convertit la tipul din antet: `int f() { return 2.9; }` dă `2`
un număr transmis	nu poate fi schimbat	`int x` : copie · `int &x` : chiar variabila din apel
mai multe rezultate	`return a, b`	parametri `&` : `void f(int n, int &a, int &b)`
colecții	lista se poate modifica	tabloul `int a[]` se poate modifica; trimiți și `n`
`<cmath>`	`math.sqrt(-1)` : eroare; `round(2.5)` → `2`	`sqrt(-1)` → `nan` ; `round(2.5)` → `3` ; `pow` dă **`double`**
`<algorithm>`	`max(v)` , `v.sort(reverse=True)`	`min(a, b)` (același tip!), `max({a, b, c})` , `sort(a, a + n, greater<int>())`
zecimale fixe	`f"{x:.2f}"`	`cout << fixed << setprecision(2) << x;` (`<iomanip>`)

SFAT

Compilează cu `-Wall`: `control reaches end of non-void function` înseamnă un `return` lipsă.

Nivelul 1 · Prezice

★ citește și prezice

În toate programele, presupune `#include`-urile necesare și `using namespace std;`.

P1. Ce afișează?

```
int f(double x) { return x * 2; }
double g(int a, int b) { return a / b; }
double h(int a, int b) { return (double)a / b; }
// în main:
cout << f(2.7) << ' ' << g(7, 2) << ' ' << h(7, 2);
```

P2. Ce afișează? Ce s-ar schimba dacă și `a` ar avea `&`?

```
void p(int a, int &b) { a++; b += a; }
int main() { int x = 1, y = 10; p(x, y); p(x, y); cout << x << ' ' << y; }
```

P3. Un nume, trei variabile. Ce afișează?

```
int k = 5;
int f(int k) { k = k * 2; return k; }
int g() { k = k + 1; return k; }
int main() {
    cout << f(k) << ' ' << k << ' ';
    cout << g() << ' ' << k;
}
```

P4. Ce afișează C++? Care dintre cele cinci valori ar fi diferită în Python (cu `math.floor`, `math.ceil`, `round`, `int`)?

```
cout << floor(-3.5) << ' ' << ceil(-3.5) << ' ' << round(-3.5) << ' '
     << round(4.5) << ' ' << (int)(-3.5);
```

P5. Ce afișează rândul de mai jos fără nicio setare? Dar după `cout << fixed << setprecision(2);`?

```
cout << sqrt(2) * 100 << ' ' << pow(10, 6) << ' ' << 1.0 / 3 << ' ' << pow(2, 20);
```

P6. Ce conține tabloul după `sort` și ce afișează ultimul rând?

```
int a[] = {5, 2, 8, 2, 9, 1};
sort(a + 1, a + 5);
cout << max(a[0], a[5]) << ' ' << min({a[1], a[3], a[5]});
```

P7. Ce afișează? Ce face, în cuvinte, succesiunea celor trei apeluri?

```cpp
void ordoneaza(int &a, int &b) { if (a > b) { int t = a; a = b; b = t; } }
int main() {
    int x = 7, y = 3, z = 5;
    ordoneaza(x, y); ordoneaza(y, z); ordoneaza(x, y);
    cout << x << y << z;
}
```

P8. Același program în două limbaje. Ce afișează fiecare?

```cpp
void f(int v[], int n) { v[0] += n; n++; }
// în main:
int a[1] = {1}, k = 5;
f(a, k); f(a, k);
cout << a[0] << ' ' << k;
```

```python
def f(v, n): v[0] += n; n += 1
a = [1]; k = 5
f(a, k); f(a, k)
print(a, k)
```

Nivelul 2 · Găsește și repară

★★ găsește și repară

P9. Programul nu compilează: `error: 'arie' was not declared in this scope`. Repară-l în **două** feluri diferite, fără să modifici corpul lui `main`.

```cpp
int main() {
    int L, l;
    cin >> L >> l;
    cout << arie(L, l) << '\n';
    return 0;
}
int arie(int L, int l) { return L * l; }
```

P10. Pentru notele `9 10 10`, funcția ar trebui să dea media `9.67`, dar programul afișează `9`. Sunt **două** greșeli. Care?

```cpp
int media(int a, int b, int c) { return (a + b + c) / 3; }
```

P11. După `interschimba(x, y)` cu `x = 3`, `y = 8`, programul afișează tot `3 8`. De ce? Repară antetul. Cum se scrie interschimbarea în Python?

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

P12. Depui 10 000 de lei cu 5% pe an, 3 ani. Programul afișează `10000`. După ce repari prima greșeală, afișează `11576.3`, deși suma corectă este 11 576,25 lei. Repară ambele probleme.

```cpp
double P = 10000; int p = 5, t = 3;     // p: procentul
cout << P * pow(1 + p / 100, t);
```

P13. *Rânduri amestecate.* Programul citește coordonatele a două puncte și afișează distanța dintre ele cu 3 zecimale. Pune rândurile în ordine (după `#include <iostream>`, `<iomanip>`, `<cmath>` și

`using namespace std;`). Un rând este în plus: care și ce s-ar afișa dacă l-ai folosi?

```
A)  return sqrt((x2 - x1) * (x2 - x1) + (y2 - y1) * (y2 - y1));
B)  int main() { int a, b, c, d; cin >> a >> b >> c >> d;
C)  double distanta(int x1, int y1, int x2, int y2) {
D)  return 0; }
E)  }
F)  int distanta(int x1, int y1, int x2, int y2) {
G)  cout << fixed << setprecision(3) << distanta(a, b, c, d) << '\n';
```

Teste: `1 1 4 5` → `5.000`; `0 0 1 1` → `1.414`.

P14. Programul ar trebui să afișeze podiumul (primele trei scoruri, descrescător), dar afișează `60 99 95`. Repară-l. Apoi, rândul adăugat la final nu compilează (`no matching function for call to 'max(int&, double&)'`). De ce? Repară-l și pe el.

```cpp
int n = 5, s[] = {70, 95, 82, 99, 60};
sort(s, s + n - 1);
cout << s[n - 1] << ' ' << s[n - 2] << ' ' << s[n - 3] << '\n';
double bonus = 2.5;
cout << max(s[0], bonus);
```

Nivelul 3 · Scrie cod

★★ scrie cod

Scrie programe C++ complete. Respectă **exact** antetele date și verifică **toate** testele.

P15. Făina la pachete. `int pachete_complete(double total, double cap)` și `int pachete_necesare(double total, double cap)`: câte pachete **pline** de `cap` kg ies din `total` kg și câte pachete trebuie ca să încapă toată cantitatea. Afișează ambele valori.

intrare	ieșire
`23.5 2`	`11 12`
`10 2.5`	`4 4`
`0.3 1`	`0 1`

P16. Economii. `double suma_finala(double P, double r, int n, int t)` (cu `r` în **procente**) aplică formula `A = P·(1 + r/n)^(n·t)`; afișezi cu 2 zecimale. Apoi `int ani_pana_la(double P, double r, double tinta)`, cu dobândă anuală: după câți ani suma **depășește** ținta; returnează `-1` dacă nu se întâmplă niciodată.

Teste: `suma_finala(2000, 4, 1, 5)` → `2433.31`; `suma_finala(2000, 4, 4, 5)` → `2440.38`; `suma_finala(2500, 0, 4, 10)` → `2500.00`; `ani_pana_la(1000, 5, 1500)` → `9`; `ani_pana_la(1000, 10, 2000)` → `8`; `ani_pana_la(1000, 0, 2000)` → `-1`.

P17. Fișa sportivului. `void statistica(int v[], int n, int &mn, int &mx, double &med)` calculează scorul minim, maximul și media. Programul citește `n` și cele `n` scoruri.

intrare	ieșire
`5` / `7 9 10 6 8`	`6 10 8.00`

intrare	ieșire
`1` / `42`	`42 42 42.00`
`3` / `9 9 10`	`9 10 9.33`

P18. Terenul triunghiular. Se citesc coordonatele întregi ale vârfurilor `A`, `B`, `C`. Afișează perimetrul (cu `double distanta(int x1, int y1, int x2, int y2)`) și aria, cu formula `|(x2 – x1)(y3 – y1) – (x3 – x1)(y2 – y1)| / 2` (într-o funcție `arie`), ambele cu 2 zecimale.

intrare	ieșire
`0 0 3 0 0 4`	`12.00 6.00`
`1 1 4 5 -2 5`	`16.00 12.00`
`0 0 1 1 2 2`	`5.66 0.00`

P19. Scorul final al jocului. `int scor_final(int puncte, int secunde, int bonusuri)`: fiecare bonus aduce 50 de puncte; pentru fiecare secundă peste 120 se scad 2 puncte; scorul nu poate fi negativ. Folosește `max` (nu `if`).

intrare (puncte, secunde, bonusuri)	ieșire
`1000 100 2`	`1100`
`1000 150 0`	`940`
`50 300 0`	`0`

P20. Ceasul. `void adauga(int &ore, int &minute, int x)` adaugă `x` minute la ora dată (ceas de 24 de ore). Afișarea: două cifre pentru ore și minute (`setfill('0')` și `setw(2)` din `<iomanip>`).

intrare (ore, minute, x)	ieșire
`23 50 20`	`00:10`
`10 0 135`	`12:15`
`5 59 1441`	`06:00`

Nivelul 4 · Pune mintea la contribuție

★★★ pune mintea la contribuție

P21. Trei feluri de a rotunji. Completează tabelul pentru `m` = 8,5; 9,5; 7,49; 6,5; −2,5, cu coloanele: `round(m)` în C++, `round(m)` în Python, `floor(m + 0.5)`, `(int)m`. Care variantă respectă regula școlii („de la ,50 în sus") pentru note? Pentru ce valori diferă `round` din C++ de `floor(m + 0.5)`?

P22. pow nu e pentru numere întregi mari. Afișează `long long a = pow(3, 39);` și compară cu `long long putere(long long b, int e)`, scrisă cu o buclă. Ce diferență obții? De ce? Cum calculează Python `3 ** 39`?

P23. Ce apeluri compilează? Pentru `void salariu(double brut, double &net, double &impozit)` și `double n, i, b1 = 3000, b2 = 1000;`, care dintre apeluri sunt corecte? Explică pentru fiecare.

```
salariu(4000, n, i);          // (a)
salariu(4000, 2925, i);       // (b)
cout << salariu(4000, n, i);  // (c)
salariu(b1 + b2, n, i);       // (d)
```

Când ai alege `double net(double brut)` în locul lui `salariu`? Un argument pentru fiecare.

P24. Problemă inversă. După `int r = f(x, y); cout << r << ' ' << x;` se afișează `40 8`. Ce valori aveau `x` și `y` înainte de apel? Ce s-ar afișa, pentru aceleași valori, dacă antetul ar fi `int f(int a, int b)`?

```
int f(int &a, int b) { a = a + b; b = a - b; return a * b; }
```

P25. Mini-proiect: aruncarea sulițeI, în două limbaje. Fiecare dintre cei `n` sportivi are 3 aruncări (0 = aruncare nulă). Rezultatul unui sportiv este cea mai bună aruncare (`double cea_mai_buna(double a, double b, double c)`, cu `max({…})`). Afișează rezultatele în ordine descrescătoare și media lor, cu 2 zecimale; pentru `n = 0` afișează `fara sportivi`. Scrie-l în C++ **și** în Python. Compară: câte rânduri ai scris și ce se întâmpla în fiecare limbaj dacă uitai cazul `n = 0`.

Test: `4`, apoi `62.5 64.1 63.0` / `70.2 0 68.9` / `58.0 59.5 61.25` / `66.6 66.6 65.0` → `70.20 66.60 64.10 61.25` și `65.54`.

Probleme de lucru în clasă (pbinfo)

Le rezolvăm împreună în ședință, în ordinea de mai jos.

- [ ] #1282 **radical** — ușoară
  https://www.pbinfo.ro/probleme/1282/radical
- [ ] #1376 **laturitriunghi** — ușoară
  https://www.pbinfo.ro/probleme/1376/laturitriunghi
- [ ] #1625 **ec2** — ușoară
  https://www.pbinfo.ro/probleme/1625/ec2

Temă pentru acasă (pbinfo)

Trimite soluțiile pe pbinfo și notează punctajul obținut. Bifează ce ai terminat.

- [ ] #922 **puncte** — ușoară
  https://www.pbinfo.ro/probleme/922/puncte
- [ ] #2603 **primacifraapartiizecimale** — ușoară
  https://www.pbinfo.ro/probleme/2603/primacifraapartiizecimale
- [ ] #129 **sortare** — ușoară
  https://www.pbinfo.ro/probleme/129/sortare