FIȘĂ DE LUCRU · CLASA A XI-A · FIȘA 1 · ACOPERĂ LECȚIILE XI.1–XI.2

Subprograme: funcții, parametri, referințe

Durată: 100 min Nivel: toate profilurile

CE VEI ȘTI DUPĂ LECȚIE

- Urmărești apeluri cu parametri prin valoare și prin referință, variabile locale și globale.
- Alegi antetul corect: tipul returnat, ce se primește și ce se furnizează prin `&`.
- Scrii subprograme ca la BAC (III.1): doar definiția, cu numele și parametrii din enunț.

Ce trebuie să știi (o pagină)

Regula	Exemplu	Ce înseamnă
funcția întoarce o valoare	`int nrCif(int n) { … return c; }`	se folosește în expresii: `nrCif(x) + 1`
`bool` pentru „este … ?”	`bool prim(int n)`	`if (prim(x))`
`void` face ceva, nu întoarce	`void afis(int v[], int n)`	se apelează ca instrucțiune
`return` oprește **toată** funcția	`if (n % d == 0) return false;`	nu doar bucla
prin **valoare** = copie	`void f(int x)`	`main` nu vede modificarea
prin **referință** = același obiect	`void f(int &x)`	`main` vede modificarea; la apel: **variabilă**, nu `5`
vectorul se transmite ca adresă	`void f(int v[], int &n)`	`v` se modifică fără `&`; `n` cu `&` dacă se schimbă
matrice: coloanele sunt obligatorii	`int f(int a[][101], int n, int m)`	`int a[][]` nu compilează
locală vs globală	locala „ascunde” globala cu același nume	globalele pornesc de la 0, localele de la „gunoi”
BAC III.1: „furnizează prin `n`”	`void elimin(int &n)`	pui rezultatul **înapoi** în `n`
BAC III.1: „returnează”	`int f(int n)`	fără `cin`, fără `cout`, fără `main`

ATENȚIE — GREȘELI FRECVENTE

Dacă primești `n` prin referință și îl „consumi” cu `n /= 10`, la final scrie rezultatul în `n`. Dacă ai nevoie de valoarea inițială, fă o copie locală la început.

Nivelul 1 · Prezice

★ citește și prezice

P1. Funcțiile `f` și `g` se ajută între ele. Completează tabelul, apoi spune **în cuvinte** ce calculează fiecare.

```
int f(int a, int b) { while (a != b) if (a > b) a -= b; else b -= a; return a; }
int g(int a, int b) { return a / f(a, b) * b; }
```

f(12, 18)	f(7, 3)	f(f(12, 18), 4)	g(4, 6)	g(f(24, 36), 8)	f(100, 1)	g(5, 5)

P2. Patru funcții cu **același corp**, dar parametri diferiți. Pornind de fiecare dată de la `a = 3, b = 4`, ce valori au `a` și `b` după apel?

```
void A(int x, int y)   { x = x + y; y = y * 2; }
void B(int &x, int y)  { x = x + y; y = y * 2; }
void C(int x, int &y)  { x = x + y; y = y * 2; }
void D(int &x, int &y) { x = x + y; y = y * 2; }
```

A(a, b)	B(a, b)	C(a, b)	D(a, b)	D(b, a)	D(a, a)

P3. Globală, locală cu același nume, parametru cu același nume. Ce afișează?

```
int k = 10;
int f(int x) { k = k + x; return k; }
int g(int k) { k = k * 2; return k; }
int main() {
    int a = f(1); int b = g(a); int c = f(b);
    cout << a << ' ' << b << ' ' << c << ' ' << k;
}
```

P4. Ce afișează? Ce se află în `v` **după** poziția `n - 1` și de ce nu e „curat”?

```
void elimina(int v[], int &n, int x) {
    int k = 0;
    for (int i = 0; i < n; i++) if (v[i] != x) v[k++] = v[i];
    n = k;
}
int main() {
    int v[] = {3, 1, 3, 3, 2, 3}, n = 6;
    elimina(v, n, 3);
    cout << n << " |";
    for (int i = 0; i < 6; i++) cout << ' ' << v[i];
}
```

P5. (BAC) Potrivește fiecare enunț cu antetul corect. Două antete sunt în plus.

Enunțuri: **1)** primește `n` și returnează cifra maximă; **2)** primește `n` și furnizează prin `mn`, `mx` cifra minimă și maximă; **3)** primește `n` și îl înlocuiește cu oglinditul lui; **4)** primește un vector cu `n` elemente și șterge din el elementele nule.

```
a) void cifre(int n, int &mn, int &mx)     b) int cifMax(int n)
c) void oglinda(int &n)                    d) int cifre(int n, int mn, int mx)
e) void sterge(int v[], int &n)            f) void sterge(int v[], int n)
```

P6. Ce returnează `f(91)`, `f(97)`, `f(1)` și `f(f(91) * f(15))`? Ce afișează secvența de dedesubt pentru `n = 360`? Spune în cuvinte ce face fiecare.

```
int f(int n) { for (int d = 2; d <= n; d++) if (n % d == 0) return d; return 0; }
// …
while (n > 1) { cout << f(n) << ' '; n = n / f(n); }
```

P7. Matricea `a` are 4 linii și 4 coloane (indexate de la 1). Cât este `s(a, 4, 5)`? Pentru ce valoare a lui `k` se obține cea mai mare sumă? Ce elemente adună funcția, în cuvinte?

```
int s(int a[][10], int n, int k) {
    int r = 0;
    for (int i = 1; i <= n; i++)
        for (int j = 1; j <= n; j++) if (i + j == k) r += a[i][j];
    return r;
}
```

Matricea, pe linii: `1 2 3 4` / `5 6 7 8` / `9 1 2 3` / `4 5 6 7`.

P8. Ce afișează pentru `x = 9, y = 4, z = 6`? Dacă ștergi **al treilea** apel, dă valori pentru `x, y, z` pentru care rezultatul nu mai este ordonat.

```
void ord(int &a, int &b) { if (a > b) { int t = a; a = b; b = t; } }
// …
ord(x, y); ord(y, z); ord(x, y);
cout << x << y << z;
```

Nivelul 2 · Găsește și repară

★★ găsește și repară

P9. Subprogramul trebuie să furnizeze cifra minimă și maximă a lui `n`. Pentru `n = 3752` furnizează `0` și `9`, ba chiar pentru orice `n`. Ce e greșit? Repară **două** valori.

```
void cifMinMax(int n, int &mn, int &mx) {
    mn = 0; mx = 9;
    while (n) { int c = n % 10; if (c < mn) mn = c; if (c > mx) mx = c; n /= 10; }
}
```

P10. Funcția „verifică” dacă `n ≥ 2` este prim. Care este cel mai mic `n ≥ 4` pentru care greșește? Scrie toate valorile până la 30 pentru care greșește. Ce se întâmplă la `n = 2` sau `n = 3` (ce `return` se execută)? Repară.

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

P11. *Puzzle cu rânduri amestecate.* Subprogramul `inlocuire(n, c1, c2)` înlocuiește în `n` fiecare cifră `c1` cu `c2`, iar rezultatul este furnizat tot prin `n` (de exemplu `12131`, `1`, `7` → `72737`). Pune rândurile în ordine; un rând este în plus.

```
A)  n = r;
B)  void inlocuire(int &n, int c1, int c2) {
C)      r = r + c * p;  p = p * 10;  n = n / 10;
D)  } while (n != 0);
E)  return r;
F)      int c = n % 10;  if (c == c1) c = c2;
G)  int r = 0, p = 1;
H)  do {
I)  }
```

P12. Completează golurile. `pozMax` întoarce **ultima** poziție pe care apare maximul în `v[1..n]`, iar `rotire` mută toate elementele cu o poziție la stânga, primul ajungând ultimul. Ce dau pentru `v = (4, 9, 2, 9, 1)`?

```
int pozMax(int v[], int n) {
    int p = 1;
    for (int i = 2; i <= n; i++) if (v[i] ____ v[p]) p = i;
    return p;
}
void rotire(int v[], int n) {
    int aux = ______;
    for (int i = 1; i ______; i++) v[i] = ______;
    v[n] = aux;
}
```

P13. *Ce greșeală a făcut elevul?* Spune de ce e falsă fiecare afirmație.

a. „Ca să modific un vector într-o funcție, trebuie să-l scriu `int &v[]`."

b. „`swap(a[i], a[j])` nu poate funcționa, pentru că `a[i]` nu e o variabilă."

c. „Scriu `void dublu(int x) { x = 2 * x; }` și apoi `dublu(a); cout << a;` : afișează dublul lui `a`."

P14. Ce afișează `cout << nrDiv(6) << ' ' << nrDiv(6);`? De ce cele două apeluri dau rezultate diferite? Repară mutând **un singur rând**.

```
int cnt;
int nrDiv(int n) {
    for (int d = 1; d <= n; d++) if (n % d == 0) cnt++;
    return cnt;
}
```

Nivelul 3 · Scrie cod

★★ scrie cod

Scrie **subprogramul** cerut (ca la BAC), apoi un `main` scurt care citește datele, îl apelează și afișează rezultatul, ca să-l poți testa.

P15. PIN-ul sigur. `bool pinBun(int n)` primește un număr de exact 4 cifre și întoarce `true` dacă nu are două cifre vecine egale și cifrele **nu** sunt consecutive crescător (ca `1234`) sau descrescător (ca `4321`).

1234	1123	4321	1357	2024	6789	1243
NU	NU	NU	DA	DA	NU	DA

P16. Ceasul digital. `void adunaMinute(int &h, int &m, int x)` adaugă `x` minute la ora `h:m` (format 24 h). Teste: `23 50 15 → 0 5`; `10 0 1440 → 10 0`; `8 30 95 → 10 5`; `0 0 0 → 0 0`.

P17. Playlistul fără repetări. `void comprima(int v[], int &n)` păstrează din fiecare grup de valori **vecine egale** o singură valoare (`v` indexat de la 1).

intrare	ieșire
`7` / `3 3 5 5 5 3 8`	`3 5 3 8`
`1` / `4`	`4`
`4` / `2 2 2 2`	`2`

P18. Cifra de control. Scrie `int sumCif(int n)` și apoi `int cifraControl(int n)`, care adună cifrele **în mod repetat** până rămâne o singură cifră, **apelând** `sumCif`. Teste: `9875 → 2` (29 → 11 → 2); `0 → 0`; `7 → 7`; `999999999 → 9`.

P19. Sala de cinema. Matricea are `n` rânduri și `m` locuri (`0` = liber, `1` = ocupat). `int randLiber(int a[][101], int n, int m)` întoarce rândul cu **cele mai multe** locuri libere (primul, la egalitate), sau `0` dacă sala e plină.

intrare	ieșire
`3 4` / `1 0 1 1` / `0 0 1 0` / `1 1 1 1`	`2`
`2 3` / `1 1 1` / `1 1 1`	`0`
`2 2` / `0 0` / `0 0`	`1`

P20. Rețeta simplificată. `void fractie(int &a, int &b)` aduce fracția `a/b` (b > 0, a ≥ 0) la forma ireductibilă, folosind o funcție `cmmdc`. Teste: `12 18 → 2 3`; `7 5 → 7 5`; `100 25 → 4 1`; `0 5 → 0 1`. (Atenție: varianta cu scăderi a lui `cmmdc` se blochează la `0 5`!)

Nivelul 4 · Pune mintea la contribuție

★★★ pune mintea la contribuție

P21. Problemă inversă. Cu `void t(int &x, int y) { x = x - y; y = x + y; }`, după apelurile `t(a, b); t(b, a);` se afișează `cout << a << ' ' << b;` și apare `1 7`. Ce valori aveau `a` și `b` la început? Arată calculul (fără încercări la întâmplare).

P22. Interschimbare fără auxiliar. `void schimb(int &a, int &b) { a = a + b; b = a - b; a = a - b; }` funcționează pentru `schimb(x, y)`. Ce se întâmplă la `schimb(x, x)` pentru `x = 5`? De ce? Varianta cu variabilă auxiliară are aceeași problemă?

P23. Mini-proiect: palindromul din adunări. Cât timp `n` nu e palindrom, îl înlocuiești cu `n + oglindit(n)`. Scrie `oglindit`, `palindrom` și `int pasi(long long n)` (numărul de pași; `-1` dacă nu ajungi la palindrom în cel mult 5 pași). Teste: `56 → 1`, `57 → 2`, `11 → 0`, `87 → 4`, `89 → -1`. De ce trebuie `long long`?

P24. De ce funcționează? Subprogramul `nrDivPare(n)` trebuie să întoarcă numărul divizorilor **pari** ai lui `n`. Un elev scrie: `if (n % 2) return 0; return nrDiv(n / 2);`. Explică de ce e corect și verifică pe `12 → 4`, `64 → 6`, `7 → 0`, `2 → 1`.

P25. Detectivul. Funcția `int f(int n) { int s = 0; while (n) { s += n % 10; n /= 100; } return s; }`. Ce calculează (încearcă `f(12345)`)? **Câte** numere de trei cifre au `f(n) = 17`? Care este cel mai mic și cel mai mare? Raționează, apoi verifică printr-un program.

Probleme de lucru în clasă (pbinfo)

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

- [ ] `#14` **primalitate** — ușoară
  https://www.pbinfo.ro/probleme/14/primalitate
- [ ] `#905` **detcifre** — ușoară
  https://www.pbinfo.ro/probleme/905/detcifre
- [ ] `#72` **fsumvec** — ușoară
  https://www.pbinfo.ro/probleme/72/fsumvec
- [ ] `#899` **ordonaref** — ușoară
  https://www.pbinfo.ro/probleme/899/ordonaref

Temă pentru acasă (pbinfo)

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

- [ ] `#3613` **fmultiplu-bac** — ușoară
  https://www.pbinfo.ro/probleme/3613/fmultiplu-bac
- [ ] `#3614` **fsuma-bac** — ușoară
  https://www.pbinfo.ro/probleme/3614/fsuma-bac
- [ ] `#2805` **duplicare** — ușoară
  https://www.pbinfo.ro/probleme/2805/duplicare
- [ ] `#894` **cifminmax** — medie
  https://www.pbinfo.ro/probleme/894/cifminmax