FIȘĂ DE LUCRU · CLASA A X-A · FIȘA 8 · ACOPERĂ LECȚIILE X.10–X.11

Șiruri de caractere

Durată: 100 min Nivel: mate-info

CE VEI ȘTI DUPĂ LECȚIE

- Prezici ce fac strlen, strcpy, strcat, strcmp, strchr, strstr și strtok pe un șir dat.
- Recunoști capcanele: == pe șiruri, '\0' uitat, strtok care taie textul, indicele sărit la ștergere.
- Scrii programe scurte care prelucrează caractere, cuvinte și secvențe dintr-un text.

Ce trebuie să știi (o pagină)

Regula	Exemplu	Rezultat
`char s[N]`: cel mult `N − 1` caractere + `'\0'`	`char s[5] = "info";`	plin
`cin >> s`: un cuvânt; `cin.getline(s, N)`: o linie	`ana are`	`"ana"` / `"ana are"`
ASCII: `'0'` = 48, `'A'` = 65, `'a'` = 97, consecutive	`'d' - 'a'`, `'q' - 'a' + 'A'`	`3`, `'Q'`
`strlen(s)`: lungimea, fără `'\0'`	`strlen("ana are")`	`7`
`strcpy(d, s)` copiază; `strcat(d, s)` lipește	`strcpy(d, "ab"); strcat(d, "cd");`	`"abcd"`
`strcmp(a, b)`: `< 0`, `0`, `> 0`, după coduri	`strcmp("Zoe", "ana")`	`< 0`
`strchr`, `strstr`: adresă sau `NULL`; poziția = `p - s`	`strchr(s, 'e') - s`, `s = "lalea"`	`3`
`s + k`: șirul de la poziția `k`	`cout << s + 2`, `s = "pixel"`	`xel`
`strtok(s, sep)`, apoi `strtok(NULL, sep)`	`p = strtok(s, " "); while (p) {…}`	cuvânt cu cuvânt

ATENȚIE — GREȘELI FRECVENTE

`a == b` și `a = b` **nu** lucrează cu textul, ci cu adresele: folosește `strcmp(a, b) == 0` și `strcpy(a, b)`. `strtok` **modifică** șirul (pune `'\0'` peste separatori): dacă mai ai nevoie de text, fă întâi o copie. După ștergeri manuale nu uita `s[k] = '\0';`.

Nivelul 1 · Prezice

★ citește și prezice

P1. Pentru `char s[] = "Timisoara2026";` completează valoarea fiecărei expresii.

expresie	valoare	expresie	valoare
`strlen(s)`		`strchr(s, 'o') - s`	
`s[4]`		`strstr(s, "20") - s`	
`s[4] - 'a'`		`s[strlen(s) - 1]`	
`s[9] - '0' + 1`		`char(s[0] + 1)`	

P2. Ce afișează?

```
char s[] = "programare";
for (int i = 0; s[i]; i++)
    if (strchr("aeiou", s[i])) s[i] = '*';
    else if (i % 2 == 0) s[i] = s[i] - 'a' + 'A';
cout << s;
```

P3. Scrie semnul lui `strcmp(a, b)` (`< 0`, `0` sau `> 0`) pentru fiecare pereche. La care dintre ele te-ar fi păcălit „bunul simț”?

`a`	`"mere"`	`"ana"`	`"100"`	`"Zebra"`	`"b"`	`"ok"`
`b`	`"Mere"`	`"ana "`	`"99"`	`"alb"`	`"abecedar"`	`"ok"`
semn						

P4. Ce afișează secvența? Atenție: ultimul rând afișat este chiar șirul `s`, după ce a trecut `strtok` prin el.

```
char s[] = "  12,ana;;  are,,3 mere ";
int nr = 0;
for (char *p = strtok(s, " ,;"); p; p = strtok(NULL, " ,;"))
{ nr++; cout << p[0]; }
cout << ' ' << nr << '\n' << s;
```

P5. Mărțișor. Ce afișează?

```
char a[30] = "mar", b[30] = "tisor", c[30];
strcat(a, b);
strcpy(c, a + 4);
c[0] = 'u';
strcpy(a + 3, c);
cout << a << ' ' << c << ' ' << strlen(a);
```

P6. Ce afișează pentru `s = "aaabccdddd"`? Spune **în cuvinte** ce numără secvența. Dă apoi un șir de 6 litere pentru care se afișează `6` și unul pentru care se afișează `1`.

```
int k = 1;
for (int i = 1; s[i]; i++)
    if (s[i] != s[i - 1]) k++;
cout << k;
```

P7. Cifrul „oglindă” (Atbash). În ce se transformă `info bac!`? Ce obții dacă aplici secvența **de două ori**? De ce?

```
for (int i = 0; s[i]; i++)
    if (s[i] >= 'a' && s[i] <= 'z') s[i] = 'z' - (s[i] - 'a');
```

P8. Ce afișează? (scrie mai întâi indicii sub litere)

```
char s[] = "abracadabra";
char *p = strstr(s, "cad");
cout << p - s << ' ' << p + 3 << ' ' << strchr(p, 'b') - s << ' '
     << strstr(s + 1, "abra") - s;
```

Nivelul 2 · Găsește și repară

★★ găsește și repară

P9. Programul trebuie să transforme tot textul în majuscule. Pentru `ana are mere` afișează doar `ANA`. Unde a dispărut restul? Repară.

```
cin.getline(s, 256);
int n = strlen(s);
for (int i = 0; i < n; i++)
    if (s[i] >= 'a' || s[i] <= 'z') s[i] = s[i] - 32;
cout << s;
```

P10. Parola corectă este `bac2026`. Programul răspunde mereu „Parola gresita”, chiar și pentru parola corectă. Găsește **două** greșeli.

```
char parola[7];
cin >> parola;
if (parola == "bac2026") cout << "Acces permis";
else cout << "Parola gresita";
```

P11. Secvența trebuie să șteargă **toate** spațiile. Pentru `a b c` afișează `a b c`. Explică de ce și repară în două feluri.

```
for (int i = 0; s[i]; i++)
    if (s[i] == ' ')
        for (int j = i; s[j]; j++) s[j] = s[j + 1];
```

P12. *Puzzle cu rânduri amestecate.* Programul transformă în majusculă prima literă a fiecărui cuvânt (`ana are mere` → `Ana Are Mere`). Un rând este în plus.

```
A)  if (s[i] >= 'a' && s[i] <= 'z' && (i == 0 || s[i - 1] == ' '))
B)  cin.getline(s, 201);
C)  for (int i = 0; s[i]; i++)
D)  s[i] = s[i] - 'a' + 'A';
E)  cout << s;
F)  s[i] = s[i] + 'a' - 'A';
G)  char s[201];
```

P13. Completează golurile ca programul să afișeze litera mică cea mai frecventă și de câte ori apare (la egalitate, prima în alfabet). Ce s-ar schimba pentru `mississippi` dacă ai scrie `>=` în loc de `>`?

```
char s[256]; int fr[26];          // globale
cin.getline(s, 256);
for (int i = 0; s[i]; i++)
    if (s[i] >= 'a' && s[i] <= 'z') fr[________]++;
int best = 0;
for (int c = 1; c < 26; c++)
    if (fr[c] ____ fr[best]) best = c;
cout << ________ << ' ' << fr[best];
```

P14. Elevul vrea să numere de câte ori apare cuvântul `bac` în text. Pentru `bac bac info bac` programul afișează `1`. Ce greșeală a făcut?

```
int cnt = 0;
for (char *p = strtok(s, " "); p; p = strtok(NULL, " "))
    if (strcmp(p, "bac")) cnt++;
cout << cnt;
```

Nivelul 3 · Scrie cod

★★ scrie cod

Pentru fiecare program, verifică **toate** testele date înainte să spui că e gata.

P15. Parola puternică. Se citește o parolă (fără spații). Afișează `PUTERNICA` dacă are cel puțin 8 caractere și conține cel puțin o literă mică, o literă mare și o cifră; altfel `SLABA`.

intrare	`Bac2026ok`	`parola123`	`Ab1`	`ABCdefgh`
ieșire	`PUTERNICA`	`SLABA`	`SLABA`	`SLABA`

P16. Numele de utilizator. Din numele complet al unui elev (cuvinte separate prin câte un spațiu) se formează contul școlii: inițialele tuturor cuvintelor, **în afară de ultimul**, urmate de ultimul cuvânt întreg, totul cu litere mici.

intrare	`Ana Popescu`	`mihai Andrei IONESCU`	`Eva`
ieșire	`apopescu`	`maionescu`	`eva`

P17. Cenzura. Se citesc un text (prima linie) și un cuvânt `t` (a doua linie). Înlocuiește fiecare apariție a lui `t` în text cu tot atâtea `*`. Folosește `strstr`.

intrare	ieșire
`ce mai faci, mai bine?` / `mai`	`ce *** faci, *** bine?`
`aaaa` / `aa`	`****`
`aaa` / `aa`	`**a`
`salut` / `xyz`	`salut`

P18. Bonul de cumpărături. Afișează suma tuturor numerelor care apar într-un text (un număr = cifre consecutive).

intrare	`am 3 mere si 12 pere, 100 lei`	`fara numere`	`2026`	`a1b2c3`
ieșire	`115`	`0`	`2026`	`6`

P19. Compresia. Un cuvânt se comprimă scriind fiecare grup de litere identice consecutive ca litera urmată de lungimea grupului (vezi P6).

intrare	`aaabccdddd`	`x`	`aabbaa`	`zzzzzzzzzzz`
ieșire	`a3b1c2d4`	`x1`	`a2b2a2`	`z12`

P20. Adresa de e-mail. O adresă este `VALID` dacă are exact un `@`, care nu e primul caracter, iar după `@` există un `.` care nu e imediat după `@` și nu e ultimul caracter. Altfel `INVALID`. Folosește `strchr`.

intrare	`ana@liceu.ro`	`ana@@liceu.ro`	`@liceu.ro`	`ana@liceu`	`ana@liceu.`	`ana@.ro`
ieșire	`VALID`	`INVALID`	`INVALID`	`INVALID`	`INVALID`	`INVALID`

Nivelul 4 · Pune mintea la contribuție

★★★ pune mintea la contribuție

P21. Spionul invers. Secvența de mai jos a criptat un cuvânt și a afișat `sbnxx`. Care era cuvântul? Ce cuvânt dă `zzzz`? Scrie secvența de **decriptare** și explică de ce ai nevoie de `+ 26`.

```
for (int i = 0; s[i]; i++)
    if (s[i] >= 'a' && s[i] <= 'z') s[i] = 'a' + (s[i] - 'a' + i) % 26;
```

P22. Palindrom fără copie. Verifică dacă o propoziție este palindrom ignorând spațiile și diferența mare/mică, **fără** să construiești alt șir (doar doi indici, `i` de la stânga și `j` de la dreapta). Teste: `Ele fac cafele` → `DA`, `Step on no pets` → `DA`, `Ana are mere` → `NU`.

P23. Numere prea mari pentru `long long`. Două numere naturale cu până la 100 de cifre (fără zerouri în față) sunt citite ca șiruri. De ce `strcmp` spune că `"123"` < `"99"`? Scrie o comparație corectă folosind `strlen` și `strcmp` și explică **de ce** funcționează când lungimile sunt egale.

P24. Ce se întâmplă dacă... folosești `strtok` în interiorul altei bucle cu `strtok`? Prezice ce afișează și explică. Cum ai repara?

```
char s[] = "a-b c-d e-f";
for (char *p = strtok(s, " "); p; p = strtok(NULL, " "))
{
    for (char *q = strtok(p, "-"); q; q = strtok(NULL, "-")) cout << q;
    cout << '|';
}
```

P25. Anagrame cu un singur vector. Două cuvinte (litere mici) sunt anagrame dacă au aceleași litere de același număr de ori (`sare` și `rase`). Rezolvă cu **un singur** vector de frecvență `f[26]` și o singură verificare la final. De ce ajunge un singur vector? Teste: `sare rase` → `DA`, `mama mamm` → `NU`, `ab abc` → `NU`.

Probleme de lucru în clasă (pbinfo)

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

- [ ] #11 **vocale** — ușoară
  https://www.pbinfo.ro/probleme/11/vocale
- [ ] #136 **numararecuvinte** — ușoară
  https://www.pbinfo.ro/probleme/136/numararecuvinte
- [ ] #973 **cuvinte1** — ușoară
  https://www.pbinfo.ro/probleme/973/cuvinte1
- [ ] #199 **eliminaresubsir** — medie
  https://www.pbinfo.ro/probleme/199/eliminaresubsir

Temă pentru acasă (pbinfo)

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

- [ ] #903 **cezar** — ușoară
  https://www.pbinfo.ro/probleme/903/cezar
- [ ] #138 **frecventa** — ușoară
  https://www.pbinfo.ro/probleme/138/frecventa
- [ ] #2828 **acronim** — ușoară
  https://www.pbinfo.ro/probleme/2828/acronim
- [ ] #3112 **bac-cuvinte** — ușoară
  https://www.pbinfo.ro/probleme/3112/bac-cuvinte