

## Capitolul 6 – Prelucrarea șirurilor de caractere în C++

### Funcții suplimentare

**Funcția rfind():** caută ultima apariție a unui șir în alt șir.

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

int main() {
    string text = "Ana are mere si Ana bea apa.";
    size_t poz = text.rfind("Ana");

    if (poz != string::npos) {
        cout << "Ultima aparitie a cuvantului 'Ana' este la pozitia: " << poz;
    } else {
        cout << "Cuvantul nu a fost gasit.";
    }

    return 0;
}
```

#### Explicație:

- rfind() caută de la final spre început.
- Returnează poziția primei litere din ultima apariție a subșirului.
- Dacă nu găsește subșirul, returnează string::npos.

---

**Funcția clear():** șterge complet conținutul unui șir.

```
#include <iostream>
```

```
#include <string>
using namespace std;

int main() {
    string mesaj = "Salut!";
    mesaj.clear();
    cout << "Lungimea sirului este: " << mesaj.length(); // 0
    return 0;
}
```

### Explicație:

- După `clear()`, șirul devine gol (`""`).
- Poți verifica cu `empty()` dacă e gol.

---

**Funcția `empty()`:** verifică dacă un șir este gol.

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

int main() {
    string sir = "";
    if (sir.empty()) {
        cout << "Sirul este gol.";
    } else {
        cout << "Sirul NU este gol.";
    }
    return 0;
}
```

### Explicație:

- Returnează true dacă lungimea șirului este 0.

---

**Funcția compare():** compară două șiruri. Returnează:

- 0 dacă sunt egale,
- <0 dacă primul e mai mic,
- 0 dacă primul e mai mare (lexicografic).

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

```
int main() {
    string a = "Ana";
    string b = "Anca";

    int rezultat = a.compare(b);
    cout << "Rezultatul compararii: " << rezultat;

    return 0;
}
```

**Explicație:**

- Compararea se face lexicografic (alfabetic).
- Poți folosi și operatorii <, >, ==, dar compare() oferă mai multe opțiuni.

---

**Funcția substr():** extrage o subsecvență dintr-un șir.

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

```
int main() {
```

```
string text = "Programare C++";
string sub = text.substr(11, 3); // incepe la poz. 11 si ia 3 caractere
cout << "Subsir: " << sub; // C++
return 0;
}
```

### Explicație:

- substr(poz, lungime) extrage o porțiune din șir.
- Dacă lungimea este prea mare, va lua până la final.

---

**Funcția swap():** face schimb de conținut între două șiruri.

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

int main() {
    string a = "mere";
    string b = "pere";
    a.swap(b);
    cout << "a = " << a << ", b = " << b;
    return 0;
}
```

### Explicație:

- După swap(), valorile șirurilor sunt inversate.

---

**Funcțiile: insert(), erase(), replace()**

```
#include <iostream>
```

```
#include <string>
using namespace std;

int main() {
    string text = "Salut lume!";

    text.insert(6, "draga "); // adauga la pozitia 6
    text.erase(0, 6);        // sterge de la 0, 6 caractere
    text.replace(0, 5, "Hello"); // inlocuieste 5 caractere cu "Hello"

    cout << text;
    return 0;
}
```

### Explicație:

- insert(poz, text) inserează text într-un șir.
- erase(poz, lungime) șterge o secvență.
- replace(poz, lungime, text) înlocuiește o secvență.