**2. Operaţii cu şiruri**

1. Declararea unei variabile şir:

**string s1; //şir C++ neiniţializat**

**string s2(“ion”);//şir C++ iniţializat cu şir C**

**string s3(s2); //şir C++ iniţializat cu alt şir**

2.Accesul la caracter:

**//acces la caracter prin indexare**

**s2[0]=’B’;**

**//selecţie subşir de lungime 3 pornind din poz.2**

**string s3=s2.substr(2, 3);//**

**//conversie şir C++ în şir C**

**string nume;**

**FILE\* f=fopen(nume.c\_str(),”r”);**

3.Lungime şir

**s.length(); //numar de caractere din sir**

**s.resize(100,’x’);//schimba dim.la 100 si-l umple cu ‘x’**

**s.empty(); //test sir vid**

4.Atribuire si concatenare

**s=s1; //atribuire de sir**

**s+=s1; //adauga s1 la s**

**s1+s2 //sir format prin concatenare s1 cu s2**

5.Iteratori

**string::iterator i; //declarare iterator**

**s.begin(); //iterator de pornire**

**s.end(); //iteratorul dupa ultimul element**

6.Inserări, ştergeri, înlocuiri

**s.insert(pos,s1); //insereaza s1 in s dupa pos**

**s.remove(st,p); //sterge p caractere din s, dupa st**

**s.replace(st,l,s1);//insereaza s1 in s,inlocuind l car**

7.Comparaţii

**s1==s2, s1!=s2, s1<s2, s1>s2, s1<=s2, s1>=s2**

8.Operaţii de căutare

**s.find(s1); //cauta in s unde incepe s1**

**s.find(s1,pos); //idem, incepand cautarea din pos**

**s.find\_first\_of(s1,pos);//determina prima aparitie a primului caracter din s in s1**

**//pornind din pos**

**s.find\_first\_not\_of(s1,pos);//determina prima aparitie a primului caracter din s**

**// ce nu e in s1 pornind din pos**

9.Intrări / ieşiri

**stream << s; //inserare s in stream**

**stream >> s; //extractie s din stream**

**getline(stream, s, sep);//citeste o linie din stream in s, terminata prin sep**

O variabilă **string** poate primi prin atribuire:

* un alt string
* un literal
* un caracter
* un tablou de caractere stil C

Funcţiile **begin()** şi **end()** întorc iteratori cu acces direct de început şi sfârşit de şir.

Caracteristicile setului de caractere folosit(ASCII sau UNICODE) sunt descrise în structura **char\_traits**.

Prin indexare se asigură acces la un caracter din string fără verificarea domeniului; accesul cu verificare este asigurat de funcţia **at()**.