**C++ da matnlar bilan ishlash**

C++ da satrlar bilan ishlashni qulaylashtirish uchun string sinfi kiritilgan. string sinfi satrlarida  satr oxirini  '\0' belgisi  belgilamaydi.  
Standart kutubxonadagi string sinfidan foydalanish uchun <string> sarlavha faylini dasturga qo'shish kerak.  
C tilida kompilyatorlarda <cstring.h> yoki <bstring.h> sarlavha faylini qo'shish kerak bo'ladi. Oddiy usuldagi satrlar bilan ishlash uchun esa, <string.h> sarlavha fayli qo'shiladi.  
Eng afzali,  o'zingiz ishlatayotgan kompilyator bilan yaxshilab tanishib chiqing.  
Satrlar bilan ishlovchi  asosiy funksiyalar  bilan tanishib chiqamiz.

Satr хususiyatlarini aniqlash uchun quyidagi funksiyalardan  
foydalanish mumkin:  
unsigned int size() const;        // satr o’lchami  
unsigned int length() const;      // satr elеmеntlar sоni  
unsigned int max\_size() const;    // satrning maksimal uzunligi  
unsigned int capacity() const;    // satr egallagan хоtira hajmi  
bool empty() const;               // satrning bo’shligini  
aniqlash

Satrning uzunligini aniqlash uchun length() yoki size()  
funksiyalaridan foydalanish mumkin.

Misol: Satr uzunligini aniqlash dasturi

**#include <iostream.h>**  
**#include <string>**  
**int main()**  
**{**  
**string s;**  
**cout << "Satr kiriting" << endl;**  
**getline(cin, s);**  
**cout << "Siz kiritgan satr " << s.length() << " ta belgidan iborat";**  
**cout << "Siz kiritgan satr " << s.size() << " ta belgidan iborat";**  
**system ("pause");**  
**return 0;**  
**}**

Satr uzunligini o'zgartirish uchun resize funksiyasidan  foydalaniladi.  
1) void resize ( size\_t n, char c );  
2) void resize ( size\_t n );

**#include <iostream.h>**  
**#include <string>**  
**int main()**  
**{**  
**size\_t n;**  
**string str ("C++ matn kodi so`zi");**  
**cout << str << endl;**  
**// satr uzunligini aniqlash**  
**n = str.size();**  
**// satr uzunligini 2 ta belgiga uzaytirish**  
**str.resize (n + 2, '+');**  
**cout << str << endl;**  
**//satr uzunligini o'zgartirish**  
**str.resize(14);**  
**cout << str << endl;**  
**system ("pause");**  
**return 0;**  
**}**

void clear(); - funksiyasi  satrni tozalash (to'liq o'chirish) uchun ishlatiladi.  
bool empty()  const; - funksiyasi satrni bo'shligini  tekshirish uchun ishlatiladi.  Agar  
satr  bo'sh bo'lsa, true  qiymat  qaytaradi.  
Satrning  biror qismidan nusxa olish  
string& assign ( const string &str );  
Satrga str o'zgaruvchisidagi  satrning  to'liq nusxasini  olish.  
string& assign ( const string& str, size\_t pos, size\_t n );  
Satrga str o'zgaruvchisidagi  satrning  pos o'rindagi belgisidan  boshlab n ta belgi  
nusxasini olish.  
string& assign ( const char\* s, size\_t n );  
string toifasidagi  satrga char toifasidagi  satrning  n ta belgisi  nusxasini  olish.

string  s1, s2, s3;  
s1 = "tami.uz";  
s2.assign(s1); // s2 = "tami.uz"  
s3.assign(s1, 0, 4); // s3 = "tami"  
append funksiyasining assigndan  farqi satrning  davomiga  satr qismining qo'shishidadir.  
string&  append (  const string&  str );  
string& append ( const string& str, size\_t pos, size\_t n );  
string& append ( const char\* s, size\_t n );

Satrdan nusxa olish

**#include <iostream.h>  
#include <string>  
int main()  
{  
string s1, s2, s3;  
s1 = "rdvatami";  
s2.assign(s1, 0, 2); // s2 = "rd"  
s3.assign(s1, 4, 4); // s3 = "tami"  
cout << s1 << endl;  
cout << s2 << endl;  
cout << s3 << endl;  
s1 = s3 + s2; // s1 = "tamird"  
cout << s1 << endl;  
s2.append(s3); // s2 = " rdvatami "  
cout << s2 << endl;  
system ("pause");  
return 0;  
}**

char toifasidagi satrni string toifasiga o'tkazish. clear funksiyasiga misol.

**#include <iostream.h>  
#include <string>  
int main()  
{  
char c[100];  
string s = "Boshlang'ich qiymat";  
s.clear(); // s ning qiymati tozalandi  
cout << "Satr kiriting" << endl;  
cin.getline(c, sizeof(c));  
// s satriga char toifasidagi c satrini nusxasini olish  
s.assign(c, strlen(c));  
cout << s << endl;  
system ("pause");  
return 0;  
}**

string toifasidagi  satrni char toifasiga o'tkazish  
string toifasidagi  satrni char toifasiga  o'tkazish uchun c\_str yoki data funksiyalaridan foydalanish mumkin.  
const char  \*  c\_str() const;  
const char \* data() const;  
Bularning bir - biridan farqi, data funksiyasida satr oxiriga  '\0' satr oxiri  belgisi  
qo'shilmaydi.

**#include <iostream.h>  
#include <string>  
#include <string.h>  
int main()  
{  
char c[100];  
string s;  
cout << "Satr kiriting" << endl;  
getline(cin, s);  
strcpy(c, s.c\_str());  
cout << c << endl;  
system ("pause");  
return 0;  
}**

Satrning  biror qismini  o’ chirish  
erase(unsigned int pos=0, unsigned int n=npos);  
erase funksiyasi  satrni  pos o'zgaruvchisida ko'rsatilgan  o'rindan boshlab n ta belgini o'chiradi.  
Agar nechta belgi  o'chirilishi  n ko'rsatilmagan  bo'lsa, pos o'zgaruvchisida  ko'rsatilgan o'rindan boshlab satr oxirigacha  o'chiriladi.  
Agar pos va n ko'rsatilmagan  bo'lsa, satr to'liq o'chiriladi.

**#include <iostream.h>  
#include <string>  
int main()  
{  
string s = "Assalomu alaykum bo'lajak dasturchilar";  
cout << s << endl;  
// 16 - belgidan boshlab 9 ta belgini o'chirish  
s.erase(16, 9);  
cout << s << endl; // Assalomu alaykum dasturchilar - chiqadi  
// s = "Assalomu alaykum dasturchilar";  
// 16 - belgidan boshlab satr oxirigacha o'chirish  
s.erase(16);  
cout << s << endl; // Assalomu alaykum - chiqadi  
s.erase();  
cout << s << endl; // bo'sh satr chiqariladi  
system ("pause");  
return 0;  
}**

Satrni satr orasiga qo'shish  
Biror satrga boshqa satrning  istalgan qismini  qo'shish uchun  insert  funksiyasidan  
foydalaniladi.  
insert(unsigned int  pos1, const string  &str); Satrga pos1 o'rindan boshlab, str satrini  qo'shish. insert(unsigned  int pos1, const string &str, unsigned int pos2,unsigned int n);  
Satrga pos1 o'rindan boshlab, str satrining  pos2 o'rnidan boshlab n ta belgini  qo'shish.  
insert(unsigned int  pos1, const char  \*str, int  n);  
Satrga pos1 o'rindan boshlab, char toifasidagi  satrning  n ta belgisini  qo'shish.

**#include <iostream.h>  
#include <string>  
int main()  
{  
string s = "Assalomu alaykum do'stlar";  
string c = "Merhibon\_va\_muhtarama\_ayollar";  
char p[] = "Salomlashish odobi";  
cout << s << endl;  
// 17 - belgidan boshlab yangi satrni qo'shish  
s.insert(17, "aziz\_");  
cout << s << endl; // Assalomu alaykum aziz do'stlar  
// c satridan 8 - belgidan boshlab 10 ta belgi qo'shish  
s.insert(21, c, 8, 10);  
cout << s << endl; // Assalomu alaykum aziz\_va\_muhtar\_do'stlar  
// char toifasidagi satrdan 13 ta belgini qo'shish  
s.insert(0, p, 13);  
cout << s << endl;  
system ("pause");  
return 0;  
}**

Satr qismini  almashtirish  
Satrning  biror qismini  almashtirish  kerak bo'lsa, replace funksiyasidan  foydalanish  
mumkin.  
replace (unsigned int pos1, unsigned int n1, const string &str);

replace (unsigned int pos1, unsigned int n1, const string & str, unsigned int pos2,  
unsigned int n2);  
replace (unsigned int pos1, unsigned int n1, const char \*str, int n);  
replace funksiyasi insert kabi ishlaydi, faqat qo'shilishi kerak bo'lan satrni pos1 -  
o'rindan boshlab n1 ta belgi  o'rniga qo'shadi.  
2 ta satrni to'la almashtirish  uchun swap funksiyasi  ishlatilad i.

**#include <iostream.h>  
#include <string>  
using namespace std;  
int main()  
{  
string s = "Assalomu alaykum do'stlar";  
string c = "Merhibon va muhtarama ayol";  
cout << s << endl;  
// 17 - belgidan boshlab 5 ta belgi o'rniga c satrni qo'shish  
s.replace(17, 5, c);  
cout << s << endl;  
s.swap(c); // 2 ta satrni to'la almashtirish  
cout << s << endl;  
s.replace(0, 0, c, 0, 17);  
s.erase(25);  
cout << s << endl;  
system ("pause");  
return 0;  
}**