

Simbolių eilutė char

Eilučių analizės funkcijos

Funkcijos prototipas	Paaiškinimai
int strcmp (const char *s1, const char *s2);	<p>Palygina eilutes s1 ir s2. Eilučių simbolių kodai iš eilės lyginami tol, kol randami skirtingi simboliai arba kurios nors eilutės pabaiga (didžiosios ir mažosios raidės – skiriasi).</p> <p>Funkcija išveda:</p> <ul style="list-style-type: none">• 0, jei eilutės sutampa (s1 ==s2);• neigiamą skaičių, jei eilutės s1 kodas mažesnis už eilutės s2;• teigiamą skaičių, jei eilutės s1 kodas didesnis už eilutės s2; <p>Pvz.: char e[] = "medis"; char s[] = "medelis"; if (strcmp (e, s) == 0) cout << "e = s"; else if (strcmp (e, s) < 0) cout << "e < s"; else cout << "e > s";</p> <p style="text-align: right;">Atsakymas: e > s</p>
int stricmp (const char *s1, const char *s2);	<p>Palygina eilutes s1 ir s2. Eilučių simbolių kodai iš eilės lyginami tol, kol randami skirtingi simboliai arba kurios nors eilutės pabaiga (didžiosios ir mažosios raidės – skiriamos).</p> <p>Funkcija išveda:</p> <ul style="list-style-type: none">• 0, jei eilutės sutampa (s1 ==s2);• neigiamą skaičių, jei eilutės s1 kodas mažesnis už eilutės s2;• teigiamą skaičių, jei eilutės s1 kodas didesnis už eilutės s2; <p>Pvz.: char e[] = "medis"; char s[] = "Medis"; if (stricmp (e, s) == 0) cout << "e = s"; else if (stricmp (e, s) < 0) cout << "e < s"; else cout << "e > s";</p> <p style="text-align: right;">Atsakymas: e = s</p>
int strncmp (const char *s1, const char *s2, int m);	<p>Palygina eilučių s1 ir s2 pirmuosius m simbolių (didžiosios ir mažosios raidės – skiriamos).</p> <p>Funkcija išveda:</p> <ul style="list-style-type: none">• 0, jei eilučių dalys (pirmųjų m simbolių) sutampa;• neigiamą skaičių, s1 dalis mažesnė už s2 dalį;• teigiamą skaičių, s1 dalis didesnė už s2 dalį; <p>Pvz.: char e[] = "medis"; char s[] = "medelis"; if (strncmp (e, s, 3) == 0) cout << "e = s"; else if (strncmp (e, s, 3) < 0) cout << "e < s"; else cout << "e > s";</p> <p style="text-align: right;">Atsakymas: e = s</p>
int strnicmp (const char *s1, const char *s2, int m);	<p>Palygina eilučių s1 ir s2 pirmuosius m simbolių (didžiosios ir mažosios raidės – neskiriamos).</p> <p>Funkcija išveda:</p> <ul style="list-style-type: none">• 0, jei eilučių dalys (pirmųjų m simbolių) sutampa;• neigiamą skaičių, s1 dalis mažesnė už s2 dalį;• teigiamą skaičių, s1 dalis didesnė už s2 dalį; <p>Pvz.: char e[] = "medis"; char s[] = "Medelis"; if (strnicmp (e, s, 3) == 0) cout << "e = s"; else if (strnicmp (e, s, 3) < 0) cout << "e < s"; else cout << "e > s";</p> <p style="text-align: right;">Atsakymas: e = s</p>

int strlen (char *s);	Apskaičiuoja ir grąžina eilutės s (be nulinio simbolio) ilgį. Pvz.: char e[12] = "C++ kalba"; int n = strlen (e); Atsakymas: 9 (tarpas taip pat simbolis)
------------------------------	---

Eilučių konvertavimo ir kopijavimo funkcijos

Funkcijos prototipas	Paaiškinimai
double atof (const char *s);	Konvertuoja eilutės S pirmuosius simbolius į realų skaičių, pateikiamą standartine išraiška, ir jį grąžina. Pvz.: char *S = "45.23e3abc"; double d = atof (S); Atsakymas: 45.230
int atoi (const char *s);	Konvertuoja eilutės S pirmuosius simbolius į sveiką skaičių ir jį grąžina. Pvz.: char *S = "45.23e3abc"; int n = atoi (S); Atsakymas: 45
long atol (const char *s);	Konvertuoja eilutės S pirmuosius simbolius į ilgąjį sveikąjį skaičių ir jį grąžina. Pvz.: char *S = "45.23e3abc"; long n = atol (S); Atsakymas: 45
char *strcpy (char *d, const char *s);	Kopijuoja eilutę S ir grąžina rodyklę į eilutę d. Pvz.: char e[12]; strcpy(e, "C++ kalba"); Atsakymas: eilutės e reikšmė: "C++ kalba"
char *strncpy (char *d, char *s, unsigned m);	Kopijuoja m simbolių iš eilutės S į eilutę d ir grąžina rodyklę į eilutę d (eilutės pabaigos simbolio (\0) neįrašo). Pvz.: char e[12]; char S[] = "C++ kalba"; strncpy(e, S, 3); e[3] = '\0'; Atsakymas: eilutės e reikšmė: "C++"
char strlwr (const char *s);	Perrašo visą eilutę mažosiomis raidėmis ir grąžina rodyklę į pakeistą eilutę S (mažųjų raidžių ir kitų simbolių nekeičia). Pvz.: char e[] = "C++ Builder"; strlwr (e); Atsakymas: "c++ builder"
charstrup (const char *s);	Perrašo visą eilutę didžiosiomis raidėmis ir grąžina rodyklę į pakeistą eilutę S (didžiųjų raidžių ir kitų simbolių nekeičia). Pvz.: char e[] = "C++ Builder"; strup (e); Atsakymas: "C++ BUILDER"

Eilučių modifikavimo funkcijos

Funkcijos prototipas	Paaiškinimai
char *strcat (char *d, const char *s);	Prie eilutės d prijungia eilutę s ir grąžina rodyklę į sujungtą eilutę d. Pvz.: char e[12] = "C++ "; char s[] = "kalba "; strcat (e); Atsakymas: eilutės e nauja reikšmė: "C++ kalba"

char *strncat (char *d, const char *s, unsigned m);	Prie eilutės d prijungia m simbolių iš eilutės s ir grąžina rodyklę į sujungtą eilutę d. Pvz.: char e[15] = "C++ "; char s[] = "programavimas "; strncat (e, s, 8); Atsakymas: eilutės e nauja reikšmė: "C++ programa"
char *strnset (char *s, char *c, unsigned m);	Užpildo S eilutės m elementų simboliu c ir grąžina rodyklę į eilutę S. Pvz.: char e[] = "C++ Builder "; strnset (e, '*', 4); Atsakymas: eilutės e nauja reikšmė: "****Builder"
char *strset (char *s, char *c);	Užpildo S eilutę simboliu c ir grąžina rodyklę į eilutę S. Pvz.: char e[] = "C++ Builder "; strset (e, '*'); Atsakymas: eilutės e nauja reikšmė: "*****"

Klasės string funkcijos

Funkcijos prototipas	Paaiškinimai
size_type max_size() const;	Grąžina elementų, kurie turi būti eilutėje skaičių. Pvz.: string s("C++ kalba"); int n = s.max.size(); Atsakymas: 4294967281
size_type length() const;	Grąžina eilutės ilgį. Pvz.: string s("C++ kalba"); int n = s.length(); Atsakymas: 9
const char * c_str();	Grąžina rodyklę-konstantą į C eilutę, kurios reikšmė sutampa su eilute, kuriai taikoma ši funkcija. Naudojama failų atidarymo sakiniuose. Pvz.: string s = "Duomenys.txt"; ifstream fd(s.c_str()); Rezultatas - eilutė s, konvertuota į C eilutę
string substr(size_type k, size_type n = npos);	Iš eilutės, kuriai taikoma ši funkcija, kopijuojama n simbolių, pradedant k-uoju. Pvz.: string s = "Programavimo kalba C++"; string s1 = "Matematika"; s1 = s.substr(13, 5); Rezultatas – „kalba“