

Základy programování v C++ 11. cvičení

Zuzana Petříčková

12. listopadu 2018

Přehled

1 Textové řetězce v C/C++

Textové řetězce v C/C++

Textové řetězce v jazyce C

```
char retezec [delka];
```

- v jazyce C má význam textového řetězce pole znaků
- textový řetězec musí být ukončený znakem `'\0'`
→ pole musí být alespon o 1 delší než je uložený řetězec

Inicializace

```
char s[10] = "text"; //prekladac automaticky doplni '\0'  
char s1[10] = {'t', 'e', 'x', 't', '\0'};
```

- s textovým řetězcem se pak zachází stejně jako s libovolným polem
- Funkce pro práci s textovými řetězci v hlavičkovém souboru **string.h (cstring)**

Textové řetězce v C/C++

Práce s textovými řetězci

- přístup k prvku pole pomocí indexovacího operátoru []
- tisk na konzoli pomocí **cout**
- načtení z konzole: **cin**, **cin.getline()**

```
char s[256], t[256], u[256];
cin >> s; // nacte text po prvni bily znak (' ', '\t', '\n')
cin.getline(t, 255); // nacte text do konce radku (znak '\n')
cin.getline(u, 255, ';'); //nacte text az po znak ';'
```

```
cout << s << endl << t << endl << u << endl;
```

```
// t = s // NELZE
// t = "pes"; // NELZE
// if (t == s) // NELZE
```

```
s[0] = '\0'; // vyprazdnim retezec
cout << s << endl;
```

Textové řetězce v C/C++

Příklady

- Napište vlastní funkci **int delka_retezce(const char s[])**, která spočítá délku řetězce.
- Napište vlastní funkci **void kopiruj_retezec(char t[], const char s[])**, která zkopiruje obsah řetězce **s** do řetězce **t**.

Textové řetězce v C/C++

Příklad: délka řetězce

```
int delka_retezce(const char s[])  
{  
    int n = 0;  
    while (s[n] != '\0')  
        n++;  
    return n;  
}
```

- knihovní funkce **strlen()**

```
char s1[10] = "text";  
cout << delka_retezce(s1) << endl;  
cout << strlen(s1) << endl;
```

Textové řetězce v C/C++

Příklad: kopírování řetězců

```
void kopiruj_retezec(char t[], const char s[])  
{  
    int len = delka_retezce(s); // strlen(s);  
    for (int i = 0; i <= len; i++)  
        t[i] = s[i];  
}
```

- knihovní funkce **strcpy_s(kam,pocet_bytu,odkud)**

```
char s[10] = "text", t[10], u[10];  
kopiruj_retezce(t, s);  
strcpy_s(u, sizeof(s), s);  
cout << endl << t << endl << u << endl;  
  
// lexikograficke porovnaní  
cout << strcmp(t, u) << endl;
```

Textové řetězce v C/C++

Velikost datového typu v bytech

- funkce **sizeof()**

```
...
```

```
const int n = 3;
```

```
int a[n];
```

```
int x;
```

```
int velPole = sizeof(a);
```

```
int velInt = sizeof(int);
```

```
int velX = sizeof(x);
```

```
int pocetPrvku = velPole/velInt;
```

```
...
```


Délka statického pole

Funkce **sizeof()** musí "bezprostředně následovat" po definici pole:

```
...
const int n = 10;
char a[n] = "muj_text";

int velPole = sizeof(a); // velikost pole v bytech
int velChar = sizeof(char); // velikost char-u v bytech
int pocetPrvku = velPole/velChar;
cout << "Delka_pole_je" << pocetPrvku << endl;
cout << "Delka_textu_je" << strlen(a) << endl;
...
```

Funkce **sizeof()** volaná ve funkci nedá správný výsledek:

```
void vypis_delku(const char a[])
{
    cout << "Delka_textu_je" << strlen(a) << endl;
    cout << "Delka_pole_NENI_" << sizeof(a)/sizeof(char);
}
```

Funkce pro práci s textovými řetězci

Výběr funkcí z knihovny `string.h` :

- `strlen()` délka řetězce
- `strcpy()` kopírování obsahu řetězce z pole do pole
- `strncpy()` kopírování řetězce délky `n`
- `strchr()` vyhledání prvního výskytu znaku
- `strrchr()` vyhledání posledního výskytu znaku
- `strstr()` vyhledání prvního výskytu podřetězce
- `strcmp()` lexikografické porovnání řetězců
- `strcat()` připojení řetězce k řetězci

...

Textové řetězce v C++

- navíc třídy pro reprezentaci řetězců:
 - **string** ... řetězce z úzkých znaků
 - **wstring** ... řetězce z Unicode znaků
- hlavičkový soubor **string**
- Výhody:
 - jednodušší práce (chová se podobně jako struktura)
 - řada metod a operátorů:
 - = pro přiřazení
 - [] pro indexaci
 - + a += pro zřetězení
 - <, <=, >, >=, ==, != pro lexikografické porovnání

Textové řetězce v C++

Datový typ string

```
string s1 = "text", s2 = { 't','e','x','t','\0' };  
string s3, s4, s5;
```

```
int delka = s1.length();
```

```
// nacistani a vypis  
cin >> s3;  
getline(cin, s4);  
getline(cin, s5, '.');  
cout << s3 << endl;
```

```
// indexace  
cout << s1[0] << endl;  
cout << s1.at(0) << endl;
```

Textové řetězce v C++

Datový typ string

```
string s1 = "text", s2 = { 't','e','x','t','\0' };
string s3, s4, s5;
...
s5 += s3 + "_" + s4;           // konkatence
s3 = s1;                       // kopirovani
s3 = "jiny_text";             // kopirovani
if (s3 < s4)                   // porovnaní
    cout << "je_lexikograficky_mensi" << endl;
if (s3 == s4)                  // porovnaní
    cout << "retezce_jsou_identicke" << endl;
char x[20];
kopiruj_retezec(x,s3.data());  // prevod na pole charu
kopiruj_retezec(x,s3.c_str()); // prevod na pole charu
...
```

Textové řetězce v C/C++

Příklady

- Napište funkci **string na_velka(string s)** resp. **void na_velka(char s[])**, která změní všechna malá písmena v řetězci na velká.
- Napište funkci **string zasifruj_substituci(string zprava, int posun)** resp. **void zasifruj_substituci(char s[], int posun)**, která zašifruje text **s** pomocí substituční (Cézarovy) šifry (posun písmen o **posun** znaků).
- Napište vlastní funkci **int do_desitkove(string s, int zaklad)**, která převede číslo v soustavě o základu **zaklad** zadané jako textový řetězec na číslo v desítkové soustavě (analogie knihovní funkce **stoi()**).

Textové řetězce v C/C++

Další příklady na procvičení

- Napište funkci **bool je_palindrom(string text)**, která vrátí true, pokud je text palindromem (tj. čte se stejně popředu jako pozpátku, př. "kobyLAMamalybok").
- Napište funkci **string zasifruj_substituci(string zprava, string vzor)**, která zašifruje text **s** pomocí substituce zadané řetězcem **vzor**. (např. `z = zasifruj_substituci("ABECEDA 1", "KLMN")`); `z` bude rovno "KLEMENK 1")

Textové řetězce v C/C++

Příklad (měli jsme)

- Funkce postupně načte celý řádek textu a změní v něm malá písmena na velká. Výsledek vypíše na konzoli.

```
void cely_radek ()
{
    cout << "Zadej_radek_textu:" << endl;
    char znak;
    do
    {
        znak = getchar();
        cout << na_velke(znak);
    } while (znak != '\n');
}
```


Textové řetězce v C/C++

Příklad (pomocí řetězců)

- Funkce postupně načte celý řádek textu a změní v něm malá písmena na velká. Výsledek vypíše na konzoli.

```
void cely_radek ()  
{  
    string s;  
    getline(cin, s);  
    cout << na_velka(s) << endl;  
}
```

- pro pole znaků:

```
void cely_radek1 ()  
{  
    char s[256];;  
    cin.getline(s, 255);  
    na_velka(s);  
    cout << s << endl;  
}
```