

Řetězec – string

- řetězcem je míněno pole typu char, jehož posledním znakem je znak `'\0'` (nula)
- koncový znak je důležitý, protože s ním pracují všechny funkce pro řetězce
- pro práci s řetězcí existuje knihovna funkcí (existuje také knihovna pro string jako struktura, třída – těm se zde nevěnujeme)
- s řetězcem se pracuje jako s běžným polem

definice pole dané délky s inicializací pomocí znaků (použita pouze část pole):

```
char Ret1[10]= {'Z','d','a','r','\0'};
```

definice pole pomocí řetězce. Znak `'\0'` je přidán automaticky. Délku určí překladač tak aby se řetězec (včetně ukončovacího znaku) vešel:

```
char Ret2[ ]= "Zdar";
```

`Ret2 = "Zdar";` // nelze - mimo definici již musíme s polem pracovat po prvcích

```
Ret2[0]='Z'; Ret2[1]='d'; Ret2[2]='a'; Ret2[3]='r'; Ret2[4]='\0';
```

Napište funkci pro zjištění délky řetězce.

```
// vrací délku řetězce bez zakončovacího znaku (terminated null)
// Jako ve standardní knihovně
int Delka1(char *retez) // pomocí operátoru indexovaného přístupu []
{ int i=0;
  while (retez[i++] != '\0') ;
  return i-1;
}
```

```
int Delka2(char *retez) // pomocí ukazatelové aritmetiky
{
  char * pom=retez;
  while ( * (pom++) != '\0') ;
  // použijeme ukazatel k přístupu a posuneme se na vedlejší prvek
  return (pom – retez); // rozdíl ukazatelů udává počet prvků mezi nimi
}
```

```
char Pole[]="asdf fghj kldori etrt";
int d1 = DelkaX(Pole); // testujeme celé pole (X je 1 nebo 2 – výsledek je stejný)
int d2 = DelkaX(Pole+3); // testujeme pole bez prvních znaků
```

Knihovní funkce pro práci s řetězci

- jsou v knihovně string.h
- většina funkcí nekontroluje, zda je v cílovém řetězci dostatek místa (!!!)
- strlen(ret) vrací délku **bez** ukončovacího znaku (!!!)
- strcpy(cil, zdroj) – kopíruje řetězec včetně ukončení
- strcat(cil,zdroj) – připojí zdroj za původní řetězec
- strcmp(ret1,ter2) – porovná dva řetězce podle „velikosti“ (výsledek -1,0,1)
- strchr – hledá první výskyt znaku v řetězci
- strstr – vyhledá podřetězec v řetězci
- strncpy(cíl, zdroj, počet), strncat, strncmp – pracuje jen s uvedeným počtem znaků

+ „bezpečné“ varianty

Předchozí funkce pracují s typem char, který je dán hodnotou 0-255. Znaky definuje tzv. ASCII tabulka.

Pro UNICODE knihovna uchar.h, wchar.h.

Rozšířený bitový typ char16_t, char32_t.