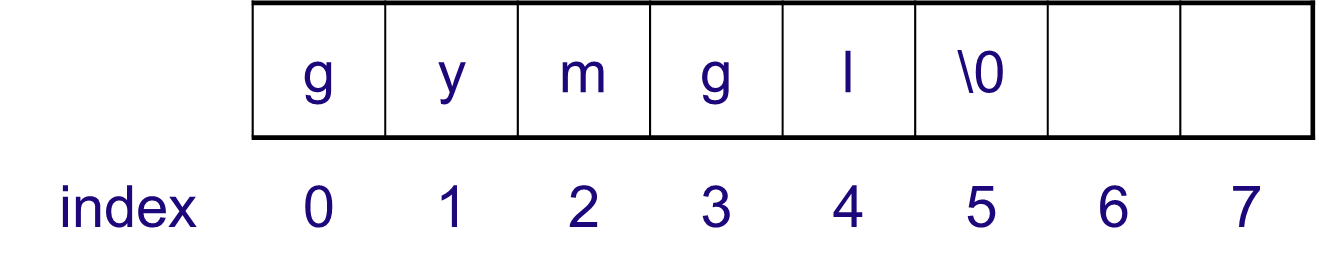
**Reťazce**

**** Reťazec znakov je zvláštnym druhom jednorozmerného **poľa**. Je zložený z prvkov typu char. Reťazce sú základom textových a databázových úborov.  Reťazcové konštanty zapisujeme v programe ako postupnosť znakov uzavretých medzi úvodzovkami. V pamäti je reťazec ukončený znakom  '\0', ale pri vypisovaní to nepoznať, lebo tento znak je "neviditeľný". Naopak, v prípade, že omylom zmažeme tento znak, programy pracujúce s reťazcom, dovtedy budú hľadať koniec reťazca (stringu), pokiaľ náhodne nenarazia na najbližší znak  '\0'.   (aj prázdne znaky zaberajú miesto v pamäti počítača !)

Reťazce môžeme spracovávať pomocou polí a pomocou pointrov (smerníkov)

Deklarácia reťazca, resp. poľa je podobná deklarácii jednoduchej premennej s tým rozdielom, že za menom premennej uvedieme v hranatých zátvorkách počet prvkov poľa.  
     dátový\_typ meno\_poľa[rozmer];

* dátový\_typ všeobecne určuje z akých prvkov bude pole zložené - reťazce (stringy) pozostávajú z *char*
* meno\_poľa tvorí identifikátor premennej typu pole
* rozmer je počet prvkov z ktorých sa pole skladá. Je to kladné číslo.

char meno[20];

Čo vieme zistiť z deklarácie?

deklarácia poľa 20 - tich znakov

rozmer poľa je 20 prvkov

veľkosť 1 prvku poľa 1B, teda veľkosť poľa 20B (byte)

index prvého prvku poľa je 0 a posledného 19 (posledný znak musí byť  '\0' t.j. údaj má max. 19 znakov 0-18)

meno poľa je meno, pole je typu char

V prípade stringov nahradia zložené zátvorky dvojité úvodzovky. Za platnými znakmi

string sa automaticky doplní koncový znak '\0'. Pozor na číslo v hranatých zátvorkách. Koncový znak musí byť súčasťou stringu (započítava sa do počtu znakov !).

char meno[20]="gymgl";

je to isté ako:

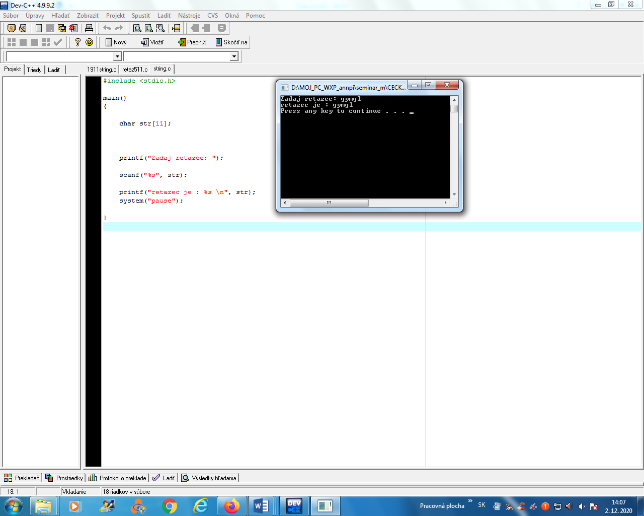
char meno[20]={ 'g','y','m','g' ,'l' \0' };

alebo

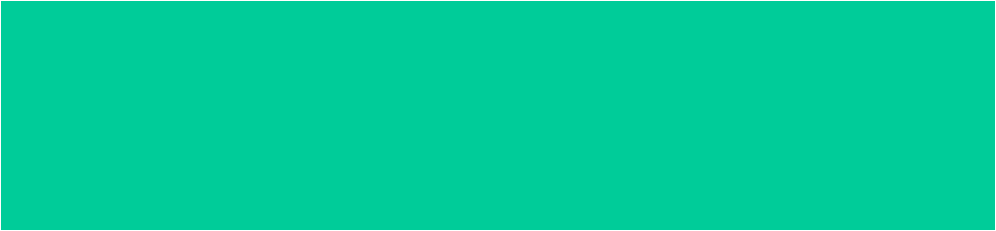
char meno[ ]="gymgl"; // v tomto prípade vzniklo pole meno[6]

**Reťazce**

* reťazce sú v jazyku C reprezentované ako polia znakov (písmená abecedy, čísla,

**char s[20];**

**char s[ ] = ”gymgl”;**

toto pole bude obsahovať 6 znakov

**g y m g l \0**

* reťazce sú ukončené znakom \0

**#include <stdio.h>**

**main()**

**{ char str[11];**

**printf("Zadaj retazec: ");**

**scanf("%s", str);**

**printf("retazec je : %s \n", str);**

**system("pause");**

**}**

**Práca s reťazcami**

načítanie reťazca

**scanf ( ” %s ”, s );** **//bez &** načíta reťazec po biely znak

**gets ( s );**

načíta všetko po ENTER

vypísanie reťazca funkciou **printf ( ” %s ”, s ); puts ( s );**

ďalšie príkazy: fscanf ( ), fprintf ( ), fgets ( ), fputs ( )

**Práca s reťazcami**

knižnica pre prácu s reťazcami <string.h>

dĺžka reťazca

**int strlen(char \*s);**

kopírovanie reťazca

**char strcpy(char \*kam, char \*co);**

spájanie reťazcov

**char strcat(char \*s1, char \*s2);**

hľadanie znaku v reťazci

**char strchr(char \*s, char c);**

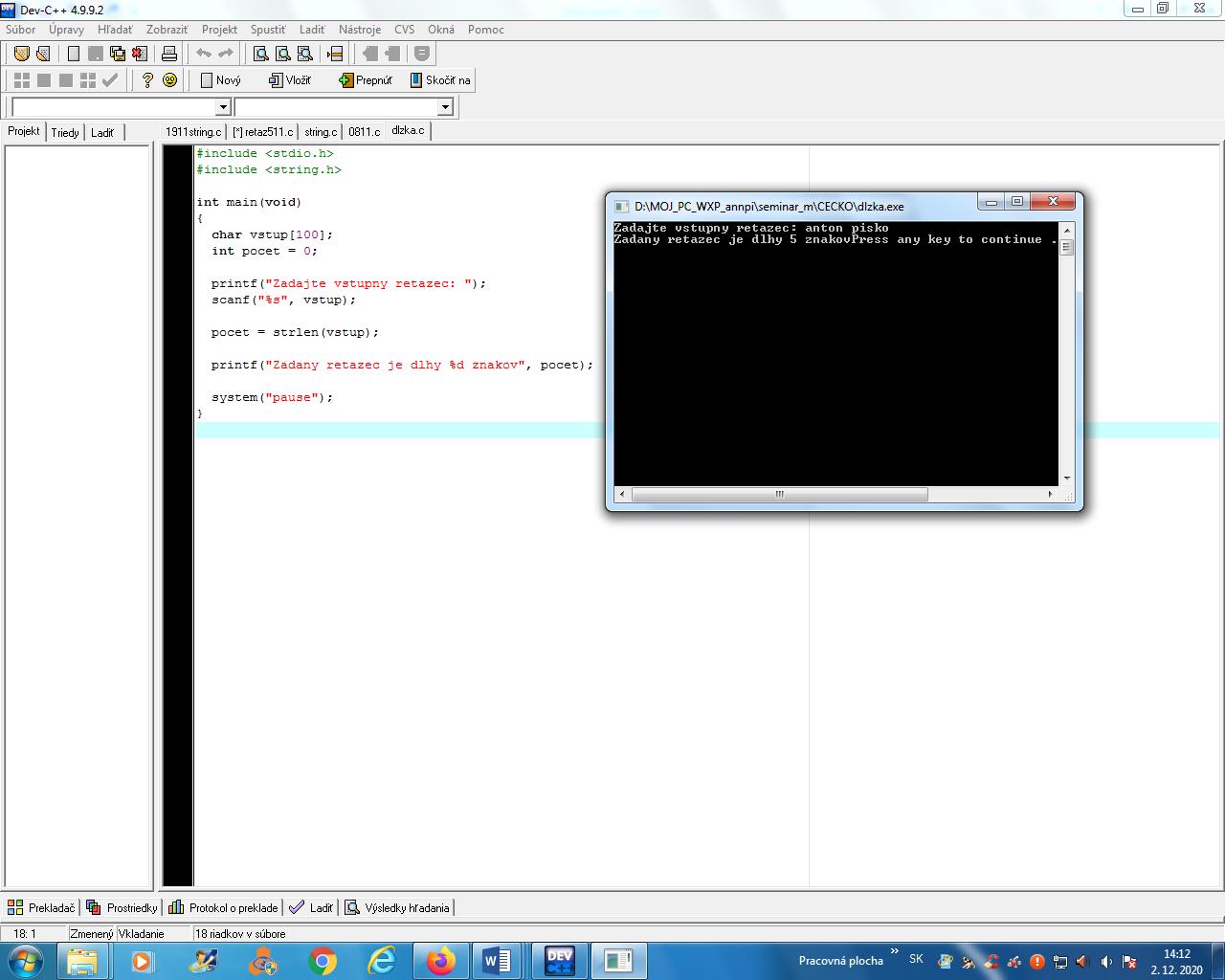
porovnanie dvoch reťazcov

**int strcmp(char \*s1, char \*s2);**

hľadanie podreťazca v reťazci

**char strstr(char \*kde, char \*co);**

**príklad na dĺžku reťazca**



#include <stdio.h>

#include <string.h>

int main(void)

{

char vstup[100];

int pocet = 0;

printf("Zadajte vstupny retazec: ");

scanf("%s", vstup);

pocet = strlen(vstup);

printf("Zadany retazec je dlhy %d znakov", pocet);

system("pause");

}

**Práca s reťazcami**

prevod reťazcov na čísla <stdlib.h>

celé číslo

int **atoi(char \*string);**

celé číslo typu long

long **atol(char \*string);**

reálne číslo

double **atof(char \*string);**

**Práca s reťazcami**

funkcie **sprintf** a **sscanf**

pracujú rovnako ako printf a scanf

ich výstup nie je vytlačený alebo načítaný zo štandardných zariadení pre vstup a výstup ale do/z nami zadaného reťazca

* pre prácu s viac ako 1 byte znakmi (UNICODE) podľa normy ”C99”

( 1 byte = 8 bitov) existujú knižnice # include <ctype.h>

ASCII znaky UNICODE

znaky <ctype.h> <wctype.h>

reťazce <string.h> <wchar.h>

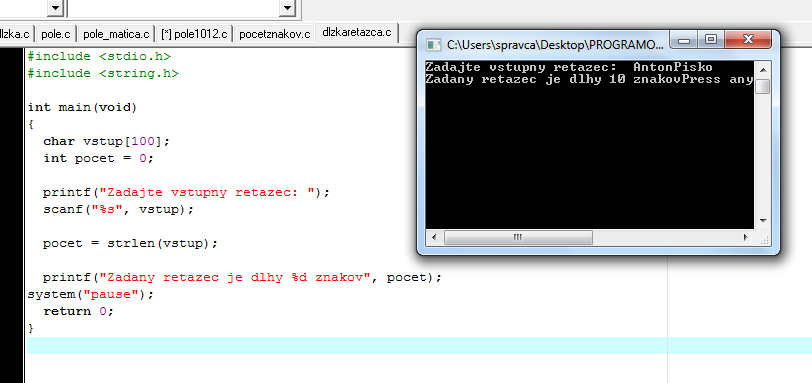
zdroje:

<https://spseke.sk/tutor/projekt/c_jazyk_string.htm#parmain>

<https://spseke.sk/tutor/projekt/c_jazyk_dato.htm#jednopole>

<http://people.tuke.sk/igor.podlubny/C/Kap7.htm>

**Štandardné funkcie pre prácu s reťazcom**



**Zistenie dĺžky reťazca**

#include <stdio.h>

#include <string.h>

int main(void)

{

char vstup[100];

int pocet = 0;

printf("Zadajte vstupny retazec: ");

scanf("%s", vstup);

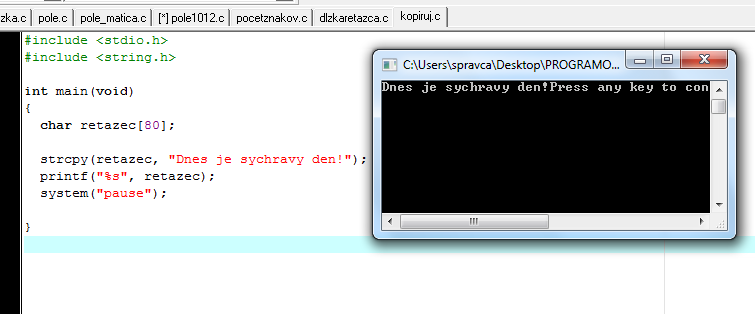
pocet = strlen(vstup);

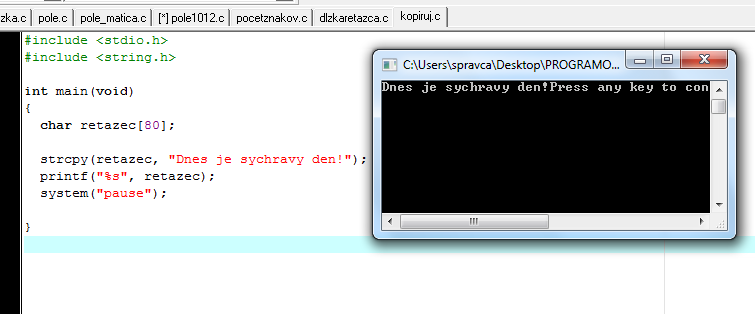
printf("Zadany retazec je dlhy %d znakov", pocet);

return 0;

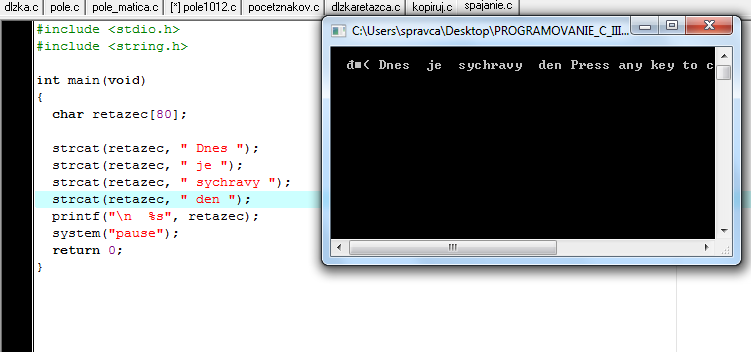
}

**Kopírovanie reťazcov**

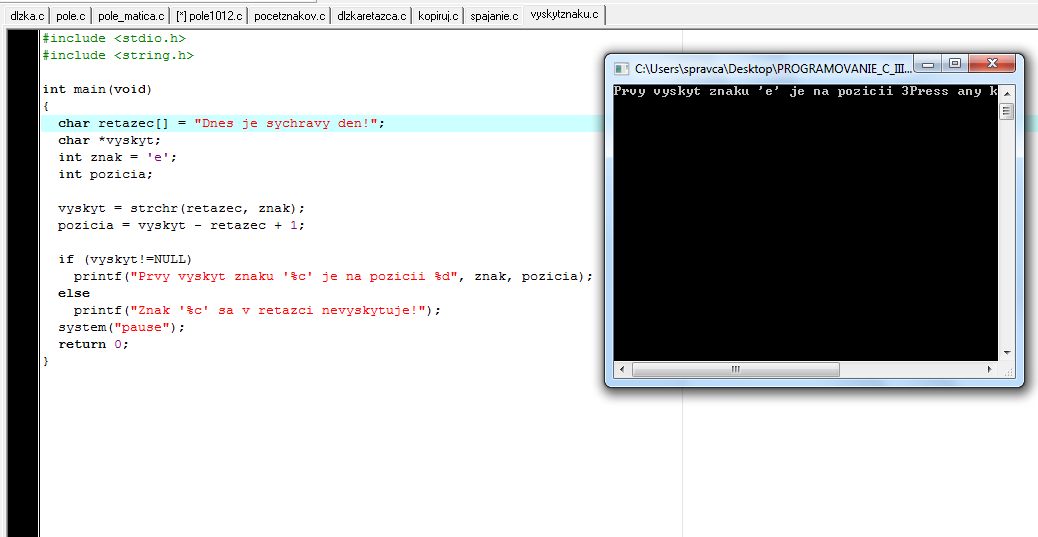




**Spájanie reťazcov**



**Hľadanie znaku v reťazci**



**Porovnávanie dvoch reťazcov**

#include <stdio.h>

#include <string.h>

int main(void)

{

char retazec1[] = "Dnes je sychravy den!";

char retazec2[] = "Dnes je sychravy den !";

int porovnanie;

porovnanie = strcmp(retazec1, retazec2);

if (porovnanie == 0)

printf("Retazce su totozne.");

else

printf("Retazce nie su totozne.");

return 0;

}

**Hľadanie reťazca v reťazci**

#include <stdio.h>

#include <string.h>

int main(void)

{

char retazec[] = "Dnes je skvely den!";

char hladaj[] = "je";

if (strstr(retazec, hladaj)!=NULL)

printf("Hladany retazec bol najdeny.");

else

printf("Retazec nebol najdeny.");

return 0;

}

**Konverzia reťazca na číslo**

#include <stdio.h>

#include <string.h>

#include <stdlib.h>

int main(void)

{

char retazec1[] = "V kosiari je 128 oviec.";

char retazec2[] = "128 oviec.";

int pocet;

pocet = atoi(retazec1);

printf("V kosiari je %d oviec\n", pocet);

pocet = atoi(retazec2);

printf("V kosiari je %d oviec\n", pocet);

return 0;

}

[http://www2.fiit.stuba.sk/~andrejko/alpr/cvicenie06.html#](http://www2.fiit.stuba.sk/~andrejko/alpr/cvicenie06.html)