

%% rosirujici drobnosti programovani

S = char(65) %konvertuje nezaporna cisla do pole znaku, vystupem je char  
%typ z hodnoty 0-65535

%nachystame si pole retezcu string (string typ)

C1=["maximum","minimum","průměrná hodnota","střední hodnota"]; %4x string

%nachystame si pole retezcu string (cell typ, zde jsou vsechny typy string)

%cell je typ promene (ulozneho prostoru), která muze obsahovat temer

%libovolny typ v jednom poli naraz, nemusime tedy mit pole stejných typů

%(mimo ramec tohoto kurzu)

C2={'maximum','minimum','průměrná hodnota','střední hodnota'}; %4x cell se string

Schar1=char(C1) %prevedeme cele pole naraz na char znaky 1x16x4 char

Schar2=char(C2) %prevedeme cele pole naraz na char znaky 4x16 char

Schar1(:,2:4,3) %vyber jen konkretních znaku 2 az 4 znak 3tiho retezce

Schar2(3,2:4) %vyber jen konkretních znaku 2 az 4 znak 3tiho retezce

Schar1(:,1,1:3) %vyber jen jednoho prvního znaku z retezcu 1 az 3

Schar2(1:3,1) %vyber jen jednoho prvního znaku z retezcu 1 az 3

%% char pokračovani

p=65; %ulozeni ASCII kodu do A

char(p) %prevedeni ASCII kodu na znak char

T=[65:127]; %nacteme si znaky ASCII od 65 do 127

s=char(T) %prevedeme cely vektor T znaku naraz do char

s(1:3) %vypise znaky ASCII na pozici 1 az 3 promene s

s(32+1:32+3) %vypise znaky ASCII na pozici 33 az 35 promene s

%% testy

A=65; %ASCII znak pismene A

%pokud je zadana hodnota to na co se ptame, je vystupem log 1,

%jinak bude vystupem logicka 0 tedy ze to pravda neni

ischar('A') %test zda 'A' je znak, ano je to znak

ischar(A) %test zda A je to znak, toto neni znak ale promena s 65 to znak neni

ischar(s) %test zda s je znak, ano je to znak (pole znaku v tomto pripade)

isnumeric(A) %test zda A je cislo (numericka hodnota) toto 65 bude to pravda

isnumeric(s) %test zda s je cislo, toto neni pravda je to char

isnumeric(pi) %test zda A je cislo toto je 3.14... bude to pravda

islogical(true) %test zda true je logicka hodnota, toto BUDE pravda

islogical(1) %test zda true je logicka hodnota, toto NEBUDE pravda je to cislo

E1={}; %zadani prazdne hodnoty

E2=char.empty; %take zadani prazdne hodnoty

E3=5;

%typicky priklad je pri input, kdy se stiskne jen enter bez zadani hodnoty

isempty(E1) %test zda je promenna prazdana (bez hodnoty), ano je prazdna

isempty(E2) %test zda je promenna prazdana (bez hodnoty), ano je prazdna

isempty(A) %test zda je promenna prazdana (bez hodnoty), neni prazdna

%isempty(E) %toto způsobi chybu, promenna neexistuje, ne ze by byla prazdna

%% porovnávání

```
s1=s(1:3); %budou to znaky ABC
s2=s(33:35); %budou to znaky abc
s1==s2 %porovna se POZNACICH, zde nebudou stejne vystup je 0 0 0
'fabio'=='fabia' %opet po znacich, vystup je 1 1 1 1 0
%'ANO'=='NE' %toto zpusobi chybu jelikoz pole NEJSOU STEJNE DLOUHE

%% porovnavani pomoci strcmp
%porovna se bez ohledu na nutnost delku retezcu !! vytup je jen jedna
%hodnota log 1 ci 0
strcmp('ANO','NE') %test zda jsou retezce stejne, nejsou :o), tedy log 0

%pokud chceme testovat jen cast pouzijeme jinou funkci strncmp s n uprostred
vystup = strncmp(s1,s2,2) %porovna v techto retezcich jen prvni 2 znaky,
%vystupem je jen jedna logicka hodnota

%% spojovani retezcu pomoci strcat
%spoji dva retezce char dohromady bez mezery mezi nimi

strcat('cerno','bila',' kocka')

%% vystup pomoci sprintf

%prakticky jen vse tozadne co pro funkci fprintf s VYJIMKOU vystupu
%vystup u sprintf NENI vystup do I/O proudu, jako je obrazovka ci soubor,
%ale vystup je jako retezec char znaku nebo retezec typu string
%pouziva se tedy napr pro ukladani do promennych

A1 = 11;
A2 = 20;
A3 = 'a.m.';
str1 = sprintf('aktualni cas je: %d:%d %s',A1,A2,A3) %toto bude char pole
str2 = sprintf("aktualni cas je: %d:%d %s",A1,A2,A3) %toto bude string retezec
```