



# PROGRAMOVACIE JAZYKY PRE VSTAVANÉ SYSTÉMY

Cvičenie 3

# NÁPLŇ CVIČENIA

1. Komplexné čísla.
2. Jednorozmerné polia.
3. Reťazce.
4. Jednoduché funkcie a práca na úlohách.



# KOMPLEXNÉ ČÍSLA (C99)

- Jazyk C pozná okrem reálnych čísel aj imaginárne a komplexné čísla.
- C99 – komplexné čísla sú povinné, imaginárne voliteľné.
- C11 – komplexné aj imaginárne čísla sú voliteľné:
  - komplexné čísla **nie** sú podporované, ak je definované makro `__STDC_NO_COMPLEX__`
  - imaginárne čísla **sú** podporované, ak je definované makro `__STDC_IEC_559_COMPLEX__`
- GCC umožňuje pracovať len s komplexnými číslami.



# KOMPLEXNÉ ČÍSLA (C99)

- Definícia:

```
double _Complex z = 1 + 1*I;  
double complex y = 1;
```

- Hlavičkový súbor <complex.h>:

<http://en.cppreference.com/w/c/numeric/complex>

- makrá:

- complex, I

- funkcie:

- creal, cimag, cabs, carg, conj

- cexp, clog, cpow, csqrt

- trigonometrické (csin, casin,...) a hyperbolické (csinh, casinh,...)



# KOMPLEXNÉ ČÍSLA (C99) - UKÁŽKA

```
1  #include <stdlib.h>
2  #include <stdio.h>
3  #include <math.h>
4  #include <complex.h>
5
6  double complex nacistajKomplexneCislo() {
7      double real, imag;
8      scanf("%lf%lf", &real, &imag);
9      return real + imag*I;
10 }
11
12 int main(int argc, char** argv) {
13     double complex a;
14     double complex b;
15
16     printf("Zadaj realnu a imagiarnu cast komplexneho cisla\n");
17     a = nacistajKomplexneCislo();
18     b = 1 + 1*I;
19     printf("Sucet komplexnych cisel %.2f%.2fi a %.2f%.2fi = %.2f%.2fi.\n",
20           creal(a), cimag(a), creal(b), cimag(b),
21           creal(a + b), cimag(a + b));
22     printf("Podiel komplexnych cisel je %.2f%.2fi.\n",
23           creal(a / b), cimag(a / b));
24     printf("Goniometricky tvar komplexneho cisla %.2f%.2fi je "
25           "%.2f*(cos%.2f+sin%.2fi).\n",
26           creal(a), cimag(a), cabs(a), carg(a)/M_PI * 180, carg(a)/M_PI * 180);
27     return 0;
28 }
```



# JEDNOROZMERNÉ POLIA

- Pole sa alokuje ako súvislý blok pamäte.
- Jazyk C pozná 3 typy polí:
  - **polia konštantnej známej veľkosti** (veľkosť je známa pri preklade) – dátový segment, zásobník
  - **polia voliteľnej veľkosti** (C99) (veľkosť je známa až za behu) – zásobník
  - dynamicky alokované polia (veľkosť je známa až za behu) – halda
- Definícia:

```
int pole[30];  
int pole[n];  
int pole[0] = {}; //gcc  
int pole[] = {1,2,3,4,5,6};  
int pole[20] = {}; //gcc  
int pole[10] = {1,2,3,[9]=10};  
int pole[n] = {1,2,3};  
int pole[n] = {};
```



# JEDNOROZMERNÉ POLE AKO PARAMETER FUNKCIE

- Vo všeobecnosti **neexistuje** možnosť získať z poľa, ktoré bolo odovzdané funkcii ako parameter, informáciu o počte prvkov. Preto je nutné odovzdávať do funkcie nielen pole, ale aj informáciu o počte prvkov.
- Ukážka:

```
void vypisPole(int n, int pole[n]) {  
    for (int i = 0; i < n; i++) {  
        printf("%d ", pole[i]);  
    }  
    printf("\n");  
}
```



# JEDNOROZMERNÉ POLE AKO PARAMETER FUNKCIE

- Vo všeobecnosti **neexistuje** možnosť získať z poľa, ktoré bolo odovzdané funkcii ako parameter, informáciu o počte prvkov. Preto je nutné odovzdávať do funkcie nielen pole, ale aj informáciu o počte prvkov.
- Ukážka:

```
void vypisPole(int n, int pole[n]) {  
    for (int i = 0; i < n; i++) {  
        printf("%d ", pole[i]);  
    }  
    printf("\n");  
}
```





# JEDNOROZMERNÉ POLE AKO PARAMETER FUNKCIE - PRÍKLAD

- Aký bude výstup z nasledujúceho programu?

```
1  #include <stdlib.h>
2  #include <stdio.h>
3
4  void funkcia(int n, int pole[n]) {
5      printf("Pocet prvkov v poli odovzdanom ako parameter je: %ld\n",
6             sizeof(pole)/sizeof(int));
7  }
8
9  int main(int argc, char** argv) {
10     int pole[10] = {};
11     printf("Pocet prvkov v poli je: %ld\n", sizeof(pole)/sizeof(int));
12     funkcia(10, pole);
13
14     return 0;
15 }
```



# REŤAZCE

- V jazyku C sa pod reťazcom rozumie pole znakov ukončené znakom '\0'.

- Definícia:

- ako pole znakov:

```
char ret[30];
```

```
char ret[20] = {'A', 'h', 'o', 'j'};
```

```
...
```

- definícia a inicializácia pomocou literálu:

```
char ret[10] = "Ahoj";
```

```
char ret[] = "Ahoj";
```

```
char ret[5]; ret = "Ahoj"; //len cez strncpy()
```



# UŽITOČNÉ FUNKCIE A KONŠTANTY (1)

## ○ `<stdlib.h>` (<http://en.cppreference.com/w/c/numeric/random>):

- `rand()` – generuje náhodné celé čísla z intervalu `<0, RAND_MAX>`
- `srand()` – inicializácia generátora náhodných čísel
  - `srand(time(NULL))` – „náhodná“ inicializácia `//<time.h>`

## ○ `<stdio.h>` (<http://en.cppreference.com/w/c/io>):

- `gets()` (v C11 nahradená `gets_s()`), `fgets()`, `puts`, `fputs()`
- `getchar()`, `getc()`, `fgetc()`, `putchar()`, `putc()`, `fputc()`, `ungetc()`
- `sprintf()`, `snprintf()`
- `stdin`, `stdout`, `stderr`



# UŽITOČNÉ FUNKCIE A KONŠTANTY (2)

- `<string.h>` (<http://en.cppreference.com/w/c/string/byte>):
  - práca so znakmi:
    - `isalpha()`, `isdigit()`, `isupper()`, `islower()`, `isblank()` (C99)
    - `tolower()`, `toupper()`
    - ...
  - práca s reťazcami:
    - `strlen()`
    - `strcmp()`, `strncmp()`, `strcoll()`
    - `strchr()`, `strrchr()`, `strstr()`
    - `strstr`
    - `strcpy()`, `strncpy()`, `strcat()`, `strncat()`
    - ...
  - konverzia reťazca na číslo:
    - `atof`, `atoi`, `atol`, `atoll` (C99)



# ÚLOHY – JEDNODUCHÉ FUNKCIE A PROGRAMY

- Vytvorte funkcie (navrhnite parametre a návratový typ), ktoré umožnia:
  - naplniť pole odovzdané ako parameter náhodnými celými číslami z intervalu  $\langle a, b \rangle$ ;
  - nájsť minimum v poli celých čísel odovzdanom ako parameter;
  - naplniť pole char-ov náhodnými alfanumerickými znakmi tak, aby sa jednalo o reťazec;
  - naplniť pole odovzdané ako parameter náhodnými komplexnými číslami z množiny  $\langle a, b \rangle \times \langle a, b \rangle i$ ;
  - nájsť komplexné číslo s minimálnou absolútnou hodnotou v poli komplexných čísel odovzdanom ako parameter;
  - vypísať všetky prvky v poli komplexných čísel odovzdanom ako parameter buď v algebraickom alebo goniometrickom tvare (typ výpisu bude určovať dodatočný parameter funkcie);
  - utriediť pole komplexných čísel odovzdané ako parameter podľa absolútnej hodnoty vzostupne alebo zostupne (typ triedenie bude určovať dodatočný parameter funkcie);
  - vypísať súčet dvoch polynómov, ktorých koeficienty sú komplexné čísla.
- Vytvorte jednoduché programy, ktoré umožnia:
  - vyriešiť kvadratickú rovnicu (z klávesnice sa zadajú 3 reálne čísla predstavujúce koeficienty kvadratickej rovnice) v obore komplexných čísel;
  - vykonať frekvenčnú analýzu textu zadaného z klávesnice, t.j. z klávesnice sa načíta dlhý reťazec a následne sa zistia relatívne početnosti jednotlivých písmen anglickej abecedy a číslíc.

# ÚLOHY – BONUSOVÉ

- Vytvorte funkcie, ktoré umožnia:
  - vypočítať a vrátiť hodnotu komplexného polynómu v danom komplexnom čísle (využite Hornerovu schému);
  - vypísať súčin dvoch komplexných polynómov (koeficienty polynómov sa budú odovzdávať do funkcie ako polia komplexných čísel);
  - vypísať podiel dvoch komplexných polynómov (koeficienty polynómov sa budú odovzdávať do funkcie ako polia komplexných čísel).

