

První HODINA

Vývojové prostředí s kompilátorem [Dev-C++](#)

Učebnice a texty k programování <http://docs.linux.cz/>

VYPSÁNÍ ZNAKU

```
#include <stdio.h>
main()
{
    putchar('A'); //vypsani jednoho znaku
    putchar(16); //vypsani znaku v ASCII sade
    putchar(0x5a); //vypsani znaku v ASCII sade 16-kovou soustavou
    putchar(10); //novy radek
    printf("I Hate Mondays.\nYEAH\n"); //vypsani souvisleho textu
    printf("dalsi znaky - \" a %%"); //vypsani " a %

    getchar(); //ceka na zadani znaku - zabraneni zavreni okna
}
```

UŽITEČNÉ ZNAKY

zápis	funkce
\t	tabelátor
\\	\
\'	'
\n	nový řádek
\b	backspace

PŘÍKAZY PRO PRÁCI V PŘÍKAZOVÉ ŘÁDCE MS WINDOWS

```
D: // jako aktuální nastaví disk D
cd xnovak\cecko // jako aktualni nastavi slozku D:\xnovak\cecko
                 (tato složka musí na disku D existovat!)
dir // výpis souborů a složek v aktuální složce
dir /p // totéž jako dir, ale zpomalený výpis
první.exe // spuštění souboru (soubor první.exe musí v aktuální
           složce existovat)
```

ASCII TABLE

0-31	řídící znaky (10 – konec řádku)
32	mezera
33-47	pomocné znaky
48-57	čísllice
58-64	pomocné znaky
65-90	velká písmena
91-96	pravopisné znaky
97-122	malá písmena
123-127	další znaky
128-255	horní polovina ASCII tabulky (liši se podle zvolené znakové sady)



Cvičení:

1. Pouze s použitím funkce `putchar()` vypište text „Ahoj :-“.
2. Pomocí funkce `putchar()` vypište znaky s ASCII kódy 9, 36, 64.
3. Pomocí jediné funkce `printf()` vypište text a figurku panáčka:

Na 100% nezvladnu tento "tezky" úkol :(

o
/	\

DOLNÍ POLOVINA ASCII TABULKY (ZNAKY 0 - 127)

ASCII = American Standard Code for Information Interchange)

Decimal	Octal	Hex	Binary	Value
-----	-----	---	-----	-----
000	000	000	00000000	NUL (Null char.)
001	001	001	00000001	SOH (Start of Header)
002	002	002	00000010	STX (Start of Text)
003	003	003	00000011	ETX (End of Text)
004	004	004	00000100	EOT (End of Transmission)
005	005	005	00000101	ENQ (Enquiry)
006	006	006	00000110	ACK (Acknowledgment)
007	007	007	00000111	BEL (Bell)
008	010	008	00001000	BS (Backspace)
009	011	009	00001001	HT (Horizontal Tab)
010	012	00A	00001010	LF (Line Feed)
011	013	00B	00001011	VT (Vertical Tab)
012	014	00C	00001100	FF (Form Feed)
013	015	00D	00001101	CR (Carriage Return)
014	016	00E	00001110	SO (Shift Out)
015	017	00F	00001111	SI (Shift In)
016	020	010	00010000	DLE (Data Link Escape)
017	021	011	00010001	DC1 (XON) (Device Control 1)
018	022	012	00010010	DC2 (Device Control 2)
019	023	013	00010011	DC3 (XOFF) (Device Control 3)
020	024	014	00010100	DC4 (Device Control 4)
021	025	015	00010101	NAK (Negative Acknowledgement)
022	026	016	00010110	SYN (Synchronous Idle)
023	027	017	00010111	ETB (End of Trans. Block)
024	030	018	00011000	CAN (Cancel)
025	031	019	00011001	EM (End of Medium)
026	032	01A	00011010	SUB (Substitute)
027	033	01B	00011011	ESC (Escape)
028	034	01C	00011100	FS (File Separator)
029	035	01D	00011101	GS (Group Separator)
030	036	01E	00011110	RS (Request to Send) (Record Separator)
031	037	01F	00011111	US (Unit Separator)
032	040	020	00100000	SP (Space)
033	041	021	00100001	! (exclamation mark)
034	042	022	00100010	" (double quote)
035	043	023	00100011	# (number sign)
036	044	024	00100100	\$ (dollar sign)
037	045	025	00100101	% (percent)
038	046	026	00100110	& (ampersand)
039	047	027	00100111	' (single quote)
040	050	028	00101000	((left/opening parenthesis)
041	051	029	00101001) (right/closing parenthesis)
042	052	02A	00101010	* (asterisk)
043	053	02B	00101011	+ (plus)
044	054	02C	00101100	, (comma)
045	055	02D	00101101	- (minus or dash)
046	056	02E	00101110	. (dot)
047	057	02F	00101111	/ (forward slash)
048	060	030	00110000	0
049	061	031	00110001	1
050	062	032	00110010	2
051	063	033	00110011	3
052	064	034	00110100	4
053	065	035	00110101	5
054	066	036	00110110	6
055	067	037	00110111	7
056	070	038	00111000	8
057	071	039	00111001	9

058	072	03A	00111010	:	(colon)
059	073	03B	00111011	;	(semi-colon)
060	074	03C	00111100	<	(less than)
061	075	03D	00111101	=	(equal sign)
062	076	03E	00111110	>	(greater than)
063	077	03F	00111111	?	(question mark)
064	100	040	01000000	@	(AT symbol)
065	101	041	01000001	A	
066	102	042	01000010	B	
067	103	043	01000011	C	
068	104	044	01000100	D	
069	105	045	01000101	E	
070	106	046	01000110	F	
071	107	047	01000111	G	
072	110	048	01001000	H	
073	111	049	01001001	I	
074	112	04A	01001010	J	
075	113	04B	01001011	K	
076	114	04C	01001100	L	
077	115	04D	01001101	M	
078	116	04E	01001110	N	
079	117	04F	01001111	O	
080	120	050	01010000	P	
081	121	051	01010001	Q	
082	122	052	01010010	R	
083	123	053	01010011	S	
084	124	054	01010100	T	
085	125	055	01010101	U	
086	126	056	01010110	V	
087	127	057	01010111	W	
088	130	058	01011000	X	
089	131	059	01011001	Y	
090	132	05A	01011010	Z	
091	133	05B	01011011	[(left/opening bracket)
092	134	05C	01011100	\	(back slash)
093	135	05D	01011101]	(right/closing bracket)
094	136	05E	01011110	^	(caret/circumflex)
095	137	05F	01011111	_	(underscore)
096	140	060	01100000	`	
097	141	061	01100001	a	
098	142	062	01100010	b	
099	143	063	01100011	c	
100	144	064	01100100	d	
101	145	065	01100101	e	
102	146	066	01100110	f	
103	147	067	01100111	g	
104	150	068	01101000	h	
105	151	069	01101001	i	
106	152	06A	01101010	j	
107	153	06B	01101011	k	
108	154	06C	01101100	l	
109	155	06D	01101101	m	
110	156	06E	01101110	n	
111	157	06F	01101111	o	
112	160	070	01110000	p	
113	161	071	01110001	q	
114	162	072	01110010	r	
115	163	073	01110011	s	
116	164	074	01110100	t	
117	165	075	01110101	u	
118	166	076	01110110	v	
119	167	077	01110111	w	
120	170	078	01111000	x	
121	171	079	01111001	y	
122	172	07A	01111010	z	
123	173	07B	01111011	{	(left/opening brace)
124	174	07C	01111100		(vertical bar)
125	175	07D	01111101	}	(right/closing brace)
126	176	07E	01111110	~	(tilde)
127	177	07F	01111111	DEL	(delete)

Druhá HODINA

DATOVÉ TYPY

Datové typy	Velikost	Rozsah	Formát specifikace
int	4 Byte	- 2147483648 až 2147483647	%d
long int	4 Byte		%ld
short int	1 Byte	- 128 až 127	%d
char	1 Byte	- 128 až 127	%c
float	4 Byte		%f
double	8 Byte		%lf
long double	12 Byte		%Ld

ZJIŠTĚNÍ VELIKOSTI DATOVÉHO TYPY

```
#include <stdio.h>
main()
{
    printf("%d",sizeof(int));    //vypise velikost napr. datového typu int
    getchar();
    getchar();
}
```

SČÍTÁNÍ CELÝCH ČÍSEL

```
#include <stdio.h>
main()
{
    int a,b,c; //3 promenne a,b,c typu int
    scanf("%d",&a); //scanf - zadání znaku; "%d" - celociselny format
    scanf("%d",&b);
    //pres tohle se nactou nezname a,b
    c=a+b; //zadani operace pro program
    printf("%d",c); //vypise hodnotu ktera byla prirazena promenne c
    getchar();
    getchar();
}
```

SČÍTÁNÍ DESETINNÝCH ČÍSEL

```
#include <stdio.h>
main()
{
    float a,b,c; //3 promenne a,b,c typu float
    scanf("%f",&a); //scanf - zadání znaku; "%f" - format float
    scanf("%f",&b);
    //pres tohle se nactou nezname a,b
    c=a+b; //zadani operace pro program
    printf("%.2f",c); //vypise hodnotu ktera byla prirazena promenne c
    //na dvě desetinná místa

    getchar();
    getchar();
}
```

SČÍTÁNÍ ASCII KÓDŮ ZNAKŮ

```
#include <stdio.h>
main()
{
    char a,b,c;          //3 promenne a,b,c typu int
    scanf("%c",&a);      //scanf - zadání znaku; "%c" - format char
    getchar();          //nutny při pouziti char -> precte znak ENTER z bufferu
    scanf("%c",&b);
    //pres tohle se nactou nezname a,b
    c=a+b;              //zadani operace pro program
    printf("%c",c);     //vypise znak s ASCII hodnotou, která byla prirazena
                        //promenne c

    getchar();
    getchar();
}
```

OPERÁTORY v C

+	sčítání
-	odčítání
*	násobení
/	dělení (float, double, long double)
/	celočíselné dělení (int, long int)
%	zbytek při dělení (int, long int)

VÝPOČET OBJEMU VÁLCE

```
#include <stdio.h>
main()
{
    double r,v,o;
    printf("zadej polomer r: ");
    scanf("%lf",&r);
    printf("zadej vysku v: ");
    scanf("%lf",&v);
    o=3.14*r*r*v;
    printf("objem valce je: %.2lf jednotek",o);
    getchar();
    getchar();
}
```

CELOČÍSELNÉ DĚLENÍ DVOU ČÍSEL

```
#include <stdio.h>
main()
{
    int a,b,podil,zbytek;
    printf("zadej dve cisla: ");
    scanf("%d",&a);
    scanf("%d",&b);
    podil=a/b; //deleni a : b
    zbytek=a%b; //urceni zbytku pri deleni a : b
    printf("Podil je: %d\n", podil);
    printf("Zbytek je: %d", zbytek);

    getchar();
    getchar();
}
```

PŘEVOD *malých* PÍSMEN NA *VELKÁ*

```
#include <stdio.h>
main()
{
    char a;
    printf("Zadej maly pismeno: ");
    scanf("%c",&a);
    printf("A tady ho mas veeeeliky :) : %c", a-32);
    //a-32 protoze znaky velkejch pismen maji v ASCII o 32 mensi kód
    getchar();
    getchar();
}
```



Cvičení:

1. Napište program, který přečte tři čísla a vypočítá jejich aritmetický průměr a vypíše jej s přesností na tři desetinná místa.
2. Napište program pro převod velkých písmen na malá.
3. Napište program pro výpočet povrchu a objemu koule, je-li zadán její poloměr.
4. Napište program pro výpočet ceny zboží po slevě x%. Příklad: Na vstupu zadáte cenu zboží 450 Kč a slevu 15% a program vypočítá, že po slevě bude cena zboží 382,50 Kč.

BOOLEOVSKÉ OPERÁTORY

= =	Rovná se
>	menší
<	větší
<=	menší nebo rovno
>=	větší nebo rovno
!=	nerovná se
&&	konjunkce („a zároveň“)
	disjunkce („nebo“)
!	negace („není pravda, že“)

ROZHODNUTÍ O VELIKOSTI DVOU ČÍSEL

```
#include <stdio.h>
main()
{
    int a, b, c;
    printf("zadej cislo a: ");
    scanf("%d", &a);
    printf("zadej cislo b: ");
    scanf("%d", &b);
    if (a>b) {
        printf("cislo a je vetsi nez b");
    }
    else {
        printf("cislo a je mensi nebi rovno cislu b");
    }
    getchar();
    getchar();
}
```

URČENÍ TYPU VLOŽENÉHO ZNAKU

```
#include <stdio.h>
main()
{
    char c;
    printf("zadej pismeno nebo cifru: ");
    scanf("%c",&c);
    if (c>=65 && c<=90) {
        printf("znak %c je velke pismeno", c);
    }
    if (c>=97 && c<=122) {
        printf("znak %c je male pismeno", c);
    }
    if (c>=48 && c<=57) {
        printf("znak %c je cifra", c);
    }
    if (c<48 || (c>=58 && c<=64) || (c>=91 && c<=96) || c>122) {
        printf("CO TO TAM PISES?!");
    }
    getchar();
    getchar();
}
```



Cvičení:

1. Napište program, který přečte tři čísla a určí největší z nich.
2. Napište program, který přečte tři čísla a vypíše je v pořadí od nejmenšího k největšímu.

čtvrtá HODINA

URČENÍ NEJVĚTŠÍHO ČÍSLA ZE TŘÍ ČÍSEL

```
#include <stdio.h>
main()
{
    int a,b,c;
    printf("zadej prvni cislo: ");
    scanf("%d",&a);
    printf("\nzadej druhe cislo: ");
    scanf("%d",&b);
    printf("\nzadej treti cislo: ");
    scanf("%d",&c);
    if (a>b) {
        if (a>c) {
            printf ("\nNejvetsi je prvni cislo -> %d", a);
        }
        else {
            printf ("\nNejvetsi je treti cislo -> %d", c);
        }
    }
    else {
        if (b>c) {
            printf ("\nNejvetsi druhe cislo -> %d", b);
        }
        else {
            printf ("\nNejvetsi je treti cislo -> %d", c);
        }
    }
    getch();
    getch();
}
```

DALŠÍ OPERÁTORY

++	zvětšení o 1
--	zmenšení o 1

OPERACE S PROMĚNNOU

```
#include <stdio.h>
main()
{
    int a;
    printf("zadej a: ");
    scanf("%d", &a);
    a++; // zvetsi hodnotu promenne a o 1 (a--; zmensi a o 1)
    printf ("\nhodnota a je %d", a);
    a=a+5; //zvetsi hodnotu promenne a o 5
    printf ("\nhodnota a je ted %d", a);

    getch();
    getch();
}
```

VYPSÁNÍ ČÍSEL OD 1 DO 100

```
#include <stdio.h>
main()
{
    int a;
    a=1;
    while (a<=100) { //dokud bude a mensi nebo rovno 100 tak se bude provadet:
        printf ("\n%d", a);
        a++;
    } // konec cyklu while
    printf ("\nTHE END");

    getchar();
}
```

VYPSÁNÍ ZNAKŮ DOLNÍ POLOVINY ASCII

```
#include <stdio.h>
main()
{
    int a;
    a=32; // zacneme vypsanim mezery (ASCII kod 32)
    while (a<=127) { // dolni polovina ASCII tabulky konci znakem s kodem 127
        printf ("%d %c \n", a);
        a++;
    }
    printf ("\nTHE END");

    getchar();
}
```

SOUČET ČÍSEL OD 1 DO 1000

```
#include <stdio.h>
main()
{
    int a,s;
    a=1;
    s=0;
    while (a<=1000) {
        s=s+a;
        a++;
    }
    printf ("\n%d", s);

    getchar();
    getchar();
}
```



Cvičení:

1. Vypište čísla od 100 do 1 sestupně.
2. Vypište všechny násobky čísla 7 menší než 1000 (vzestupně).
3. Napište program, který vyzve uživatele, aby zadal přirozené číslo. Program potom vypíše všechna lichá čísla, která jsou menší než zadané číslo.
4. Napište program, který vyzve uživatele, aby zadal dvě celá čísla. Program potom vypíše všechna čísla, která leží mezi těmito dvěma zadanými čísly. (Pozor, program musí fungovat i v případě, že první zadané číslo je větší než druhé.)
5. Napište program, který vyzve uživatele, aby zadal dva znaky. Program potom vypíše všechny znaky z ASCII tabulky, které leží mezi těmito dvěma

- zadanými znaky.
6. Určete součet čísel všech sudých čísel ležících mezi čísly 1000 a 2000.
 7. Napište program, který vypočítá faktoriál zadaného čísla.
 8. Napište program, který čte znaky zadávané z klávesnice tak dlouho, dokud nezadáme znak „q“. U každého zadaného znaku program vypíše jeho ASCII kód.
 9. Napište program, který čte znaky zadávané z klávesnice tak dlouho, dokud nezadáme znak „q“. Na závěr program vypíše, kolik bylo mezi zadanými znaky číslic.
 10. Napište program, který přečte n čísel z klávesnice (n se zadá také z klávesnice). Pro každý vstup čísla program vypíše:
Zadej 1. číslo:
Ze zadaných čísel určete, kolik jich leží v intervalu $\langle 50, 100 \rangle$.
 11. Napište program, který přečte n čísel z klávesnice (n se zadá také z klávesnice) a určí největší z nich.

pátá HODINA

Řešení vybraných úloh ze cvičení z minulé hodiny:

Cvičení 3:

Můžeme postupovat takto:

```
#include <stdio.h>
main()
{
    int a;
    printf("zadej cele cislo: ");
    scanf("%d", &a);
    while (a >= 0) {
        printf ("%d ", a);
        a=a-2;
    }
    printf("\nTHE END");

    getchar();
    getchar();
}
```

...nebo třeba takto:

```
#include <stdio.h>
main()
{
    int a, n;
    printf("zadej cele cislo: ");
    scanf("%d", &n);
    a=1;
    while (a < n) {
        printf ("%d ", k);
        a=a+2;
    }
    printf ("\nTHE END");

    getchar();
    getchar();
}
```

Cvičení 4:

```
#include <stdio.h>
main()
{
    int a, b;
    printf("zadej cele cislo a: ");
    scanf("%d", &a);
    printf("zadej cele cislo b: ");
    scanf("%d", &b);
    if(a < b) {
        while (a < b-1) {
            a=a+1;
            printf ("%d ", a);
        }
    }
}
```

```

else {
    while (b < a-1) {
        b=b+1;
        printf ("%d ", b);
    }
}
printf ("\nTHE END");

getchar();
getchar();
}

```

Cvičení 7:

```

#include <stdio.h>
main()
{
    double n, f;
    f=1;
    printf("zadej prirodzene cislo pro vypocet jeho faktorialu: ");
    scanf("%lf", &n);
    while (n > 0) {
        f=f*n;
        n--;
    }
    printf ("\nfaktorial je: %.0lf", f);

    getchar();
    getchar();
}

```

Cvičení 8:

```

#include <stdio.h>
main()
{
    int a;
    printf("napis znak pro vypis jeho ASCII hodnoty, nebo 'q' pro konec:\n");
    a = getchar();
    getchar();
    while (a != 'q') {
        printf ("\nASCII hodno znaku '%c' je: %d\n", a, a);
        a = getchar();
        getchar();
    }
    printf ("\n\nTHE END");

    getchar();
}

```

FOR CYKLUS - VYPSÁNÍ ČÍSEL OD 1 DO 100

```
#include <stdio.h>
main()
{
    int a;
    for (a=1; a<=100; a++) {
        printf("%d ", a);
    }
    getchar();
}
```

...totéž pomocí while cyklu:

```
#include <stdio.h>
main()
{
    int a;
    a=1;
    while (a <= 100) {
        printf ("%d ", a);
        a++;
    }

    getchar();
}
```

FAKTORIÁL POMOCÍ FOR CYKLU

```
#include <stdio.h>
main()
{
    double i, n, f;
    f=1;
    printf("zadej prirodzene cislo pro vypocet jeho faktorialu: ");
    scanf("%lf", &n);
    for (i=1; i<=n; i++) {
        f=f*i;
    }
    printf ("\nfaktorial cisla %.0lf je %.0lf", n, f);

    getchar();
    getchar();
}
```



Cvičení:

1. Pomocí FOR cyklu vypište čísla od 100 do 1 sestupně.
2. Pomocí FOR cyklu vypište všechny násobky čísla 7 menší než 1000 (vzestupně).
3. Napište program, který vyzve uživatele, aby zadal přirozené číslo. Program potom vypíše všechna lichá čísla, která jsou menší než zadané číslo (řešte pomocí FOR cyklu).
4. Pomocí FOR cyklu určete součet čísel všech sudých čísel ležících mezi čísly 1000 a 2000.

šestá HODINA

<http://devpaks.org/>
<http://usuarios.lycos.es/charlytospaige/dev.htm>
<http://codecutter.org/tools/winbgim/>

VYBRANÉ FUNKCE Z KNIHOVNY WINBGIM

```
#include <winbgim.h>
main()
{
    int i;
    initwindow(500,500); // otevře grafické okno 500 x 500 bodů
    setcolor(YELLOW);   // nastavení barvy pera
    line(30,50,200,100); // úsečka z bodu [30, 50] do bodu [200, 100]
    circle(200,200,100); // kružnice se středem v bodě [200, 200] a poloměr. 100
    rectangle(30,50,200,100); // obdélník s levým horním rohem v bodě [30, 50]
                                // a pravým dolním rohem v bodě [200, 100]
    while(!kbhit());      // čeká na stisk klávesy
    closegraph();        // zavře grafické okno
}
```



Cvičení:

1. Nakreslete obrázek domečku s plotem a sluncem nad ním.
2. Nakreslete několik soustředných kružnic.
3. Vytvořte nějakou jednoduchou animaci (postupně se vykreslující úsečky, soustředné kružnice, obdélníky, apod.). Použijte funkci `delay()`.

sedmá HODINA

ANIMOVANÁ ÚSEČKA

```
#include <winbgim.h>
main()
{
    int i;
    initwindow(500,500); // otevře grafické okno 500 x 500 bodů
    for (i=1; i<=100; i++) {
        putpixel(50+i,50,YELLOW); // nakreslí bod na zadané souřadnice
        delay(50);
    }
    while(!kbhit()); // čeká na stisk klávesy
    closegraph(); // zavře grafické okno
}
```

GRAF FUNKCE SINUS

```
#include<winbgim.h>
#include<stdlib.h>
#include<math.h>
main()
{
    const float pi=3.1415;
    initwindow(640,480);
    for (int x=0; x<640; x++)
        putpixel(x, 100+50*sin(2*pi*(float)x/640), WHITE);
    system("pause");
    closegraph();
}
```

MÍČEK ODRÁŽEJÍCÍ SE OD STĚN

```
#include <winbgim.h>
main()
{
    int i=2;
    initwindow(500,500);
    setcolor(GREEN);
    setlinestyle(0, 1, 3); // nastaví typ čáry - plná o tloušťce 3 body
    line(50,0, 50,300); // nakreslí levou stěnu o tloušťce 3 body zelenou barvou
    line(200,0, 200,300); // nakreslí pravou stěnu o tloušťce 3 body zelenou barvou
    while(!kbhit()) {
        // kreslení bodu letícího doprava
        while (getpixel(50+i,50) != GREEN) { // dokud nenarazí na stěnu
            putpixel(50+i,50, WHITE); // nakreslí bod na novou pozici
            putpixel(50+i-1,50, BLACK); // smaže bod z předchozí pozice
            i++;
            delay(10);
        }
        i--;
        // a podobně kreslení bodu letícího doleva
        while (getpixel(50+i,50) != GREEN) {
            putpixel(50+i,50, WHITE);
            putpixel(50+i+1,50, BLACK);
            i--;
            delay(10);
        }
        i++;
    }
}
```

```
}  
while(!kbhit());  
closegraph();  
}
```



Cvičení:

1. Pomocí for cyklu a funkce `putpixel()` nakreslete svislou a šikmou úsečku.
2. Pomocí for cyklu a funkce `putpixel()` nakreslete pravoúhlý trojúhelník.
3. Pomocí dvou do sebe vnořených for cyklů a funkce `putpixel()` nakreslete vyplněný obdélník. Vytvořte jednoduchou animaci - vykreslování obdélníku zpomalte pomocí funkce `delay()`.
4. Podobně jako v předchozí úloze nakreslete vyplněný pravoúhlý trojúhelník.
5. Pomocí funkce `delay()` vytvořte jednoduchou animaci – letící bod.

ŽÍŽALA OVLÁDANÁ KLÁVESAMI

```
#include <winbgim.h>
main()
{
    char c;
    int i=100, j=100;
    initwindow(1000,600);
    do {
        putpixel(i, j, WHITE);
        c = (char) getch();
        switch (c) {
            case 'j': i--; break;
            case 'k': i++; break;
            case 'i': j--; break;
            case 'm': j++; break;
        }
    } while (c != 'q');
    closegraph();
}
```

ŽÍŽALA OVLÁDANÁ KLÁVESAMI - ZRYCHLENÁ VERZE

```
#include <winbgim.h>
main()
{
    char c;
    int i=100, j=100;
    initwindow(1000,600);
    do {
        putpixel(i, j, WHITE);
        if (kbhit()) {
            c = (char) getch();
        }
        delay(10);
        switch (c) {
            case 'j': i--; break;
            case 'k': i++; break;
            case 'i': j--; break;
            case 'm': j++; break;
        }
    } while (c != 'q');
    closegraph();
}
```

ŽÍŽALA OVLÁDANÁ KLÁVESAMI - OBECNĚJŠÍ VERZE

```
#include <winbgim.h>
main()
{
    char c;
    int ax=450, ay=300;
    int adx=0, ady=0;
    initwindow(1000,600);
    do {
        putpixel(ax, ay, WHITE);
        if (kbhit()) {
            c = (char) getch();
            switch (c) {
                case 's': adx=-1; ady=0; break;
                case 'd': adx=1; ady=0; break;
                case 'e': adx=0; ady=-1; break;
                case 'x': adx=0; ady=1; break;
            }
        }
        ax = ax + adx;
        ay = ay + ady;
        delay(10);
    } while (c != 'g');
    closegraph();
}
```

DVĚ ŽÍŽALY OVLÁDANÉ KLÁVESAMI

```
#include <winbgim.h>
main()
{
    char c;
    int ax=450, ay=300;
    int adx=0, ady=0;
    int bx=550, by=300;
    int bdx=0, bdy=0;
    initwindow(1000,600);
    do {
        putpixel(ax, ay, WHITE);
        putpixel(bx, by, YELLOW);
        if (kbhit()) {
            c = (char) getch();
            switch (c) {
                case 's': adx=-1; ady=0; break;
                case 'd': adx=1; ady=0; break;
                case 'e': adx=0; ady=-1; break;
                case 'x': adx=0; ady=1; break;
                case 'j': bdx=-1; bdy=0; break;
                case 'k': bdx=1; bdy=0; break;
                case 'i': bdx=0; bdy=-1; break;
                case 'm': bdx=0; bdy=1; break;
            }
        }
        ax = ax + adx; ay = ay + ady;
        bx = bx + bdx; by = by + bdy;
        delay(10);
    } while (c != 'g');
    closegraph();
}
```

HRA TRON

```
#include <winbgim.h>
main()
{
    char c;
    int ax=450, ay=300, adx=-1, ady=0;
    int bx=550, by=300, bdx=1, bdy=0;
    initwindow(1000,600);
    do {
        putpixel(ax, ay, WHITE); putpixel(bx, by, YELLOW);
        if (kbhit()) {
            c = (char) getch();
            switch (c) {
                case 's': adx=-1; ady=0; break;
                case 'd': adx=1; ady=0; break;
                case 'e': adx=0; ady=-1; break;
                case 'x': adx=0; ady=1; break;
                case 'j': bdx=-1; bdy=0; break;
                case 'k': bdx=1; bdy=0; break;
                case 'i': bdx=0; bdy=-1; break;
                case 'm': bdx=0; bdy=1; break;
            }
            ax = ax + adx; ay = ay + ady;
            bx = bx + bdx; by = by + bdy;
            if (getpixel(ax, ay) != BLACK) break;
            if (getpixel(bx, by) != BLACK) break;
            delay(10);
        } while (c != 'g');
        if (getpixel(ax, ay) != BLACK)
            outtextxy(100, 100, "Vyhral hrac B!");
        if (getpixel(bx, by) != BLACK)
            outtextxy(100, 100, "Vyhral hrac A!");
        delay(5000);
        closegraph();
    }
}
```



Cvičení:

1. Vylepšete hru Tron tak, aby se hráči nemohli dostat mimo viditelnou oblast okna. Nápopvěda: Orámujte hrací plochu barevným rámečkem.
2. Vylepšete hru Tron tak, aby hráči mohli hrát opakované partie bez toho, aby museli hru znovu startovat.
3. Doplněte do hry Tron počítadlo stavu partií.
4. Doplněte do hrací plochy hry Tron další překážky (obdélníčky, kruhy). Některé z nich mohou být smrtelné jen pro některého hráče, jiné mohou zrychlovat pohyb hráče, apod.

PŘÍKLADY DEFINIC FUNKCÍ

```
#include <stdio.h>
#include <math.h>

// definice fce pro sčítání dvou čísel
double secti(double a, double b)
{
    return a+b;
}

// definice fce pro výpočet aritmetického průměru
double aprumer(double a, double b)
{
    return (a+b)/2;
}

// definice fce, která určí větší ze dvou čísel
double max(double a, double b)
{
    if (a>b) {
        return a;
    }
    else {
        return b;
    }
}

//definice fce, která počítá délku přepony pravoúhlého trojúhelníku
double pythagor(double a,double b)
{
    return sqrt(a*a+b*b);
}

//definice fce, která sečte všechna celá čísla od a do b
int suma(int a, int b)
{
    int c, s=0;
    if (a>b) {
        c=a;
        a=b;
        b=c;
    }
    while (a<=b) {
        s=s+a;
        a++;
    }
    return s;
}

main()
{
    double a,b,c;

    //nacteni hodnot
    printf("zadej cislo a: ");
    scanf("%lf", &a);
    printf("zadej cislo b: ");
    scanf("%lf", &b);
```

```

//vystup fce secti()
c=secti(a,b);
printf ("\nsoucet cisel je %0.0lf", c);

//vystup fce max()
c=max(a,b);
printf("\nvetsi cislo je %0.0lf", c);

//vystup fce aprumer()
c=aprumer(a,b);
printf("\naritmeticky prumer cisel je: %0.2lf", c);

//vystup fce pythagor()
c=pythagor(a,b);
printf("\nprepona ma velikost %0.2lf", c);

//vystup fce suma()
c=suma(a,b);
printf("\nsuma dvou cisel je %d", (int) c);

getchar();
getchar();
}

```

FUNKCE, KTERÁ PŘEVÁDÍ MALÁ PÍSMENA NA VELKÁ A NAOPAK

```

#include <stdio.h>
#include <math.h>

//definice fce konverze()
char konverze(char a)
{
    if ((a>=65) && (a<=90)) {
        a=a+32;
        return a;
    }
    if ((a>=97) && (a<=122)) {
        a=a-32;
        return a;
    }
    return a;
}

main()
{
    char a,c;
    printf("zadej velke nebo male pismeno (bez diakritiky): ");
    scanf("%c", &a);
    if ((a<65) || ((a>90) && (a<97)) || (a>122)) {
        printf("\nvis co je to pismeno?!?!... \"%c\" to urcite neni!", a);
    }
    else {
        c=konverze(a);
        printf("\ntady je vysledek konverze: %c", c);
    }
    getchar();
    getchar();
}

```

PŘÍKLADY DEFINIC GRAFICKÝCH FUNKCÍ

```

#include <winbgim.h>
#include <math.h>
// vodorovná čára z bodu [x,y] o délce delka a barvě barva
void hline (int x, int y, int delka, int barva) {
    int i=0;
    while (i<=delka) {
        putpixel (x+i, y, barva);
        i++;
    }
}
// svislá čára z bodu [x,y] o délce delka a barvě barva
void vline (int x, int y, int delka, int barva) {
    int i=0;
    while (i<=delka) {
        putpixel (x, y+i, barva);
        i++;
    }
}
// čtverec s levým horním rohem v bodě [x,y] a stranou délky delka a barvou barva
void ctverec (int x, int y, int delka, int barva) {
    int i=0;
    while (i<=delka) {
        putpixel (x+i, y, barva);
        putpixel (x, y+i, barva);
        putpixel (x+delka, y+i, barva);
        putpixel (x+i, y+delka, barva);
        i++;
    }
}
// obdélník s levým horním rohem v b. [ax,ay] a pravým dolním rohem v b. [bx, by]
void obdelnik (int ax, int ay, int bx, int by, int barva) {
    hline (ax, ay, bx-ax, barva);
    hline (ax, by, bx-ax, barva);
    vline (ax, ay, by-ay, barva);
    vline (bx, by-(by-ay), by-ay, barva);
}
// domeček s levým horním rohem v bodě [x,y] a šířkou základny strana
void domecek (int x, int y, int strana) {
    int a=0;
    while ((a+x)<=(x+(strana/2))) {
        putpixel (x+a, y-a, RED);
        a++;
    }
    a=0;
    while ((a+x)<=(x+(strana/2))){
        putpixel (x+strana-a, y-a, RED);
        a++;
    }
    ctverec (x, y, strana, WHITE);
}
// kružnice se středem v bodě [ax,ay], poloměrem r
void kruznice (int sx, int sy, int r, int barva) {
    double fi;
    for(fi=0; fi<=6.28; fi=fi+0.01) {
        putpixel ((int)sx+r*cos(fi),(int)sy+r*sin(fi), barva);
    }
}

```

```
main()
{
    initwindow(1000,500);
    //    hline(50, 80, 30);
    //    vline(50, 80, 30);
    //    ctverec(50,80,300);
    //    obdelnik(50,50,200,100);
    //    domecek(200,200,200);
    kruznice (100,200,50);
    while(!kbhit());
    closegraph();
}
```

UKÁZKA PRÁCE S POLEM CELÝCH ČÍSEL

```
#include <stdio.h>
#include <stdlib.h>

main()
{
    int a[10]; // deklarace pole deseti hodnot celočíselného typu
    int n, max=0, min=100, s=0;
    srand(time(NULL)); // nastaví výchozí vhodnotu pro rand() na aktuální čas

    for(n=0; n<=9; n++) {
        a[n]=rand()%100+1; // generuje náhodná čísla od 1 do 100
    }
    for(n=0; n<=9; n++) {
        printf("%d ", a[n]); // výpis hodnot pole
    }
    for(n=0; n<=9; n++) {
        if (a[n]>max) { // hledání největší hodnoty v poli
            max=a[n];
        }
    }
    for(n=0; n<=9; n++) {
        if (a[n]<min) { // hledání nejmenší hodnoty v poli
            min=a[n];
        }
    }
    for(n=0; n<=9; n++) {
        s=s+a[n]; // sčítání hodnot v poli
    }
    printf("\nNejvetsi cislo v poli je %d", max);
    printf("\nNejmensi cislo v poli je %d", min);
    printf("\nSoucet cisel v poli je %d", s);
    getchar();
}
```



Cvičení:

1. Deklarujte pole pro uložení 20 hodnot typu `int` a naplňte jej přirozenými násobky čísla 5, tj. čísla 5, 10, 15, ..., 1000. Poté hodnoty v poli vypište na řádek oddělené mezerami.
2. Deklarujte pole pro uložení 50 hodnot typu `double` a naplňte jej desetinnými čísly 0,01; 0,02; 0,03; ...; 0,50. Poté hodnoty v poli vypište každou na nový řádek.
3. Deklarujte pole pro uložení 100 hodů kostkou a poté jej naplňte náhodnými čísly z rozsahu 1 až 6. Získané hodnoty vypište a určete, kolikrát se mezi nimi vyskytuje číslo 6.
4. Deklarujte pole pro uložení 30 zatočení rulety a poté jej naplňte náhodnými čísly z rozsahu 0 až 36. Získané hodnoty vypište a určete, kolik z nich je sudých.
5. Deklarujte pole pro uložení 1000 hodů mincí a poté jej naplňte náhodnými čísly 0 nebo 1 (0=rub mince, 1=líc mince). Získané hodnoty vypište a určete v procentech, kolik padlo rubů a koli líců.