**http://jazyk-c.wz.cz/cykly.html**

**Spracované zdrojové kódy v jazyku C**

1. Jednorozmerné pole – načítanie a výpis znakov

Jednorozmerné pole, vysvetlite postupnosť činností počítača pri práci s poľom,

pri deklarácii jednorozmerného poľa, ktoré funkcie sú použité pre zápis a výpis znakov.

*Príklad*

Vytvorte program, pomocou ktorého zapíšete do znakového poľa text maximálne o dĺžke tridsiatich znakov. Pomocou cyklu vypíšte tento text od posledného po prvý znak ( pospiatky): napr. abcdefgh kobyla ma maly bok

hgfedcba kob ylam am alybok

**#include <stdio.h>**

**#define N 30**

**int main()**

**{**

**char pole[N];**

**int i=0,j,znak;**

**printf("Napiste slovo(max. %d pismen) -> ", N);**

**do**

**{**

**znak=getch();**

**if (znak!=13) //pokym nestlacim enter**

**{**

**putch(znak); //vypis znak**

**pole[i]=znak; //zapis do pola**

**i++;**

**}**

**} while (znak!=13); --i; //znaky su v poli[0] az poli[i-1]**

**printf("\n Slovo pospiatky: \n");**

**for (j=i; j>=0; j--)**

**putch(pole[j]);**

**getch();**

**return 0;**

**}**

**2. Podmienený príkaz – vnorené podmienky**

Vysvetlite kedy je potrebné použiť vnorené podmienky v algoritme. Prečo nie je možné vždy použiť príkaz viacnásobného vetvenia ?

*Príklad*

Napíšte program na určovanie BMI indexu človeka. Program načíta hmotnosť človeka v kg a výšku v m. Potom vypočíta BMI index podľa vzorca



kde hmotnosť sa udáva v kilogramoch a výška v metroch. Program nech vypíše, či máte:

* podváhu (BMI < 18,5),
* normálnu hmotnosť (18,5<=BMI<25,0),
* nadváhu (25,0<=BMI<30,0),
* obezitu (BMI>30,0).

Pri riešení úlohy využite vnorené podmienky:

**#include<stdio.h>**

**#include<stdlib.h>**

**#include<time.h>**

**int main()**

**{**

**int hmotnost;**

**float BMI, vyska,i ;**

**printf("zadaj svoju vysku v metroch:\n");**

**scanf("%f",&vyska);**

**printf("zadaj svoju hmotnost \n");**

**scanf("%d",&hmotnost);**

**i =vyska\*vyska;**

**BMI = (float) hmotnost /(float)i;**

**printf("BMI je%f",BMI);**

**if(BMI<18.5)**

**printf("podvaha\n");**

**if(BMI >=18.5&&BMI<25.0)**

**printf("normalna hmotnost\n");**

**if(BMI >=25.0&&BMI<30.0)**

**printf("nadvaha\n");**

**if(BMI>30.0)**

**printf("obezita\n");**

**system("pause");**

**return 0;**

**}**

3. Základy – logická premenná DOKONčIŤ !

Vysvetlite pojmy: logická premenná, hodnoty logickej premennej

Napíšte program, pomocou ktorého načítate na vstupe celé číslo C. Vytvorte nasledujúce logické premenné a ich hodnoty vypíšte na obrazovku

* JEKLADNE, v ktorej bude uložený výsledok overenia, či dané číslo je kladné,
* JEDVOJCIFERNE, v ktorej bude uložený výsledok overenia, či je číslo dvojciferné,
* JEZVYSOK3, v ktorej bude uložený výsledok overenia, či po delení 10 dáva číslo zvyšok 3.

Výpis aspoň v prvom prípade doplňte o slovné vyjadrenie danej vlastnosti čísla (je kladné / nie je kladné).

*Možný výpis programu:*

výpis: Zadaj celé číslo:

vstup: 15

výpis: JEKLADNE=1 JEDVOJCIFERNE=1 JEZVYSOK3=0

Cislo je kladne.

#include <stdio.h>

#include <stdlib.h>

int main(void) {

int JEZVYSOK3, JEDVOJCIFERNE, JEKLADNE=1, a,i=2;

printf("Zadajte cele cislo\n");

scanf("%d\n", &a);

JEZVYSOK3= (a%10==3);

printf("JE ZVYSOK 3....%d\n", JEZVYSOK3);

JEDVOJCIFERNE= (a>=10) && (a<=99);

printf("Je dvojciferne %d\n", JEDVOJCIFERNE);

while (i<a) {

if (a%i == 0) JEKLADNE=1;

i++;

};

printf("Je kladne %d\n", JEKLADNE);

system("pause");

return 0;

}

/\* DOKONCIT VYPIS CI JE KLADNE\*/

4.Základy – príkazy vstupu a priradenia

Vysvetlite pojmy: príkaz vstupu, príkaz priradenia. Existuje nejaká ďalšia možnosť načítania údajov do premenných?

*Príklad*

Napíšte program, ktorý nakreslí na obrazovku písmeno „I“ (ako Informatika) podľa obrázku, pričom základný znak (napr. „x“) zadá užívateľ do premennej z pomocou klávesnice cez príkaz vstupu:

xxxxx  
 x  
 x

x  
 x  
xxxxx

Doplňte program tak, že počítač nakreslí ešte jedno písmeno „I“. Tento krát sa však vytvorí z náhodného znaku veľkého písmena (ASCII 65-90), ktoré uložíte do premennej Z pomocou príkazu priradenia.

#include <stdio.h>

#include <stdlib.h>

#include <time.h>

int main(void)

{

char x, Z;

srand(time(0));

Z = rand()%26 + 65;

printf("Zadaj pismeno a vykresli znak I:\n");

scanf("%c,%c", &x, &Z);

printf("\n\n%c%c%c%c%c\n", x, x, x, x, x);

printf(" %c\n", x);

printf(" %c\n", x);

printf(" %c\n", x);

printf(" %c\n", x);

printf("%c%c%c%c%c\n", x, x, x, x, x);

printf("\n");

printf("%c%c%c%c%c\n", Z, Z, Z, Z, Z);

printf(" %c\n", Z);

printf(" %c\n", Z);

printf(" %c\n", Z);

printf(" %c\n", Z);

printf("%c%c%c%c%c\n", Z, Z, Z, Z, Z);

system("pause");

return 0;

}

5: Aritmetika – operácie s reálnymi premennými

Vysvetlite pojmy: reálna premenná, aritmetické operácie s reálnymi premennými

*Príklad*

Napíšte program pre jednoduché aritmetické operácie: Na vstupe budú zadané dve čísla, ktoré program uloží do 2 reálnych premenných (dátový typ pre desatinné čísla). Následne program do 4 premenných (sucet, rozdiel, sucin, podiel) uloží výsledky pre sčítanie, odčítanie, násobenie a delenie. Nakoniec sa počítač užívateľa opýta na operáciu, ktorú chce použiť. Jeho voľba sa uloží do znakovej premennej a na obrazovku sa vypíše výsledok operácie. Doplňte program tak, aby fungoval v cykle: Po výpise výsledku sa počítač vždy opýta užívateľa, či chce nový výpočet alebo či chce skončiť.

#include <stdio.h>

int main(void)

{

float a, b, vysledok;

int volba;

char pokracovat = 1;

printf("Vitajte v kalkulacke \n");

while(pokracovat == 1)

{

printf("Zadajte prve cislo: \n");

scanf("%f", &a);

printf("Zadajte druhe cislo: \n");

scanf("%f", &b);

printf("Zvolte si aritmeticku operaciu: \n");

printf("1 - scitanie \n");

printf("2 - odcitanie \n");

printf("3 - nasobenie \n");

printf("4 - delenie \n");

scanf("%d", &volba);

switch(volba)

{

case 1:vysledok = a + b; break;

case 2:vysledok = a - b; break;

case 3:vysledok = a \* b; break;

case 4:vysledok = a / b; break;

}

if ((volba > 0) && (volba < 5))

printf("Vysledok: %f\n", vysledok);

else

printf("Neplatná volba \n");

printf("Chcete zadat dalsi priklad? [1/0]");

scanf("%d", &pokracovat);

}

printf("Dakujem za pouzitie kalkulacky");

system ("Pause");

return 0;

}

6. Základy – číselné premenné

Vysvetlite pojmy: premenná, konštanta, číselná premenná

*Príklad*

Napíšte program, ktorý bude fungovať ako registračná pokladňa. Užívateľ bude opakovane zadávať cenu za kus a počet kusov tovaru. Zadávanie skončí, keď vloží „0“.

Ku vypočítanej sume počítač ešte pripočíta DPH 20% a poštovné (cena poštovného je zapísaná ako konštanta na úvod programu).

Nakoniec počítač prehľadne pod seba vypíše „registračný bloček“, t.j. celkovú cenu tovaru, hodnotu DPH a poštovného a celkovú sumu dobierky

#include <stdio.h>

#include <stdlib.h>

int main(void) {

int pocet;

float cena, sum, DPH;

float postovne=1.50;

do {

printf("Zadaj cenu za ks:\n");

scanf("%f",&cena);

if (cena!=0) {

printf("Zadaj pocet ks:\n");

scanf("%d",&pocet);

sum=sum+cena\*pocet;

}

}

while (cena!=0);

DPH=sum\*0.2;

printf("Suma bez DPH:\t%.2f\n",sum);

printf("DPH 20%:\t\t%.2f\n",DPH);

printf("Postovne:\t%.2f\n",postovne);

printf("--------------------------\n");

printf("Celkova suma dobierky:\t%.2f\n",sum+DPH+postovne);

system("pause");

return 0;

}

7. Cykly – vnorené cykly, viacnásobné vetvenie

Vysvetlite pojmy: cyklus, vnorený cyklus, viacnásobné vetvenie

*Príklad*

Nakreslite v textovom režime pomocou hviezdičiek vo vnorených cykloch oba geometrické útvary, ktoré sú v obrázkovej prílohe:

* štvorec 5x5,
* pravouhlý trojuholník s výškou 5.

Užívateľ nech na začiatku zvolí, ktorý z útvarov chce vykresliť. Na výber medzi jednotlivými útvarmi využite príkaz viacnásobného vetvenia. Vedeli by ste zmenou niektorého parametra vykresliť aj nejaký ďalší útvar s výškou 5 riadkov?

**\*\*\*\*\* \***

**\*\*\*\*\* \*\***

**\*\*\*\*\* \*\*\***

**\*\*\*\*\* \*\*\*\***

**\*\*\*\*\* \*\*\*\*\***

**#include <stdio.h>**

**int main(void)**

**{**

**int i, j, vel;**

**printf("Zadaj cele kladne cislo: ");**

**scanf("%d", &vel);**

**for (i = 1; i <= vel; i++) {**

**for (j = 1; j <= vel; j++)**

**putchar('\*');**

**putchar('\n');**

**}**

**putchar('\n');**

**system ("pause");**

**return 0;**

**}**

**#include <stdio.h>**

**int main()**

**{**

**int i,j,vyska;**

**printf("Zadaj vysku v riadkoch:");**

**scanf("%d", &vyska);**

**for ( i = 1; i <= vyska; i++ ) {**

**for ( j = 1; j <= i; j++ ) {**

**putchar('\*');**

**}**

**putchar('\n');**

**}**

**system ("pause");**

**return 0;**

**}**

8. Aritmetika – operácie s celými číslami

Vysvetlite pojmy: celočíselný dátový typ, aritmetické operácie s celými číslami.

*Príklad*

Napíšte program Ciferny\_sucet, ktorý dokáže vypočítať súčet všetkých cifier celého čísla, ktoré je načítavané na vstupe z klávesnice. Využite vhodné operácie s celými číslami.

**#include <stdio.h>**

**int main()**

**{**

**int n, t, sum = 0, pripocitaj;**

**printf("Zadaj cele cislo zlozene z maximalne 9 cislic:\n");**

**scanf("%d", &n);**

**t = n;**

**while (t != 0)**

**{**

**pripocitaj = t % 10;**

**sum = sum + pripocitaj;**

**t = t / 10;**

**}**

**printf("Sucet vsetkych cislic v cisle %d je = %d:\n", n, sum);**

**system ("pause");**

**return 0;**

**}**

9.Cyklus s podmienkou – podmienky a ich použitie

Vysvetlite: Načo sú podmienky v príkaze cyklu, kedy ich vytvárame a ako ich programovací jazyk vyhodnocuje?

Napíšte program, ktorý pomôže vyhodnotiť výsledky testu na vysokej škole a na konci napíše dosiahnutý priemer. Program bude opakovane v cykle určovať známku žiaka z testu. Užívateľ zadá počet bodov a počítač vypíše slovné hodnotenie na základe bodov takto:

0-9 bodov: nevyhovujúco (3); 10-15 bodov: veľmi dobre (2); 16-18 bodov: výborne (1).

Uvedené známky ukladá priebežne do premennej SUM. Program sa ukončí, ak počet bodov bude zadaný ako záporné číslo. Potom na obrazovke vypíše priemer známok žiakov.

#include<stdio.h>

int main(void)

{

int body, SUM=0, i=0;

float priemer;

do {

printf("Zadaj pocet bodov:\n");

scanf("%d", &body);

if (body <= 9 && body >= 0)

{ printf("0-9 bodov: nevyhovujuco (3)\n");

i+=1;

SUM += 3;

} else if (body >= 10 && body <= 15)

{ printf("10-15 bodov: velmi dobre (2)\n");

i+=1;

SUM += 2;

}

else if (body <= 18 && body >= 16)

{

i+=1;

printf("16-18 bodov: Vyborne (1)\n");

SUM += 1;

}

else if (body > 18)

{

i +=0;

printf("Zadal si body mimo rozsah !\n");

SUM += 0;

}

printf("Pocet SUM je %d\n", SUM);

}

while(body >= 0);

priemer =(float) SUM / (float) i;

if(body < 0)

printf("Priemer znamok je %.2f\n", priemer);

system("pause"); }

**10. Cyklus s podmienkou – počet opakovaní**

Vysvetlite ako funguje (dokedy platí) podmienka v cykle a ako podmienka vplýva na počet opakovaní cyklu.

*Príklad*

Napíšte program pre nájdenie najväčšieho spoločného deliteľa (alebo najmenší spoločný násobok) dvoch celých kladných čísel. Na vstupe užívateľ zadá 2 celé kladné čísla (ošetrite vstup tak, aby pri zadaní záporného čísla musel zadať znova). Počítač uloží výpočet najväčšieho spoločného deliteľa do premennej NSD (alebo najmenšieho spoločného násobku do premennej NSN) a jej obsah vypíše na obrazovku.

**#include <stdio.h>**

**int main ()**

**{**

**int a, b, c, d;**

**printf ( "Zadaj dve cisla\n" );**

**scanf ( "%d %d", &a, &b );**

**if (( a <= 0 ) || ( b <= 0 ))**

**{**

**printf ( "Zadane cisla musia byt kladne\n" );**

**return ( 1 );**

**}**

**c = a;**

**d = b;**

**while ( a != b )**

**if ( a > b )**

**a -= b;**

**else**

**b -= a;**

**printf ( "najvacsi spolocny delitel: %d, najmensi spolocny nasobok: %d\n", a, c \* d, d / a );**

**system ("Pause");**

**return 0;**

**}**

11.Podmienený príkaz – neúplný

Vysvetlite syntax neúplného podmieneného príkazu. Uvedením vhodných viet demonštrujte použitie tohto cyklu v bežnej hovorovej reči.

*Príklad*

Vytvorte program, ktorý ohodnotí užívateľa podľa známok. Užívateľ zadáva opakovane známky v cykle, zadávanie končí napísaním čísla 0. Pri každej zadávanej známke počítač vypíše aj poradie zadávanej známky. Počítač vypočíta priemer a podľa neho ohodnotí užívateľa – výborný (priemer menej ako 1,5), chválitebný (väčší alebo rovný 1,5 a menší ako 2,5), dobrý (väčší alebo rovný 2.5 a menší ako 3.5), dostatočný (väčší alebo rovný 3,5 a menší ako 4,5), nedostatočný (väčší alebo rovný 4,5).

Pri riešení úlohy aspoň raz použite neúplný podmienený príkaz.

**#include <stdio.h>**

**#include <stdlib.h>**

**int main(void)**

**{**

**int znamka, i=0, pocet=0;**

**float priemer;**

**do{**

**printf("Zadaj znamku:\n");**

**scanf("%d", &znamka);**

**if(znamka == 0){**

**i += 0;**

**}**

**if(znamka > 0&&znamka<6)**

**i += 1;**

**printf("Znamka: %d, Poradie: %d\n", znamka, i);**

**pocet += znamka;**

**}**

**while(znamka != 0);**

**priemer = (float) pocet/(float) i;**

**if (priemer < 1.5)**

**printf("Vyborny\n");**

**if (priemer >= 1.5 && priemer < 2.5)**

**printf("Chvalitebny\n");**

**if (priemer >= 2.5 && priemer < 3.5)**

**printf("Dobry");**

**if (priemer >= 3.5 && priemer < 4.5)**

**printf("Dostatocny");**

**if (priemer >= 4.5)**

**printf("Nedostatocny");**

**system("pause");**

**return 0;**

**}**

12: Podmienený príkaz – úplný

Vysvetlite syntax úplného podmieneného príkazu. Uvedením vhodných viet demonštrujte použitie tohto príkazu v bežnej hovorovej reči.

Napíšte program, ktorý bude skúšať užívateľa formou testu. Počítač vypíše náhodné číslo mesiaca (1..6) a žiačik musí napísať malými písmenami názov mesiaca. Program porovná zadaný názov s tým, ktorý má v pamäti a vypíše buď „SPRAVNE“ alebo „NESPRAVNE“. Umožnite žiakovi v cykle hádať ďalšie a ďalšie mesiace až kým sa nerozhodne skončiť napr. napísaním slova „STOP“. Pri riešení úlohy aspoň raz použite úplný podmienený príkaz.

POZRI NA INE RIESENIE

#include <stdio.h>

#include <string.h>

#include <time.h>

#include <stdlib.h>

int main(void)

{

int mesiac;

srand(time(0));

char nazov[10];

do {

mesiac = rand()%3+1;

printf("Cislo mesiaca je: %d, Napis jeho pomenovanie malymi pismenami:\n", mesiac);

scanf("%s", &nazov);

if ((strcmp(nazov, "STOP")) != 0)

{

switch(mesiac){

case 1: if ((strcmp(nazov, "januar")) == 0) printf("Spravne\n");

else printf("Nespravne\n");

break;

case 2: if ((strcmp(nazov, "februar")) == 0) printf("Spravne\n");

else printf("Nespravne\n");

break;

case 3: if ((strcmp(nazov, "marec")) == 0 ) printf("Spravne\n");

else printf("Nespravne\n");

break;

}

}

}

while((strcmp(nazov, "STOP")) != 0);

}

13.Základy – logické operácie

Vysvetlite pojmy: logická premenná, logické operácie, zložená podmienka

### Príklad

Vytvorte program na výpis pravdivostnej tabuľky zložených výrokov – negácie („nie je pravda, že“), konjunkcie („a“) a disjunkcie („alebo“) – podľa tabuľkovej prílohy.

* Logické hodnoty v tabuľke neopisujte z predlohy, ale nechajte počítač vyhodnotiť výsledok zložených podmienok, v ktorých budú vystupovať logické premenné A, B. Porovnajte tabuľku s vašim výpisom a objavte chybu.
* Stĺpce v tabuľke odsádzajte pomocou znaku tabulátora v príkaze výstupu.

### Tabuľková príloha

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| A | B | negacia A | A B | A alebo B |
| TRUE | TRUE | FALSE | TRUE | TRUE |
| TRUE | FALSE | FALSE | FALSE | TRUE |
| FALSE | TRUE | TRUE | FALSE | TRUE |
| FALSE | FALSE | TRUE | TRUE | FALSE |

#include <stdio.h>

#include <stdlib.h>

int main()

{

int A,B;

printf("\tA|\tB|\tnegacia A|\tA a B|\tA alebo B|\n");

A=1; B=1;

printf("\t%d|\t%d|\t\t%d|\t\t%d|\t%d|\n", A,B,!B,A&&B,A||B);

A=1; B=0;

printf("\t%d|\t%d|\t\t%d|\t\t%d|\t%d|\n", A,B,!B,A&&B,A||B);

A=0; B=1;

printf("\t%d|\t%d|\t\t%d|\t\t%d|\t%d|\n", A,B,!B,A&&B,A||B);

A=0; B=0;

printf("\t%d|\t%d|\t\t%d|\t\t%d|\t%d|\n", A,B,!B,A&&B,A||B);

system ("Pause");

return 0;

}

14. Cyklus - cyklus s pevným počtom opakovaní

Vysvetlite pojmy: cyklus s pevným počtom opakovaní, riadiaca premenná cyklu

*Príklad*

Napíšte program, ktorý vytvorí pravouhlý trojuholník v jednoduchom číselnom rade tak, aby vrchol začínal číslom „1“. Výšku trojuholníka (počet riadkov) zadáte na vstupe z klávesnice do premennej V. Výsledok vyzerá napr. takto (pre V=6):

1

2 3

4 5 6

7 8 9 10

11 12 13 14 15

16 17 18 19 20 21

#include <stdio.h>

int main()

{

int V, i, c, a = 1;

printf("Zadajte cislom pocet riadkov pre vysku trojuholnika\n");

printf("V = ");

scanf("%d", &V);

for (i = 1; i <= V; i++)

{

for (c = 1; c <= i; c++)

{

printf("%d ",a);

a++;

}

printf("\n");

}

system ("pause");

return 0;

}

15. Cyklus – porovnanie príkazov cyklu

Vysvetlite rozdiely medzi príkazom cyklu s pevným počtom opakovaní a príkazom cyklu s podmienkou. Ako sa rozhodneme, ktorý z nich použiť?

*Príklad*

Napíšte program, pomocou ktorého vypočítate N faktoriál (N!), kde N je na vstupe z klávesnice. Použite cyklus s pevným počtom opakovaní. Vytvorte na papieri krokovaciu tabuľku pre N=6 a vysvetlite pomocou nej ako cyklus funguje.

Ako by ste pretvorili program, keby ste mali použiť cyklus s podmienkou?

**1.spôsob:**

**#include <stdio.h>**

**#include <limits.h> /\* kniznica strazi zadanie cisla nad limit ULONG\_MAX \*/**

**int main(void)**

**{**

**int i, n;**

**double j;**

**i = 1;**

**j = 1;**

**printf("Zadaj cele kladne cislo 'n' pre vypocet factorialu:\n");**

**scanf("%d", &n);**

**while (i >= 1 && i <= n - 1) {**

**printf("%d x ",i++);**

**j \*= i;**

**if (j > ULONG\_MAX) { /\* kontroluje ci je prekroceny limit \*/**

**printf("%d\n\n!!! Mimo rozsah unsigned long!!!\n\n", i);**

**system("pause");**

**return 0;**

**}**

**}**

2.spôsob:

#include <stdio.h>

int main ()

{

int n,i;

long int fakt = 1;

printf("Zadej prirozene cislo n \n");

scanf("%d",&n);

for (i=2; i<=n; i++)

{

fakt \*= i;

}

printf(" %d! je %ld \n", n, fakt);

system("pause");

return 0;

}

--------------------------------------------------------------------------------------------------------------------------------------

3.spôsob:

#include <stdio.h>

int main ()

{

int n;

printf("Zadaj prirodzene cislo n \n");

scanf("%d",&n);

if (n>1) { int i=1;

long int fakt = 1;

while(i<=n)

{ fakt \*=i;

++i;

}

printf("faktorial %d je %ld \n", n, fakt);

}

system("pause");

return (0);

}

**16. Vysvetlite pojem podmienka, vysvetlite rozdiel medzi úplným a neúplným**

**vetvením*.***

*Príklad*

Na vstupe sú 3 navzájom rôzne celé čísla A, B, C, ktoré zadá užívateľ z klávesnice. Program navzájom porovná všetky 3 čísla a vypíše ich vzostupne.

Úlohu môžete vyriešiť tak, že pomocou neúplného podmieneného príkazu vypíšete všetkých 6 existujúcich možností. Ako by ste program pretvorili tak, aby ste využili úplné vetvenie cez vnorené podmienené príkazy?

**#include <stdio.h>**

**#include <stdlib.h>**

**int main()**

**{ /\*Porovnanie troch roznych cisel podmienkami\*/**

**int i,j,k;**

**printf("Zadaj tri rozne cisla:\n");**

**scanf("%d", &i);**

**scanf("%d", &j);**

**scanf("%d", &k);**

**printf("Zadal si: %d, %d a %d\n", i,j,k);**

**if (i<j && j<k) printf("Porovnanie: %d < %d < %d\n",i,j,k);**

**if (i<k && k<j) printf("Porovnanie: %d < %d < %d\n",i,k,j);**

**if (k<i && i<j) printf("Porovnanie: %d < %d < %d\n",k,i,j);**

**if (k<j && j<i) printf("Porovnanie: %d < %d < %d\n",k,j,i);**

**if (j<i && i<k) printf("Porovnanie: %d < %d < %d\n",j,i,k);**

**if (j<k && k<i) printf("Porovnanie: %d < %d < %d\n",j,k,i);**

**system ("pause");**

**return 0;**

**}**

**17.Cyklus – prerušenie cyklu**

**Vysvetlite pojmy: cyklus, preskočenie krokov v cykle (skok), predčasné ukončenie cyklu (prerušenie).**

**Príklad**

**Napíšte program, ktorý bude simulovať vkladanie PIN kódu do bankomatu. Počítač bude pomocou príkazu cyklu s pevným počtom opakovaní načítavať od užívateľa celočíselný 4-ciferný PIN kód (ak nezadá 4-ciferné celé číslo, musí zadať znova). Ak zadá správny kód, cyklus sa predčasne ukončí. Po treťom nesprávnom zadaní cyklus skončí.**

#include <stdio.h>

#include <stdlib.h>

int main(void)

{

int PIN, pokus=1;

do {

printf("Zadaj 4-ciferny PIN kod:\n");

scanf("%d", &PIN);

if (PIN < 1000 || PIN > 9999){

printf("Nezadal si 4-ciferne cislo\n");

}

if (PIN != 4444){

printf("Pokus c. %d\n", pokus);

pokus += 1;

continue;

}

else if(PIN == 4444){

printf("Spravny PIN \n");

break;

}

}

while(pokus <= 4);

printf("Vycerpal si pokusy \n");

system("pause");

}

**18. Cyklus – porovnanie cyklov**

Vysvetlite význam a použitie cyklu for a cyklu while.

Príklad

Napíšte program, ktorý bude načítavať ceny maximálne 20 položiek (minimálna hodnota jednej položky bude 1 Euro) a na konci vypíše celkovú sumu za nákup. Zadávanie možno predčasne ukončiť zadaním 0. Nastavte program tak, aby po prekročení sumy 100 Eúr, odrátal za každú ďalšiu položku prémiovú zľavu 1 Euro. Použite príkaz while.

Ako by ste pretvorili program, keby ste mali použiť cyklus for?

**#include <stdio.h>**

**int main(void)**

**{**

**int cena, SUMA=0, poradie=1;**

**while(poradie <= 20)**

**{**

**printf("Zadaj cenu za %d polozku\n", poradie);**

**scanf("%d", &cena);**

**if (cena == 0) goto x;**

**else if (cena != 0) {**

**poradie++;**

**SUMA += cena;**

**printf("Priebezna suma %d", SUMA);**

**if (SUMA > 100 && cena != 0){**

**SUMA -= 1;**

**}**

**}**

**printf("Poradie: %d", poradie);**

**}**

**x:**

**printf("Celkova suma za 20 poloziek je: %d \n", SUMA);**

**system("pause");**

**}**

19. Polia – jednorozmerné pole reálnych čísel

Vysvetlite pojmy: jednorozmerné pole, prvok poľa, index prvku. Akým spôsobom deklarujeme pole reálnych čísel?

*Príklad*

Napíšte jednoduchý program, ktorý ilustruje definíciu a použitie jednorozmerného poľa reálnych čísel. Na vstupe z klávesnice sú dve polia po 4 reálne čísla:

* pole A (obsahujúce ceny výrobkov v Eurách)
* pole B (obsahujúce množstvo v kg).

Vypíšte na obrazovku obe polia do 2 samostatných riadkov, tak aby súvisiace hodnoty boli pod sebou. Vypočítajte celkovú sumu výrobkov a vypíšte ju na obrazovku

#include <stdio.h>

#include <stdlib.h>

int main(void)

{

float poleA[4], poleB[4];

int i;

float suma=0;

for(i = 0; i <= 3; i++){

printf(" Cena za %d polozku: ", i+1);

scanf("%f", &poleA[i]);

printf("Mnozstvo v kg za %d polozku: ", i+1);

scanf("%f", &poleB[i]);

}

printf("\n CENY: ");

for(i = 0; i <= 3; i++)

{

printf("%.2f\t", poleA[i]);

}

printf("\nMNOZSTVA: ");

for(i = 0; i <= 3; i++)

{

printf("%.2f\t", poleB[i]);

}

for(i = 0; i <= 3; i++)

{

suma = suma + poleA[i]\*poleB[i];

}

printf("\n\nCELKOVA SUMA JE: %.2f\n", suma);

system("pause");

}

20.Polia – dvojrozmerné celočíselné pole

Vysvetlite pojmy: dvojrozmerné pole, prvok poľa, indexy prvku. Akým spôsobom deklarujeme celočíselné pole?

*Príklad*

Definuj pole o 10 x 10 prvkoch celočíselného dátového typu. Naplň toto pole vo vnorenom cykle výsledkami násobenia celých čísel od 1 po 10. Vypíš toto pole na obrazovke ako tabuľku malej násobilky od 1 do 10 (10 riadkov x 10 stĺpcov).

#include <stdio.h>

#include <stdlib.h>

int main(void)

{

int poleA[10][10];

int i, j;

for (i = 1; i <= 10; i++){

for (j = 1; j <= 10; j++){

poleA[i][j] = i\*j;

}

}

for(i = 1; i <= 10; i++){

for (j = 1; j <= 10; j++){

printf(" %.2d,", poleA[i][j]);

}

printf("\n");

system("pause");

}

}

21.Základy – príkaz výstupu

Vysvetlite pojmy: príkaz výstupu, textová obrazovka. Ako dokážeme výpis presunúť o pozíciu tabulátora doprava? Ako dokážeme výpis presunúť do nového riadku.

*Príklad*

AHOJ

AHOJ

AHOJ

AHOJ

AHOJ

Zostavte program, ktorý načíta od užívateľa číslo N a vypíše na obrazovku počítača v textovom režime N-krát slovíčko AHOJ a to tak, aby každé nasledujúce slovíčko, bolo posunuté o jeden riadok dolu a o jednu medzeru vpravo (podľa obr.)

Pozmeňte program tak, aby sa každé ďalšie slovíčko posúvalo o X znakov vpravo a Y riadkov dolu, kde X a Y zadáte na vstupe.

#include <stdio.h>

int main(void)

{

int riadok, stlpec, i, N, X, Y;

printf("Zadaj pocet vypisu AHOJ:\n");

scanf("%d",&N);

printf("Zadaj pocet pre X-ovy posun:\n");

scanf("%d",&X);

printf ("Zadaj pocet pre Y-ovy posun:\n");

scanf("%d",&Y);

for (riadok = 1; riadok <= N; riadok++)

{

for (stlpec = 1; stlpec <= riadok; stlpec++)

{

for (i = 1; i <= X ; i++) printf(" ");

}

printf ("AHOJ");

for(i = 1; i <= Y ; i++) printf("\n");

}

system("pause");

return 0;

}

22.Aritmetika - náhodné čísla

Vysvetlite pojmy: pole, prvok poľa, náhodné číslo, generátor náhodných čísel.

*Príklad*

V kasíne si chceli overiť „správnosť“ softvérového generátora náhodných čísel nastaveného na hracom automate simulujúcom hru v kocky. Zaujímalo ich či výherné číslo 6 nenabieha príliš často. Napíšte program, ktorý im v tom pomôže. Program na začiatku vygeneruje a vypíše na obrazovke monitora vedľa seba 40 celých čísiel z intervalu <1,6>.

Vypočítajte a vypíšte koľkokrát padlo číslo 6. Vypíšte na jedno desatinné miesto aj percentuálne zastúpenie čísla 6 vzhľadom k celkovému počtu čísel.

#include <stdio.h>

#include <stdlib.h>

#include <time.h> /\* knižnica potrebna ku generatoru \*/

int main(void)

{ const int N = 40;

int i, sestka, p;

float perc;

srand(time(0)); /\* inicializacia generatora nahodnych cisiel \*/

p=0;

printf ("Generovane cisla su:\n");

for (i = 1; i <= N; i++)

{

sestka = rand()%6 + 1; /\* rozsah 1 až 6 \*/

printf ("%d \t", sestka);

if (sestka==6 ) {

p=p+1;

}

}

printf ("Pocet sestiek je %d \n", p);

perc=(float(p)/float(N)) \* 100;

printf ("Percento sestiek je %.1f percent. \n", perc);

system("pause");

return 0;

}

23.Reťazce – kódovanie znakov

Vysvetlite pojmy: znak, znaková premenná, ASCII tabuľka

*Príklad*

Vytvorte jednoduchý šifrovací program. Program zapisuje nezašifrovaný text zložený z veľkých písmen po znakoch,  pri šifrovaní posunie každé písmeno na jeho nasledujúce v abecede (A->B,…. Z->A) o jednu pozíciu, výsledný zašifrovaný text zobrazí na monitore počítača. Doplňte program tak, aby posúval pri šifrovaní každé písmeno –znak o dve (A->C,…. Z->B) , o tri (A->D,…. Z->C), resp. o určený počet pozícií.

Všetky ostatné nepísmenové znaky ponechá presne tak ako sú. Aký typ cyklu použijete?

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

int main()

{

char text\_zasifrovany[100], text[100];

//char text[100], text\_zasifrovany[] = "ujpsd itrfhn zptq";

int i, j, c;

printf("Zadajte zasifrovany text:\n ");

gets(text\_zasifrovany); /\*text budem chapat ako retazec znakov (char) \*/

for (j = 1; j <= 10; j++) {

for (i = 0; i < strlen(text\_zasifrovany); i++)

{

c = text\_zasifrovany[i] + j;

if (c < 'a' && c != ' ' + j)

c += 26;

if (text\_zasifrovany[i] == ' ')

c = ' ';

text[i] = c;

}

text[i + 1] = '\0';

printf("\"%s\" -> posun o %d dozadu\n\n", text, j);

}

system("pause");

return 0;

}

24. Reťazce – práca s reťazcami po znakoch

Vysvetlite pojmy: znak, znaková premenná, reťazec

Napíšte program, ktorý načíta text telegramu zadaný veľkými písmenami a cenu za jedno slovo. Nahraďte všetky medzery v telegrame hviezdičkou a vypíšte zmenený text na obrazovku. Nakoniec vypíšte sumu, ktorú by platil odosielateľ.

*Príklad výpisu:*

výpis: Zadaj text:

vstup: DNES JE MATURITNA SKUSKA

výpis: Zadaj cenu za slovo

vstup: 0.50

výstup:DNES\*JE\*MATURITNA\*SKUSKA

Zaplatite: 2.00 eur

#include <stdio.h>

#include <stdlib.h>

int main()

{

int pocet=0;

float cena=0.5, spolu;

char slovo;

printf("Zadaj text:\n");

do

{

slovo=getch();

putch(slovo);

if (slovo==' ') //ak po slove nasleduje medzera pripocitaj slovo

++pocet;

} while (slovo!=13);

++pocet;

printf("Pocet slov ve vete je %i\n",pocet);

getch();

printf("cena jedno slovo %f\n",cena);

spolu=cena\*pocet;

printf("Cena za telegram %f:\n", spolu);

system("pause");

return 0;

}

**25. : Podmienený príkaz – viacnásobné vetvenie**

vysvetlite pojem vetvenie, viacnásobné vetvenie, prepínač

*Príklad*

Vytvorte program pre jednoduchú kalkulačku s operáciami: sčítanie, odčítanie,

násobenie a delenie. V programe najprv načítajte z klávesnice prvé celé číslo, potom zadajte aritmetickú operáciu do znakovej premennej (použite znaky +, -, \*, / ), nakoniec zadajte druhé celé číslo.

Po zadaní vstupov nech program vypočíta pomocou príkazu viacnásobného vetvenia výsledok a vypíše ho na obrazovku.

#include <stdio.h>;

int main(void)

{

int a, b;

char z; /\* znaky pre aritmeticke operacie su: +, -, \*, / \*/

printf("Zadaj cislo:\n");

scanf("%d", &a);

printf("Zadaj aritmeticku operaciu\n");

getchar();

z = getchar();

switch (z){

case '+': printf("Zadaj dalsie cislo:\n");

scanf("%d", &b);

printf("%d + %d = %d", a, b, a + b);

break;

case '-': printf("Zadaj dalsie cislo:\n");

scanf("%d", &b);

printf("%d - %d = %d", a, b, a - b);

break;

case '\*': printf("Zadaj dalsie cislo:\n");

scanf("%d", &b);

printf("%d \* %d = %d", a, a, a \* b);

break;

case '/': printf("Zadaj dalsie cislo:\n");

scanf("%d", &b);

printf("%d / %d = %d", a, b, a / b);

break;

default: printf("Zadal si zle\n");

break;

}

system("pause");

}

26. Aritmetika – celočíselná a reálna premenná

Vysvetlite pojmy celočíselná a reálna premenná a ako sa deklarujú v jazyku C. Vymenujte základné matematické operácie pre oba typy premenných.Napíšte program, v ktorom si používateľ zvolí, či chce počítať objem, povrch kvádra, alebo jednu zo strán kvádra, ak pozná zvyšné dve a objem. Podľa zvolenej možnosti užívateľ zadá požadované údaje v metroch a počítač hľadanú hodnotu vyráta. Výsledky najprv vypíšte skráteným výpisom na 2 desatinné miesta. Potom ich zaokrúhlite a vypíšte po zaokrúhlení. (V=a.b.c, S=2ab+2ac+2bc)

*Príklad výpisu:*

výpis: Zadaj co chces pocitat: (a)Objem kvadra (b)Povrch kvadra (c)Stranu kvadra

vstup: b

výpis: Zadaj rozmery kvadra a, b, c:

vstup: 5.1 2.4 3.5

výstup:S = 42.84 po zaokruhleni S = 43

#include <stdio.h>

#include <stdlib.h>

int main(void)

{

char z;

float a, b, c, objem, povrch, strana, vyska;

printf("Zvol co chces pocitat: (a) Objem kvadra (b) Povrch kvadra (c) Strana kvadra\n");

scanf("%c", &z);

if (z == 'a'){

printf("Zadaj rozmery kvadra a, b, c\n");

scanf("%f%f%f", &a, &b, &c);

objem = a \* b \* c;

printf("V= %.2f po zaokruhleni V = %.0f", objem, objem);}

if (z == 'b'){

printf("Zadaj rozmery kvadra a, b, c\n");

scanf("%f%f%f", &a, &b, &c);

povrch =(a\*b)\*2 + (a\*c)\*2 + (b\*c)\*2;

printf("S= %.2f po zaokruhleni S = %.0f", povrch, povrch);

}

else if (z == 'c'){

printf("Zadaj objem a vysku kvadra \n");

scanf("%f%f", &objem, &vyska);

strana,strana =objem/vyska; /\* vzorec pre vypocet strany je a2 = V/c \*/

printf("c= %.2f po zaokruhleni S = %.0f", strana, strana);

}

system("pause");

return 0;

}

**27. Cyklus – porovnanie cyklov**

Vysvetlite pojem cyklus, akými spôsobmi sa deklaruje v jazyku C, vysvetlite podmienky behu cyklu pre jednotlivé prípady.

*Príklad*

Napíšte program pre konverziu čísla z desiatkovej číselnej sústavy (base-10) do dvojkovej číselnej sústavy (základ-2), veľkosť čísla má 32 bitov. Na vstupe zadáte celé číslo v desiatkovej sústave a výstup bude v binárnom kóde.

**#include <stdio.h>**

**int main()**

**{**

**int n, c, k;**

**printf("Zadaj cislo v desiatkovej sustave:\n");**

**scanf("%d", &n);**

**printf("%d, Prevod cisla do dvojkovej sustavy:\n", n);**

**for (c = 31; c >= 0; c--)**

**{**

**k = n >> c;**

**if (k & 1)**

**printf("1");**

**else**

**printf("0");**

**}**

**printf("\n");**

**system("pause");**

**return 0;**

**}**

**28. Cyklus – riadiaca premenná cyklu**

Vysvetlite pojmy: cyklus, riadiaca premenná cyklu. Akým spôsobom dokážeme v príkaze cyklu s podmienkou doplniť neexistujúcu riadiacu premennú cyklu?

*Príklad* : Vytvorte program, ktorý na začiatku načíta od užívateľa kladné celé číslo väčšie ako 1 a na konci vypíše či je toto číslo prvočíslo (deliteľné len samo sebou a číslom 1) alebo zložené číslo (deliteľné aspoň jedným ďalším číslom).

Doplňte program tak, že v prípade zadania nevyhovujúceho čísla (menšieho ako 2), musí užívateľ zadaj znova. Po treťom chybnom zadaní nech program skončí.

#include <stdio.h>

#include <stdlib.h>

int main(void) {

int a, i, JEPRVOCISLO;

JEPRVOCISLO=1;

i=2;

do {

printf("Zadajte cele cislo vacsie ako 1: \n");

scanf("%d",&a);

}

while (a<2);

while (i<a && JEPRVOCISLO==1) {

if (a%i == 0) JEPRVOCISLO=0;

i++;

}

if (JEPRVOCISLO==1) {

printf("Je prvocislo.");

}

else { printf("Je zlozene cislo.");

}

system ("pause");

return 0;

}

29. Cykly – porovnanie cyklov

Vysvetlite rozdiely medzi príkazom cyklu s pevným počtom opakovaní a príkazom cyklu s podmienkou. Ako sa rozhodneme, ktorý z nich použiť?

*Príklad*

Naprogramujte hru „Hádaj číslo“. Počítač zvolí náhodné celé číslo z intervalu 1..100 a priradí ho do premennej ***los***. Hráč vždy dookola v cykle zadáva číslo do premennej ***tip***, počítač mu vždy na obrazovke vypíše či uhádol a potom hráč zadáva znova. Hra končí uhádnutím čísla alebo zadaním čísla mimo intervalu 1..100. Doplňte program tak, že počítač vždy hráčovi poradí, či je hľadané číslo ***los*** menšie alebo väčšie ako zadané ***tip***.

#include <stdio.h>

int main()

{

int los,tip,i=0;

srand( (unsigned) time(NULL) ); //vygenerovane cislo bude vzdy ine

los = (rand() % 100); //hladane cislo bude v rozsahu 0-100

do

{

printf(" Uhadni cislo:\n");

scanf("%d",&tip);

if (tip > los)

printf("Hladane cislo je mensie\n");

else if (tip < los)

printf("Hladane cislo je vetsie\n");

i++;

} while (los!=tip);

printf("Cislo si uhadoll na %d.pokus\n", i);

getch();

return 0;

}

30. Jednorozmerné pole - triedenie

Vysvetlite usporiadanie položiek v jednorozmernom poli a postupnosť pri ich zoraďovaní pomocou triedenia (vysvetlite princíp bublinkového triedenia).

*Príklad*

Vytvorte program, ktorý načíta jednotlivé prvky poľa, prvky sú celočíselného

datového typu a bude ich maximálne desať. Program usporiada postupnost

čísel v neklesajucom rade jednoduchým bublinkovým triedením. Prvky usporiadanej postupnosti vypíše s ich idexom. V programe uplatnite podmienečný príkaz a jednoduchý cyklus.

**#include <stdio.h>**

**#define MAX 10**

**main()**

**{ int pocet;**

**int i,j;**

**int a[MAX],pomoc;**

**printf("\n Zadaj pocet prvkov postupnosti: ");**

**scanf("%d",&pocet);**

**if(pocet>MAX)**

**printf("\n To je prilis mnoho, maximum je %d\n",MAX);**

**else /\* pocet prvkov bol programom akceptovany \*/**

**{**

**for(i=0;i<pocet;i++) { /\* citanie prvkov postupnosti \*/**

**printf("a[%d] : ",i+1);**

**scanf("%d",&a[i]);**

**}**

**/\* algoritmus jednoducheho bublinkoveho triedenia : \*/**

**for(i=pocet-1;i>0;i--)**

**for(j=0;j<=i-1;j++)**

**if(a[j]>a[j+1])**

**{ pomoc=a[j]; a[j]=a[j+1]; a[j+1]=pomoc;**

**}**

**for(i=0;i<pocet;i++) { /\* vypis usporiadanej postupnosti \*/**

**printf("\na[%d] : ",i+1);**

**printf("%d",a[i]);**

**system("Pause");**

**}**

**} /\* koniec else \*/**

**}**

Reťazce – práca s reťazcami po znakoch ( práca so súborom txt )

Vysvetlite pojmy: znak, znaková premenná, reťazec

Napíšte program, ktorý načíta text telegramu zadaný veľkými písmenami a cenu za jedno slovo. Nahraďte všetky medzery v telegrame hviezdičkou a vypíšte zmenený text na obrazovku. Nakoniec vypíšte sumu, ktorú by platil odosielateľ.

*Príklad výpisu:*

výpis: Zadaj text:

vstup: DNES JE MATURITNA SKUSKA

výpis: Zadaj cenu za slovo

vstup: 0.50

výstup:DNES\*JE\*MATURITNA\*SKUSKA

Zaplatite: 2.00 eur

#include <stdio.h>

#include <string.h>

int main(void)

{

FILE \*f;

float cena=0;

int c, subor=0;

f = fopen("TELEGRAM.txt", "r");

while ((c= getc(f)) != EOF)

{

subor++;

}

fclose(f);

int upravsubor = 0;

f = fopen("TELEGRAM.txt", "r");

char string[subor];

while ((c= getc(f)) != EOF){

if (c == ' '){

char zmena = '\*';

string[upravsubor] = zmena;

putchar(zmena);

cena += 0.5;

}

else

string[upravsubor] =c;

putchar(c);

upravsubor++;

}

f = fopen("TELEGRAM.txt", "r");

printf("\nZaplatite: %.2f\n", cena+0.5);

system("pause");

}