

Základy programovania v Pascale pomocou príkladov 2004

1. a) Napíšte program, ktorý pomocou cyklu vypíše malú abecedu.

```
program molA;
var znak:char;
begin
for znak:='a' to 'z' do write(znak:2);
{znak:2 znamena, ze jeden znak vypise na dve miesta}
readln;{caka na [ENTER] }
end.
```

b) Napíšte program, ktorý zistí, či tri zadané čísla na vstupe môžu byť stranami pravouhlého trojuholníka.

```
program molB;
var a,b,c,x:real;
begin
writeln('zadaj strany trojuholnika');
readln(a,b,c);
{najskuor si hodnoty a,b,c pekne usporiadame aby
platilo a >= b >= c !}
  repeat
    if a<b then begin
      x:=a;
      a:=b;
      b:=x;{x je akoze a}
    end;

    if b<c then begin
      x:=b;
      b:=c;
      c:=x;{x je akoze b}
    end;

  until (a>=b) and (b>=c); {to 'and' je logicka operacia}
{konec usporiadania}

{a*a muozme napisat aj ako sqr(a) }
if (a*a)=(b*b+c*c) then writeln('Bravo, trojuholnik je pravouhly')
  else if a < (b+c) then writeln('Je to len obycajny trojuholnik')
    else writeln('Smola to nieje trojuholnik');
{Je tu viacnasobna podmienka, treba sa na to kuknut pozornejsie}
readln;
end.
```

2. a) Na vstupe máme nejakú vetu. Napíšte program, ktorý zistí, koľko je v danej vete číslic.

```
program mo2A;
var znak:char;
    pocet:integer;
begin
writeln('Zadaj vetu, v pohode. ');
pocet:=0;

repeat
read(znak);
if (znak>='0') and (znak<='9') then inc(pocet);
    { inc(x) znamena ze pramennu x zvacsi o jednotku }
until znak='.'; {ta bodka je akoze koniec vety}

writeln('Vo vete je ',pocet,' cislic(a)');
readln;
readln;{readln je tu 2 krat}
end.
```

b) Simulujte cyklus FOR pomocou WHILE.

```

program mo2B;
var I:integer;
begin
writeln('vypiseme si cisielka');
for I:=1 to 9 do write(I:2);
writeln('');{skoci do noveho riadku}

{a teraz to iste pomocou while}
I:=0;
while I<=8 do begin {je to podmienka}
    inc(I);{to je vlastne I:=I+1}
    write(I:2);
end;

readln;
end.

```

3. a) Napíšte program, ktorý premení čas zadaný v sekundách na celé hodiny, minúty a sekundy.

```

program mo3A;
var vstup:longint;
    h,m,s:integer;{vystup}
begin
writeln('zadaj sekundy');
readln(vstup);
h:=vstup div 3600;{div celociselne delenie}
vstup:=vstup mod 3600;{mod zvisok po deleni 3600}

m:=vstup div 60;
vstup:=vstup mod 60;

s:=vstup;
writeln('Predstavuje to ',h:4,' hod, ',m:3,' min a ',s:3,' sekund');
readln;
end.

```

b) Simulácia dvojrozmerného poľa: máme 10 riadkov po 5 znakov, napíšte program, ktorý všetky riadky posunie o jeden riadok nižšie a na miesto prvého riadku dá riadok posledný.

```

program mo3B;
type pole=array[1..5,1..10] of char;
{je to ako keby na ciselnej osi x a y}
    pomocne_pole=array[1..5] of char;
var matica:pole;
    help:pomocne_pole;
    x,y:shortint;
begin
    for y:=1 to 10 do begin
        for x:=1 to 5 do begin
            write(' m[' ,x:1,' ,',y:2,']= ');
            readln(matica[x,y]);
            end;
            writeln;
            end;
    {mame nacistane to pole}

    for y:=1 to 10 do begin
        for x:=1 to 5 do begin
            write(matica[x,y]:4);
            end;
            writeln;
            end;
    writeln('vysledne pole po posunuti');

```

Dalo by sa to i na oko jednoduchšie, tak že:

```

type
riadok=array[1..5] of char;
pole=array[1..10] of riadok;
var m:pole;
    a vymenili by sme len riadky, ale
    načítanie by bolo takmer to isté:
    readln(pole[x][y]);

```

```

{posunieme pole}
for x:=1 to 5 do begin
    help[x]:=matica[x,10];
    {nacistame posledny riadok pola}
end;

for y:=10 downto 2 do begin
    for x:=1 to 5 do begin
        matica[x,y]:=matica[x,y-1];
        end;
    end;

for x:=1 to 5 do begin
    matica[x,1]:=help[x];
    {napiseme prvvy riadok pola}
end;

{-----vypiseme vysledok-----}
for y:=1 to 10 do begin
    for x:=1 to 5 do begin
        write(matica[x,y]:4);
        end;
        writeln;
        end;

readln;
end.

```

4. a) Napíšte program, ktorý zistí, či na vstupe je číslo deliteľné trojkou a či je párne.

```

program mo4A;
var vstup:longint;
begin
writeln('zadaj cislo, cele');
readln(vstup);
if vstup=0 then writeln('To je nula')
else begin
    if (vstup mod 3)=0 then writeln('cislo je delitelne trojkou (3)')
    else writeln('cislo nieje delitelne trojkou (3)');
    if (vstup mod 2)=0 then writeln('cislo je parne')
    else writeln('cislo nieje parne');
end;

readln;
end.

```

b) Napíšte program, ktorý zistí, či číslo na vstupe je prvočíslo.

```

program mo4B;
var vstup:longint;
    p,c,I:integer;
begin
writeln('Zadaj cislo vacsie ako 0');
readln(vstup);
p:=0;
    case vstup of
        1: begin
            writeln('specificke nezaradene cislo');
            halt;{halt okamzite skonci cely program}
            end;
    end;

if vstup>2 then begin
    {toto je matematicke pravidlo}
    c:=trunc(sqrt(vstup)+1);
    {sqrt = odmocnina druha
    trunc = odreze desatinu cast, teda
    napr. 45.26 -> 45 }
end;

```

```

        for I:=2 to c do begin
            if (vstup mod I)=0 then inc(p);
            end;
        end;
    if p=0 then writeln('toto cislo je prvocislo')
        else writeln('toto nieje prvocislo');
    readln;
end.

```

5. a) Máme pole typu n x n. Urobte program, ktorý ho otočí o 90 stupňov.

```

program mo5A;
const n=3;
type pole=array[1..n,1..n] of char;

var matica,otocene_pole:pole;
    x,y:integer;
begin
    {nacistame pole znakov}
    for y:=1 to n do begin
        for x:=1 to n do begin
            {tento komplikovany zapis nemusis rozoberat, funguje}
            write('matica[',x:2,',',y:2,']= ');
            readln(matica[x,y]);
            end;
            writeln;
            end;

    {vypiseme pole puovodne}
    for y:=1 to n do begin
        for x:=1 to n do begin
            write(matica[x,y]:3);
            end;
            writeln;
            end;

    {teraz pole otocime o 90stupnov proti smeru hodinovych ruciciek}
    for y:=1 to n do begin
        for x:=1 to n do begin
            otocene_pole[x,y]:=matica[n+1-y,x];
            end;
            end;

    {vypiseme pole po otoceny}
    writeln('po otoceny');
    for y:=1 to n do begin
        for x:=1 to n do begin
            write(otocene_pole[x,y]:3);
            end;
            writeln;
            end;

    readln;
end.

```

b) Napíšte program, ktorý vynásobí dve čísla bez toho, aby použil násobenie.

```

program mo5B;
var zaklad,sucin:real;
    cinitel,I:integer;
begin
    writeln('zadajte cislo, ktore chcete nasobit - realne');
    readln(zaklad);

```

```
writeln('zadajte cislo, ktorim chcete nasobit - cele');
readln(cinitel);
sucin:=0;
{if (cinitel=0) or (zaklad=0) then
  else - tento zapis nahradime}
if (cinitel<>0) and (zaklad<>0) then
  for I:=1 to abs(cinitel) do begin
    sucin:=sucin+zaklad;
  end;
if cinitel<0 then sucin:=-1*sucin;
{funkcia abs(x) je absolutna hodnota z x}

writeln('Sucin je ',sucin:10:10);
{zapis sucin:10:10 znamena, ze vysledok napise na 10 celych a na
  10 desatinych miest ho zaokruhli}
readln;
end.
```

6. a) Napíšte procedúru, ktorá vykreslí štvorec z hviezdíček.

```
program mo6A;
var x,y,a:shortint;

procedure stvorec(x,y,a:shortint);{y je tam len pre krasu}
var I,M:shortint;
begin
  for I:=1 to a do begin
    {urobime odsadenie}
    for M:=1 to x do write(' ');
    {nakreslime stvorec}
    for M:=1 to a do write('* ');
    if I<>a then writeln;{vhupneme do dalsieho riadku}
  end;
end;

begin
writeln('zadajte odsadenie a hranu stvorca / obrazovka je 80*25 znakov');
readln(x,a);
stvorec(x,0,a);
readln;
end.
```

b) Dané sú súradnice bodu A. Napíšte program, ktorý zistí, či tento bod sa nachádza na priamke

$$ax + b = 0.$$

```
program mo6B;
type vektor=array[1..2] of real;
var A:vektor;
    aa,bb:real;
begin
writeln('zadajte suradnice bodu A, Ax a Ay');
readln(A[1],A[2]);
writeln('zadajte parametre pre priamku ax+b=0, teda a & b');
readln(aa,bb);
if (aa*A[1]+bb)=A[2] then writeln('Dany bod lezi na priamke.')
  else writeln('Dany bod nelezi na priamke');
readln;
end.
```

7. a) Napíšte funkciu, ktorá z dvoch čísel vyberie menšie z nich.

```
program mo7A;
var a,b:real;
function vyber(a,b:real):real;{pascalu je jedno ci su tie nazvy
  rovnake ako v hlavnom programe}
```

```
begin
  if a<=b then vyber:=a
    else vyber:=b;
end;
```

```
begin
  writeln('zadajte dve lubuvolne realne cisla');
  readln(a,b);
  writeln('mensie je -> ',vyber(a,b):10:10);
  readln;
end.
```

b) Je daný interval $\langle a, b \rangle$. Napíšte program, ktorý sčíta všetky čísla z tohto intervalu deliteľné trojkou.

```
program mo7B;
var a,b,x,I:integer;
    sucet:longint;
begin
  writeln('zadajte rozmedzie intervalu <a,b>');
  readln(a,b);
  {musime si skontrolovať veľkosť čísel a & b}
  if a>b then begin
    x:=a;
    a:=b;
    b:=x;
  end;
  sucet:=0;
  for I:=a to b do begin
    if (I mod 3)=0 then sucet:=sucet+I;
  end;
  writeln('konečný súčet je ',sucet:10);
  readln;
end.
```

8. a) Napíšte funkciu, ktorá zistí, či na vstupe je číslo alebo znak.

```
program mo8A;
var vstup:char;

function je_to_cislo(vstup:char):boolean;
begin
  if (vstup>='0') and (vstup<='9') then je_to_cislo:=true
    else je_to_cislo:=false;
end;

begin
  writeln('zadaj znak');
  readln(vstup);
  writeln('Je to číslo?> ',je_to_cislo(vstup));
  readln;
end.
```

b) ... na výber minimálneho čísla z troch čísel.

```
program mo8B;
var a,b,c,min:real;

begin
  writeln('zadaj tri čísla');
  readln(a,b,c);
  if a<=b then min:=a
    else min:=b;
  if c<min then min:=c;

  writeln('minimum je ',min:10:10);
  readln;
end.
```

9. a) Napíšte program, pomocou ktorého vypočítate objem a povrch valca.

```

program mo9A;
var v,r,povrch,objem:real;
begin
writeln('Zadajte vysku valca v a polomer podstavy r');
readln(v,r);
povrch:= v*2*PI*r + 2*PI*sqr(r);
objem:= PI*sqr(r)*v;
writeln('Povrch valca je ',povrch:10:10,' j2');
writeln('Objem valca je ',objem:10:10,' j3');
readln;
end.

```

b) Nakreslite vývojový diagram na premenu sekúnd na hodiny, minúty a sekundy. viz [MO] 3a**10. a) Napíšte program, ktorý vypočíta n!.**

```

program mo10A;
var vstup:integer;
{Pozor, netreba sa nicoho bat, pouzijem len obycajnu
funkciu. A preco by som nemohol volat aj viackrat tuto
funkciu? Da sa to a vysledkom je rekurzia}

function faktorial(n:integer):longint;
begin
if (n=1) or (n=0) then faktorial:=1
else faktorial:=n*faktorial(n-1);
end;

begin
writeln('zadajte n (len do 12!!)');
readln(vstup);
if vstup<0 then halt;{okamzite skonci program}
writeln('n!= ',faktorial(vstup):30);
readln;
end.

```

b ...na výpočet objemu a povrchu gule.

```

program mo10B;
var r,povrch,objem:real;
begin
writeln('Zadajte polomer r gule');
readln(r);
povrch:=4*PI*sqr(r);
objem:= (4/3)*PI*r*r*r;
writeln('Povrch gule je ',povrch:10:10,' j2');
writeln('Objem gule je ',objem:10:10,' j3');
readln;
end.

```

11. a) Napíšte program, ktorý z n čísel na vstupe vyberie minimum.

```

program mollA;
var n,I:integer;
min,cislo:real;
kontrolka:boolean;{dobre si to pozri}
begin
writeln('zadaj pocet cisel n');
readln(n);
if n<1 then halt;
writeln('zadaj cisla');
kontrolka:=false;
for I:=1 to n do begin
if kontrolka=false then begin {prvy krat je v cykle}

```

```

        readln(min);
        end;
        if kontrolka<>false then begin
            readln(cislo);
            if cislo<min then min:=cislo;
            end;
        kontrolka:=true;
        end;
writeln('Minimum z ',n:2,' cisel je ',min:5:5);
readln;
end.

```

b) Nakreslite vývojový diagram na výpočet n faktoriálu.

viz [MO] 10a

12. a) Napíšte program, ktorý z n čísel na vstupe vypočíta aritmetický priemer.

```

program mol2A;
var n,I:integer;
    priemer,cislo:real;
begin
writeln('zadaj pocet cisel n');
readln(n);
if n<1 then halt;
writeln('zadaj cisla');
priemer:=0;
for I:=1 to n do begin
    readln(cislo);
    priemer:=priemer+cislo;
end;
priemer:=priemer/n;
writeln('Aritmeticky priemer z ',n:2,' cisel je ',priemer:10:5);
readln;
end.

```

b) Nakreslite vývojový diagram na simuláciu cyklu FOR pomocou cyklu WHILE.

viz [MO] 2b

13. a) Napíšte program, ktorý sa ukončí, keď stlačíme ESC a bude reagovať na stlačenie kurzorových šípok, napr. po stlačení šípky hore napíše hore a podobne.

```

program mol3A;
uses crt;
var klavesa:char;
begin
repeat
klavesa:=readkey;{readkey je funkcia kniznice crt}
    case klavesa of
    #72:write(' ▲ ');{vlozil som to takto: [Alt]+30}
    #80:write(' ▼ ');{[Alt]+31}
    #75:write(' ',chr(17),' ');{v lavo}
    #77:write(' ',chr(16),' ');{v pravo}
    #32:write(' medzera');
{Esc} #27:halt;
    {mohli by sme tu este vsunut riadok
    else pre ostatne klavesy}
    end;
until klavesa=#27;
end.

```

b) Nakreslite vývojový diagram na výpočet aritmetického priemeru z n čísel.

viz [MO] 12a

14. a) Napíšte program, ktorý usporiada tri čísla od najmenšieho po najväčšie.

viz [MO] 1b – časť toho programu usporadúva čísla od najmenšieho po najväčšie

b) ... na výpočet a^n .

```

program mol4B;
var a,vysledok:real;
    n,I:longint;
begin
writeln('zadajte a & n pre vypocet a^n | (a) muoze byt realne');
readln(a,n);
vysledok:=1;
if (a=0) and (n>0) then vysledok:=0; {specificky pripad}
if a<>0 then begin {aby nedoslo k deleniu nulou}
    if n>0 then for I:=1 to n do begin
        vysledok:=vysledok*a;
    end;

    if n<0 then for I:=1 to abs(n) do begin
        vysledok:=vysledok/a;
    end;

    end;
if (a=0) and (n<=0) then writeln('nedefinovane')
    else writeln(a:10:5,'^',n,'= ',vysledok:10:5);
readln;
end.

```

15. a) Máme jednorozmerné pole s n znakmi. Napíšte program, ktorý z tohto poľa vyhodí všetky nuly.

```

program mol5A;
const n=10;
var pole_nula,pole:string[n];{string je vlastne obycajne pole znakov
    vlastne platy > sting[n]=array[1..n] of char}
    I:integer;
begin
pole:='';
writeln('zadajte ',n,' znakov');
readln(pole_nula);
for I:= 1 to n do begin {length(pole_nula) toto je dlzka pola string}
    if pole_nula[I]<>'0' then pole:=pole+pole_nula[I];
    end;
writeln('vysledne pole znakov bez nul je');
writeln(pole);
readln
end.

```

b) ... na výpočet lineárnej rovnice $ax+b=0$.

```

program mol5B;
var a,b:real;
begin
writeln('zadajte parametre rovnice a*x+b=0, teda a & b');
readln(a,b);
if (a=0)and(b=0) then writeln('riesenie je R')
    else if a=0 then writeln('rovnica nema risenie')
        else writeln('koren x= ',(-1*b/a):10:10);
readln;
end.

```

16. a) Napíšte program, ktorý po zadaní a, b, c vypočíta koreň kvadratickej rovnice $ax^2+bx+c=0$.

```

program mol6A;
var a,b,c,x1,x2,D:real;
begin
writeln('zadajte parametre kvadratickej rovnice a*x^2 + b*x + c=0');
writeln(' v poradi a,b & c');

```

```

readln(a,b,c);
{vyzera to trochu strasidelne tolko podmienok, ale treba sa na to dobre pozriet}
{zistime si diskriminant}
D:=sqr(b)-4*a*c;
if D<0 then writeln('rovnicma nema riesenie v obore realnych cisel ale komplexnych, to si vypocitajte sami ')
      else if a=0 then writeln('toto nieje kvadraticka rovnica') {tu sa da doplnit algoritmus pre riesenie linearnej
rovnicma}
      else begin
x1:=(-1*b+sqr(D))/(2*a);
if D=0 then writeln('rovnicma ma dvojnasoenny koren x=',x1:10:10)
      else begin
x2:=(-1*b-sqr(D))/(2*a);
writeln('korene: x1= ',x1:10:10,' x2= ',x2:10:10);
end;
end;

readln;
end.

```

b) Nakreslite vývojový diagram na výpočet absolútnej hodnoty.
- ten je uvedený na konci – s. 16

17. a) Napíšte program, ktorý zo vstupu a, n vráti a^n .
viz [MO] 14b

b) Nakreslite vývojový diagram na výpočet kvadratickej rovnice $ax^2 + bx + c = 0$.
viz [MO] 16a, tam sa treba pohrať hlavne s tým diskriminantom

18. a) Napíšte program, ktorý po zadaní a, b vypočíta koreň lineárnej rovnice $ax + b = 0$.
viz [MO] 15b

b) ... na výmenu dvoch premenných.

```

...      {pozor, premenne a, b, x su rovnakeho typu}
x:=a;
a:=b;
b:=x;{x je akoze a}
...

```

19. a) Napíšte funkciu, ktorá vráti z jednorozmerného poľa minimum.

```

program mo19A;
const n=10;{dlzka toho pola}
type pole=array[1..n] of real;
var cisla:pole;
    I:integer;

function min(cisla:pole):real;
  var I:integer;
      d:real;
          kontrolka:boolean;
  begin
    kontrolka:=false;
    for I:=1 to n do begin
      if kontrolka=false then d:=cisla[I]{prvy krat v cykle}
        else begin
          if cisla[I]<d then d:=cisla[I];
        end;
      kontrolka:=true;
    end;

    min:=d;
  end;
begin
writeln('zadajte ',n,' cisel');
for I:=1 to n do begin
  readln(cisla[I]);      end;
writeln('minimum z tohto pola je ',min(cisla):10:10);

```

```
readln;
end.
```

b) ... na výpočet obsahu a obvodu kružnice.

```
program mo19B;
var r,povrch,obvod:real;
begin
writeln('Zadajte polomer r kruhu');
readln(r);
povrch:=PI*sqr(r);
obvod:=2*PI*r;
writeln('Povrch kruhu ',povrch:10:10,' j2');
writeln('Obvod kruhu je ',obvod:10:10,' j');
readln;
end.
```

20. a) Napíšte program, ktorý vynásobí dve matice a napíše výslednú maticu.

- Najskôr musíme vedieť ako násobia matice:

Násobiť môžeme také dve matice, pre ktoré platí, že jedna je rovnako široká ako je druhá vysoká. Presnejšie : Nech $A=(a_{ij})$ je matica typu $m \times n$ a nech $B=(b_{ij})$ je matica typu $n \times p$. (n je počet riadkov) Potom matica $C=(c_{ij})$ typu $m \times p$, pre prvky ktorej platí

$$c_{ij}=a_{i1} \cdot b_{1j} + a_{i2} \cdot b_{2j} + \dots + a_{in} \cdot b_{nj}$$

kde $i = 1, 2, \dots, m$, $j = 1, 1, \dots, p$ sa nazýva súčinom matíc A a B . Zapisujeme to $C=A \cdot B$.

```
program mo20A;
const m=2;
      n=3;
      p=4;
type matical=array[1..m,1..n] of real;
     matica2=array[1..n,1..p] of real;
     matica= array[1..m,1..p] of real;
var A:matical;
    B:matica2;
    C:matica;
    i,j,K:integer;{indexy}

begin
writeln('zadajte prvu maticu typu ',m,'*',n);
for i:=1 to m do
  for j:=1 to n do begin
    writeln('A[' ,i ,',',j ,']= ');
    readln(A[i,j]);
  end;

writeln('zadajte druhu maticu typu ',n,'*',p);
for i:=1 to n do
  for j:=1 to p do begin
    writeln('B[' ,i ,',',j ,']= ');
    readln(B[i,j]);
  end;

{vycistime maticu C}
for i:=1 to m do
  for j:=1 to p do C[i,j]:=0;

{vynasobime matice}
for i:=1 to m do
  for j:=1 to p do begin
    for K:=1 to n do begin
      C[i,j]:=C[i,j]+A[i,K]*B[K,j]
    end;
  end;

{vypiseme vysledok}
```

```
writeln('matica C=A*B je:');
for i:=1 to m do begin
  for j:=1 to p do begin
    write(C[i,j]:15:5);
    end;
    writeln;
  end;
end;
readln;
end.
```

b) Nakreslite vývojový diagram, ktorý zistí, či tri čísla môžu byť stranami trojuholníka.
viz [MO] 1b – táto úloha je častou programom, ale navyše zistí či sú to strany pravouhlého trojuholníka.

21. a) Napíšte program, ktorý bude simulovať 10 hodov kocky.

```
program mo21A;
const pocet=10; {pocet hodov kocky}
var hod:byte; {ten ma rozmedzie 0 az 255}
    I,M:integer;
begin
randomize; {toto je tu potrebne pre generovanie nahodnych cisel}
writeln('Tu su simulovane hody kockou');
writeln('hod      |hodnota | 123456');
writeln('-----'); {tieto ciarky su len na skraslenie}
for I:=1 to pocet do begin
  write(I:2, '.....[ ');
  hod:=trunc(random*6)+1; {trunc odreze desatinu cast cisla
    random je nahodne cislo z intervalu <0 , 1)}
  write(hod:1, ' ]      ');
  for M:=1 to hod do write(chr(2));
    {7 znak v ASCII tabulke je pekny}
  writeln; {skocime do noveho riadka}
  end;
end;
readln;
end.
```

b) Nakreslite vývojový diagram na usporiadanie troch čísel od najväčšieho po najmenšie.
viz [MO] 14a – otočené zadanie

22. a) Napíšte program, ktorý vypočíta absolútnu hodnotu z čísla n.

```
program mo22A;
var vstup:real;
begin
writeln('zadajte cislo');
readln(vstup);
if vstup<0 then vstup:=-1*vstup;
writeln('absolutna hodnota cisla je ',vstup:10:10);
{Da sa to aj jednoduksie
  writeln('absolutna hodnota cisla je ',abs(vstup):10:10);
  ale to nieje cielom tejto ulohy}
readln;
end.
```

b) ... či číslo na vstupe je deliteľné číslom 4.

```
program mo22B;
var vstup:longint;
begin
writeln('zadajte cislo - cele');
readln(vstup);
if vstup=0 then writeln('neviem')
  else if (vstup mod 4)=0 then writeln('cislo je delitelne cislom 4')
    {mohli by sme to napisat aj takto
      if ( trunc(vstup/4) - vstup/4 )=0 then ...}
    else writeln('cislo nieje delitelne cislom 4');
end;
readln;
end.
```

23. a) Napíšte program, ktorý po zadaní a, b vymení ich hodnoty.

```

...      {pozor, premenne a,b,x su rovnakeho typu}
        x:=a;
        a:=b;
        b:=x;{x je akoze a}
...

```

b) Zostav program na výpočet súčtu členov postupnosti čísel. Počet členov postupnosti nie je dopredu známy, ale až keď zadáme číslo 0, tak sa vypíše súčet členov.

```

program mo23B;
var vstup,sucet:real;
begin
writeln('zadajte cisla');
sucet:=0;
    repeat
        readln(vstup);
        sucet:=sucet+vstup;
    until vstup=0;
writeln('sucet clenov postupnosti je ',sucet:15:10);
readln;
end.

```

24. a) Napíšte program, ktorý vypočíta obvod a obsah trojuholníka.

```

program mo24A;
var a,b,c,povrch,obvod,s:real;
begin
writeln('zadajte strany trojuholnika a, b & c');
readln(a,b,c);
s:=(a+b+c)/2;
povrch:=sqrt(s*(s-a)*(s-b)*(s-c));{Heronov vzorec}
obvod:=a+b+c;
writeln('Povrch trojuholnika ',povrch:10:10,' j2');
writeln('Obvod trojuholnika je ',obvod:10:10,' j');
readln;
end.

```

b) Napíšte program pre riešenie sústavy dvoch lineárnych rovníc o dvoch neznámych vo všeobecnom tvare.

```

program mo24B;
var a,b,c,d,e,f:real;
    x,y:real;
    riesenie:boolean;
begin
riesenie:=true;
writeln('zadajte parametre pre prvu rovnicu a*x + by = c , teda a, b & c');
readln(a,b,c);
writeln('zadajte parametre pre druhu rovnicu d*x + ey = f, teda d, e & f');
readln(d,e,f);

x:=(f*b - c*e);
if (d*b)=(a*e) then riesenie:=false
    else begin
        x:=x/(d*b - a*e);
        if b<>0 then y:=(c-a*x)/b
            else if e<>0 then y:=(f-d*x)/e;
        end;

if riesenie=false then writeln('sustava nema riesenie')
    else writeln('x= ',x:10:10,' y=',y:10:10);
readln;
end.

```

25. a) Napíšte program, ktorý z n čísel nájde maximum. Ak ich je viac, nech vypíše, koľko týchto maxím je.

```

program mo25A;
var n,I,pocet:integer;
    max,cislo:real;
    kontrolka:boolean;
begin
writel('zadaj pocet cisel n');
readln(n);
if n<1 then halt;
writel('zadaj cisla');
kontrolka:=false;
pocet:=0;
for I:=1 to n do begin
    if kontrolka=false then begin {prvy krat je v cykle}
        readln(max);
        pocet:=1;
        end;
    if kontrolka<>false then begin
        readln(cislo);
        if cislo=max then inc(pocet,1);
            {zvacsí pocet o 1}
        if cislo>max then begin
            max:=cislo;
            pocet:=1;
            end;
        end;
        kontrolka:=true;
    end;
writel('Maximum z ',n:2,' cisel je ',max:5:5);
writel('tychto maxim je ',pocet);
readln;
end.

```

b) Zostavte program pre výpis ASCII znakov od čísla 32 do 255.

```

program mo25B;
var I:byte;
begin
writel('ASCII tabulka');
for I:=1 to 80 do write('-');
for I:=32 to 255 do begin
    write(I:3,'=');
    write(chr(I),' | ');
    {chr(cislo) vypise znak s cislom v tabulke ASCII}
    end;
readln;
end.

```

Základy programovania v Pascale pomocou príkladov 2005

- 1. Napíšte program, ktorý bude simulovať pád guľičky a jej odrazy. Pritom budú platiť tieto pravidlá: Guľička sa po dopade odráža do výšky 70% z predchádzajúcej výšky až dovtedy, kým jej výška nie je menšia ako 1, zároveň treba počítať dráhu.**

```

program mo2004_1;
const min_h=1;{konstanta, jej typ netreba zadat}

var h: real;
    actual_h:real;{teraajsia vyska}
    s:real;{draha}
    n:integer;{pocitadlo odrazov}

begin
writelnl('Zadajte vysku h, z ktorej gulicka pada:');
repeat
    write('h:= ');
    readln(h);
    if h<0 then writelnl('chybny vstup');
until h>=0;

s:=0;
n:=0;
{vyradenie padu z vysky < min_h }
if h<min_h then begin
    writelnl('guluocka sa neodrazi, prejdena draha je s= ',h:5:10);
    readln;{pocka na |Enter|}
    halt;{ukoncenie programu, taky rychli prikaz}
end;

s:=s+h;{prve padanie}
actual_h:=h;
repeat
    actual_h:= 0.7 * actual_h;
    if actual_h>min_h then begin
        inc(n);{increas n}
        write(n:3,'-ty odraz');
        s:= s + 2*actual_h;
        write(', vyska vystupu h= ',actual_h:2:2);
        writelnl(', draha s= ',s:2:3);
    end;

until actual_h < min_h;

writelnl('SUMARIZACIA:');
writelnl('- Pocet odrazov: ',n:3);
writelnl('- Prejdena draha: ',s:3:3);
writelnl('Stlacte Enter');
readln;
end.

```

- 2. Naprogramujte program pre prácu so súborom, ktorý vypíše na obrazovku, vymení veľké písmená za veľké a naopak, zistí počet riadkov a čísel v súbore a zároveň takto zmenený súbor uloží inde.**

```

program mo2005_3;

var fr: text;{typ zastupujuci subor v tvare textoveho suboru}
    fw: text;{fr - file read, fw - file write}
    ch:char;{pomocna premenna}
    poc_cisel, poc_riadkov:integer;

{funkcia na zmenu 'case' teda 'velkosti' pismena}
function change_case(ch:char):char;{vystupna hodnota je znak}
var out:char;{vystup}
begin
    out:=upcase(ch);{a -> A}
    {A -> a;}
    if (65<=ord(ch)) and (ord(ch)<=90) then out:=chr(ord(ch)+32);
    change_case:=out;
end;

```

```

begin
{ak niektorým príkazom celkom nerozumiete použité
pascal help > Shift+F1}

{tento blok sa dá vynechať, ak máme súbor pracal.txt}
{vytvoríme si pokusný súbor}
{asociujeme súbor pod názov premennej}
assign(fw,'c:/bp/bin/pracal.txt');
{vytvorenie súboru pracal.txt a jeho otvorenie
pre zápis. pravdaze do prázdneho súboru}
rewrite(fw);
{zápis do súboru sa deje podobne ako výpis na obrazovku
s tým rozdielom, že špecifikujem cieľ, kde píšeme}
writeln(fw,'Ahojte 146, IiIiI,');
write(fw,'nový riadok súboru Pracal.txt');
{zatvorenie súboru - toto zatvorenie nesmie chybať!}
close(fw);

{-----}
{nacítá sa súbor c:/bp/bin/pracal.txt}
assign(fr,'c:/bp/bin/pracal.txt');
{otvorenie súboru len pre čítanie}
reset(fr);
{vytvoríme súbor, do ktorého zapíšeme výsledok}
assign(fw,'c:/bp/bin/praca2.txt');
rewrite(fw);

poc_cisel:=0;
poc_riadkov:=1;{súbor s n riadkami má n-1 riadkových koncov (eoln)}

{výpis, konverzia a zápis do prace2.txt}
writeln('> Súbor pracal.txt:');
repeat
  if eoln(fr) then inc(poc_riadkov);{eoln je podobne ako eof(fr)}
  read(fr,ch);
  write(ch);
  if (47 < ord(ch)) and (ord(ch) < 58) then inc(poc_cisel);
  write(fw,change_case(ch));{zápis do prace2.txt}

until eof(fr);{ak je znak, ktorý sa má prečítať koniec súboru
tak funkcia eof vráti true, a teda nám povie, že
sme prečítali všetko.}

close(fw);
close(fr);

{na koniec vypíšeme zmenený súbor a štatistiku}
writeln;
writeln('> Súbor prace2.txt:');
assign(fr,'c:/bp/bin/praca2.txt');
reset(fr);
repeat
  read(fr,ch);
  write(ch)
until eof(fr);
close(fr);

{štatistika}
writeln('');
writeln('-----');
writeln('pocet riadkov súboru: ',poc_riadkov:6);
writeln('pocet čísiel v súbore: ',poc_cisel:5);
readln;
end.

```

3. Vytvorte hru na hádanie čísla, kde si počítač pomyslí číslo z intervalu a Vy potom budete pomocou náповedy (väčšie/menšie) hádať.

```

program mo2005_5;
{háďanie čísla, počítač si myslí číslo}
var dh,hh: integer;{hranice intervalu háďania}
    cislo: longint;
    koniec,koniec_programu:boolean;{indikuje koniec hry, programu}

```

```

volba:string[10];{[10] - dlzka retazca}
tip,h:longint;
pokusy:longint;

begin
koniec_programu:=false;
repeat{blok hry}

writeln('> Zadajte hranice intervalu, z ktoreho budete typovat cislo:');
write('Dolna hranica:');
readln(dh);
write('Horna hranica:');
readln(hh);
if hh<dh then begin h:=hh; hh:=dh; dh:=hh; end;

{generator nahodnych cisel}
randomize;
{cislo je z intervalu <dh, hh>}
cislo:=trunc(dh + random(hh-dh+1));
koniec:=false;
pokusy:=0;
repeat{blok tipovania}
  write('tip: ');
  readln(tip);
  inc(pokusy);
  if tip<cislo then writeln('> vacsie')
  else if tip>cislo then writeln('> mensie')
  else begin
    writeln('> !!Gratulujem!! cislo je: ',cislo:4);
    writeln(' STATISTIKA');
    writeln(' pokusov: ',pokusy:10);
    koniec:= true;
  end;
until koniec;

write('prajete si pokracovat? ["ano"/"nie"]: ');
readln(volba);
if volba<>'ano' then koniec_programu:=true;

until koniec_programu;

end.
```

4. Urobte interaktívny program pre počítanie objemu a povrchu telies: kocka, kváder, valec, guľa. Nezabudnúť aj na kontrolu vstupných údajov.

```

program mo2005_7;
var vyber:byte;{cislo vyberu}
    objem:boolean;
    input,input2,input3:real;{vstupny udaj}

{testujem korektnost cisla}
function cislo_ok(input: real):boolean;
begin
  if input>=0 then cislo_ok:=true
  else cislo_ok:=false;
end;

{ak objem bude false tak pocitame povrch
inak objem}

function kocka(co: boolean; a: real):real;
begin
  if co then kocka:=a*a*a
  else kocka:=a*a*6;
end;

function kvader(co: boolean; a,b,c: real):real;
begin
  if co then kvader:=a*b*c
  else kvader:=a*b*2 + a*c*2 + c*b*2;
end;
```

```

function valec(co: boolean; r,h: real):real;
begin
  if co then valec:=PI*r*r*h
    else valec:=PI*r*r*2 + PI*r*2*h;
end;

function gula(co: boolean; r: real):real;
begin
  if co then gula:=(4/3)*PI*r*r*r
    else gula:=4*PI*r*r;
end;

begin
repeat

write('> Vyber povrch[1]/objem[2]: ');
  readln(vyber);
  if vyber=2 then objem:=true
    else objem:=false;
write('> Vyber kocka[1]/kvader[2]/valec[3]/gula[4]: ');
  readln(vyber);

case vyber of
  1{kocka}:begin writeln('> KOCKA');
    repeat
      write('a:= ');readln(input);
      until cislo_ok(input);

      if objem then write('> Objem ')
        else write('> Povrch ');
      writeln('kocky je ',kocka(objem,input):5:5);
    end;
  2{kvader}:begin writeln('> KVADER');
    repeat
      write('a:= ');readln(input);
      until cislo_ok(input);

      repeat
        write('b:= ');readln(input2);
        until cislo_ok(input2);

      repeat
        write('c:= ');readln(input3);
        until cislo_ok(input3);

      if objem then write('> Objem ')
        else write('> Povrch ');
      writeln('kvadra je ',kvader(objem,input,input2,input3):5:5);
    end;
  3{valec}:begin writeln('> VALEC');
    repeat
      write('r:= ');readln(input);
      until cislo_ok(input);

      repeat
        write('h:= ');readln(input2);
        until cislo_ok(input2);

      if objem then write('> Objem ')
        else write('> Povrch ');
      writeln('valca je ',valec(objem,input,input2):5:5);
    end;
  4{gula}:begin writeln('> GULA');
    repeat
      write('r:= ');readln(input);
      until cislo_ok(input);

      if objem then write('> Objem ')
        else write('> Povrch ');
      writeln('gule je ',gula(objem,input):5:5);
    end;
end;

```

```

end;{konci case}

write('> Opakovat? ano[1]/nie[2]: ');
readln(vyber);
if vyber<>1 then halt;{okamzite skoncenie}

until false;
end.

```

5. Pravdepodobne ste už natrafili na tzv. Cézarovu šifru, ktorá je založená na tom, že posúva znaky o vopred vybraný počet miest (A -> C, B -> D, Z -> B apod.). Urobte program na šifrovanie súboru, pričom zvlášť sa budú kódovať veľké, malé písmená a číslice.

```

program mo2005_11;
{sifrovanie textu}
var fr: text;{typ zastupujuci subor v tvare textoveho suboru}
    fw: text;{fr - file read, fw - file write}
    ch:char;{pomocna premenna}
    kluc:shortint;{sifrovaci kluc, na desifrovanie treba pouzit zaporny}
    zdroj,ciel:string[200];{mena suborov}

{funkcia na posun cisla o velkost kluca
 s prihliadnutim na cykliskost}
function posun(cislo,dh,hh:integer):byte;
    var out_int:integer;
        znamienko:shortint;
    begin
        out_int:=cislo+kluc;
        if out_int>hh then znamienko:=-1
            else if out_int<dh then znamienko:=+1
                else znamienko:=0;

        repeat
            out_int:= out_int + znamienko * (hh-dh+1);
        until (out_int<=hh) and (out_int>=dh);

        posun:=out_int;
    end;

{funkcia na sifrovanie pismena/cisllice}
function sifra(ch:char):char;{vystupna hodnota je znak}
    var out:char;{vystup}
    begin
        out:=ch;

        {A-Z}
        if (65<=ord(ch)) and (ord(ch)<=90) then out:=chr(posun(ord(ch),65,90));
        {a-z}
        if (97<=ord(ch)) and (ord(ch)<=122) then out:=chr(posun(ord(ch),97,122));
        {0-9}
        if (48<=ord(ch)) and (ord(ch)<=57) then out:=chr(posun(ord(ch),48,57));

        sifra:=out;
        {skuste aj
        sifra:=chr(posun(ord(ch),0,255))}
    end;

begin
writelnl('> Program Sifra');
{nacita sa subor c:/bp/bin/sifra1.txt}
write('> meno suboru na sifrovanie: c:/bp/bin/');
readln(zdroj);
zdroj:='c:/bp/bin/'+zdroj;
assign(fr,zdroj);
{otvorenie suboru len pre citanie}
reset(fr);

{vytvorime subor, do ktoreho zapiseme vysledok}
write('> meno suboru na ulozenie: c:/bp/bin/');
readln(ciel);
ciel:='c:/bp/bin/'+ciel;

```

```

assign(fw,ciel);
rewrite(fw);

{sifrovaci kluc}
write('> sifrovaci kluc [<-127 - +127>]: ');
readln(kluc);

{vypis, konverzia a zapis do ciel}
writeln('> Subor: ',zdroj);
repeat
    read(fr,ch);
    write(ch);
    write(fw,sifra(ch));
until eof(fr);{ak je znak ktory sa ma precitat koniec suboru
                tak funkcia eof vrati true, a teda nam povie, ze
                sme precitali vsetko.}

close(fw);
close(fr);

{na koniec vypiseme zmeneny subor}
writeln('');
writeln('> Subor: ',ciel);
assign(fr,ciel);
reset(fr);
repeat
    read(fr,ch);
    write(ch)
until eof(fr);
close(fr);
writeln;
readln;
end.

```

6. Urobte program, ktorý vygeneruje čísla ako v lotérií zo zadaného intervalu a potom bude hráč hádať vopred daný počet čísel, podobne ako v tipovacej súťaži.

```

program mo2005_13;
{tip}
const top_n=100;{max. pocet tipov}
type pole=array[1..top_n] of integer;
    pole_boolean=array[1..top_n] of boolean;

var dh,hh:integer;
    pocitac,clovek,vysledok:pole;{zasobniky tipov}

{pole v ktorom je napisane, ci uz cislo
z tipov pocitaca bolo uhadnute, je to preto
aby pri viacnashobnom tipe hraca nedoslo k
mnozeniu "spravnych" odpovedi}
pocitac_out:pole_boolean;
input:integer;
zhoda:integer;{pocet spravnych tipov}
n:byte;{pocet tipovanych cisel}
I,M:integer;

begin
repeat
writeln('> Zadajte interval tipovania');
write('dh: ');
readln(dh);
write('hh: ');
readln(hh);
write('> Zadajte pocet tipovanych cisel [<1,',top_n,'>]: ');
readln(n);
{kontrola}if (n<1) or (n>top_n) then n:=1;

randomize;{generator nahodnych cisel sa zapol}

{naplni sa zasobnik tipov pocitaca: pocitac[]}
for I:=1 to n do begin
    pocitac[I]:=trunc(dh+random(hh-dh+1));
end;

```

```

{naplni sa zasobnik tipov hraca: clovek[]}
for I:=1 to n do begin
    write('> tip['I, '/' ,n, ']: ');
    read(input);
    clovek[I]:=input;
end;
{porovnanie typov a vyhodnotenie}
zhoda:=0;
for I:=1 to top_n do vysledok[I]:=0;{pole vysledok ma len ulohu oznacit
    uhadnute cisla}
for I:=1 to top_n do pocitac_out[I]:=false;
for I:=1 to n do begin
    M:=1;
    repeat
        if (clovek[I]=pocitac[M]) and (not pocitac_out[M]) then begin
            inc(zhoda);
            vysledok[I]:=1;
            pocitac_out[M]:=true;
        end;
        inc(M);
    until (M>n) or (vysledok[I]=1)
    end;
writeln('> VYHODNOTENIE');
writeln('- Cisla/tipy hraca/zhody');
for I:=1 to n do write(pocitac[I]:4);writeln;
for I:=1 to n do write(clovek[I]:4);writeln;
for I:=1 to n do write(vysledok[I]:4);writeln;
writeln('> Pocet spravnych tipov: ',zhoda:5);

    write('> Opakovat? ano[1]/nie[2]: ');
    readln(input);
    if input<>1 then halt;{okamzite skoncenie}

until false;
end.

```

7. Urobte program, v ktorom budú dvaja hráči hádať súčet hodnôt na hodených dvoch kockách až kým neuhádnu. Pravdaže ten kto bude mať menej tipov vyhráva.

```

program mo2005_14;
{hody kockou}
var cislo1,cislo2:byte;
    hracl,hrac2:byte;
    koniec:boolean;
    tip:byte;
begin
randomize;

cislo1:=trunc(1+random*6);
cislo2:=trunc(1+random*6);

writeln('> Hada hrac 1.:');
koniec:=false;
hracl:=0;
repeat
    write('> tip: ');
    readln(tip);
    if tip<>(cislo1 + cislo2) then writeln('- nie')
        else begin
            writeln('- ano! ',cislo1,'+',cislo2);
            koniec:=true;
        end;
    inc(hracl);
until koniec;

cislo1:=trunc(1+random*6);
cislo2:=trunc(1+random*6);

writeln('> Hada hrac 2.(pozor, uz ine dva hody):');
koniec:=false;
hrac2:=0;
repeat

```

```

write('> tip: ');
readln(tip);
if tip<>(cislo1 + cislo2) then writeln('- nie')
  else begin
    writeln('- ano! ',cislo1,'+',cislo2);
    koniec:=true;
  end;
inc(hrac2);
until koniec;

writeln('> STATISTIKA');
writeln('hrac1: ',hrac1:1,' hrac2: ',hrac2:1);
if hrac1<hrac2 then writeln ('- vyhral hrac 1.')
  else if hrac1>hrac2 then writeln ('- vyhral hrac 2.')
    else writeln('- spravodлива remiza');
readln;
end.

```

8. Skúste urobiť simuláciu hry "Kameň, papier, nožnice" medzi človekom a počítačom.

```

program mo2005_15;
{kamen papier noznice
 1      2      3      }
const znak: array[1..3] of string[7] = ('kamen','papier','noznice');
var pocitac,clovek:byte;
    clovek_vitaz:integer;{indikuje vytaztvo cloveka, [0/5/10]}
    input:byte;

begin
randomize;
repeat
  repeat
    write('> Zadajte kamen[1]/papier[2]/noznice[3]: ');
    readln(clovek);
  until (clovek>=1) and (clovek<=3);
pocitac:=trunc(1+random*3);

writeln('> Hrac: ',znak[clovek],' Pocitac: ',znak[pocitac]);
clovek_vitaz:=0;
case clovek of
  1:begin{kamen}
    case pocitac of
      1:clovek_vitaz:=5;
      2:{-};
      3:clovek_vitaz:=10;
    end;
  end;
  2:begin{papier}
    case pocitac of
      1:clovek_vitaz:=10;
      2:clovek_vitaz:=5;
      3:{-};
    end;
  end;
  3:begin{noznice}
    case pocitac of
      1:{-};
      2:clovek_vitaz:=10;
      3:clovek_vitaz:=5;
    end;
  end;
end;

case clovek_vitaz of
  0 :writeln('- Vyhral pocitac');
  5 :writeln('- Remiza');
 10:writeln('- Vyhral hrac');
end;

write('> Opakovat? ano[1]/nie[2]: ');
readln(input);
if input<>1 then halt;{okamzite skoncenie}

```

```
until false;
end.
```

- 9. Naprogramujte hru oko. Hrá sa so žolíkovými kartami bez dolníkov, horníkov a kráľov, pričom každá karta má hodnotu čísla na karte a eso má hodnotu 1. Cieľom hry je ťahať karty, až kým nebude súčet kariet 21. To je oko. Hráč ťahá karty po jednej a hocikedy môže prestať. Ak prekročí súčet 21, automaticky prehráva. Vyhráva ten hráč, ktorý je najbližšie k súčtu 21, alebo ktorý má oko.**

```
program mo2005_16;
{"oko"}
const oko=21;
      top_n=100;
type pole_hracov=array[1..top_n] of byte;
      pole_hracov_vitaz=array[1..top_n] of boolean;
var hrac:pole_hracov;
      hrac_vitaz:pole_hracov_vitaz;{indikuje prehru hraca ak prekroci 21}
      vitaz:pole_hracov_vitaz;{cisla vitazov}
      n:byte;{pocet hracov}
      I,M:byte;
      koniec:boolean;
      volba:string[3];
      karta:byte;

begin
randomize;
repeat
      write('> Zadajte pocet hracov [<1,',top_n,'>]: ');
      readln(n);
until (n>=1) and (n<=top_n);
{"vycistenie" ptremennych}
for I:=1 to top_n do hrac_vitaz[I]:=true;
for I:=1 to top_n do hrac[I]:=0;
for I:=1 to top_n do vitaz[I]:=0;

for I:=1 to n do begin
      koniec:=false;
      writeln('> Ide hrac c.',I:2);
      repeat
            write(' > Kartu? ["ano"/"nie"]: ');
            readln(volba);
            if volba='nie' then koniec:=true
            else begin
                  karta:=trunc(1+random*10);
                  inc(hrac[I],karta);{zvacsí o hodnotu}
                  writeln(' - Vybrata karta ma hodnotu: ',karta:2);
                  if hrac[I]>oko then hrac_vitaz[I]:=false;
                  end;
            until koniec or (hrac[I]>oko);
            writeln(' - Hrac c.',I,' ma ',hrac[I],' bodov')
            end;
      {Vyhodnotenie}
      M:=1;
      for I:=1 to n do begin
            if hrac_vitaz[I] then
                  if vitaz[M]=0 then vitaz[M]:=I{prvy vstup}
                  else if abs(hrac[I]-oko) < abs(hrac[vitaz[M]]-oko) then vitaz[M]:=I
                  else if abs(hrac[I]-oko) = abs(hrac[vitaz[M]]-oko) then begin
                        {jedna sa o pripad zhody bodov}
                        inc(M);
                        vitaz[M]:=I;
                        end;
            end;
      end;
      {vypis}
      M:=1;
      write('> Vitazi su hraci s cislami: ');
      repeat
            if vitaz[M]=0 then if M=1 then writeln('Niesu vitazi')
                               else koniec:=true
                               else writeln(vitaz[M]:2,'.', ' ');
            inc(M);
```

```
until koniec or (M>n);
readln;
end.
```

10. Urobte program na vyhodnotenie písomky, kde sa zadá maximálny počet bodov, body jednotlivých žiakov, následne sa žiaci oznámujú (100%-90% 1, 89%-75% 2, 74%-60% 3, 59%-40% 4, 39%-0% 5) a vyhodnotí sa zastúpenie známok ako aj priemerná známka.

```
program mo2005_17;
const _1=90;{dölne hranice znamok}
      _2=75; _3=60; _4=40; _5=0;
      top_n=100;
type pole_ziakov=array[1..top_n,1..2] of real;
      {x[i,1] - pocet bodov, x[i,2] - znamka}
      pole_znamok=array[1..5] of byte;{pocety znamok}
var ziak:pole_ziakov;
    znamka:pole_znamok;
    n:byte;{pocet ziaikov}
    top_b:real;{maximalny pocet bodov}
    I,M:integer;
    k:real;{pomocna premenna}
begin
{vycistenie premennych}
for I:=1 to top_n do begin ziak[I,1]:=0; ziak[I,2]:=0; end;
for I:=1 to 5 do znamka[I]:=0;

repeat{zadanie poctu ziaikov s podmienkou}
  write('> Zadajte pocet ziaikov [<1,',top_n,'>]: ');
  readln(n);
until (n>=1) and (n<=top_n);

repeat{zadanie maximalneho poctu bodov s podmienkou}
  write('> Zadajte maximalny pocet bodov: ');
  readln(top_b);
until top_b>0;

{nacitanie poctu bodov ziakom}
for I:=1 to n do begin
  write(' > Ziak c.',I:2,', pocet bodov: ');
  readln(ziak[I,1]);
end;

{Vyhodnotenie}
for I:= 1 to n do begin
  k:= 100 * (ziak[I,1]/top_b);
  ziak[I,2]:=1;
  if k<_1 then ziak[I,2]:=2;
  if k<_2 then ziak[I,2]:=3;
  if k<_3 then ziak[I,2]:=4;
  if k<_4 then ziak[I,2]:=5;
end;
for M:=1 to 5 do for I:=1 to n do begin
  if ziak[I,2]=M then inc(znamka[M]);
end;
{Statistika}
writeln('> STATISTIKA');
for M:=1 to 5 do writeln(' - znamka ',M,'; pocet: ',znamka[M]:2,
  ' ; zastupenie: ',((znamka[M]/n)*100):2:2,'%');

k:=0;
for I:=1 to n do k:=k+ziak[I,2];
writeln(' - Priemerna znamka je: ',(k/n):1:3);
readln;
end.
```

11. Na vstupe programu je veta ukončená bodkou, program určí počet znakov vo vete (aj s medzerami), nahradí veľké malými písmenami, pred každý znak vloží "*", vo vete vymaže medzery a vetu vypíše aj odzadu.

```
program mo2005_20;
var veta:String[255];
    I:integer;
    ch:char;
    poc_znakov:integer;
```

```

{A -> a}
function lowercase(ch:char):char;
begin
  if (ch>='A') and (ch<='Z') then lowercase:=chr(ord(ch) + 32)
  else lowercase:=ch;
end;

begin
writeln('> Zadajte vetu ukoncenu bodkou:');

I:=1;
veta:='';
repeat
  read(ch);
  veta:=veta+ch;
  inc(I);
until (ch='.') or (I>255);
readln;
poc_znakov:=length(veta);{length vracia dlzku retazca}
writeln(' > Pocet znakov vo vete: ',poc_znakov);

{ A > a }
writeln(' > Veta so zmenou "A"->"a":');
for I:=1 to length(veta) do write(lowercase(veta[I]));
writeln;

{ *ch }
writeln(' > Veta so zmenou "A" -> "*A":');
for I:=1 to length(veta) do write('*',veta[I]);
writeln;

{ ' ' > '' }
writeln(' > Veta bez medzier:');
for I:=1 to length(veta) do if veta[I]<>' ' then write(veta[I]);
writeln;

{ 'ab' > 'ba' }
writeln(' > Veta napisana odzadu:');
for I:=length(veta) downto 1 do write(veta[I]);
writeln;

readln;
end.

```

12. Urobte program na výpočet sklárneho súčinu n-rozmerných vektorov, zároveň aj so zistením uhla, ktorý zverajú. ($|u||v|\cos(\alpha) = u \cdot v = (u_x v_x + u_y v_y + \dots)$)

```

program mo2005_21;
{skalarny sucin}
const top_n=100;
  {vektorove pole}
type pole_v=array[1..top_n] of real;
var vektor1,vektor2:pole_v;
  n:byte;
  I:integer;
  skal_sucin,uhol:real;

{vrati absolutnu hodnotu vektora}
function abs_vektor(vektor: pole_v):real;
  var I:integer;
  out:real;
  begin
    out:=0;
    for I:=1 to n do out:=out + vektor[I] * vektor[I];
    abs_vektor:=sqrt(out);
  end;

begin
repeat
  write('> Zadajte rozmer vektorov: ');
  readln(n);
until (n>0) and (n<=top_n);

```

```

for I:=1 to n do begin
    write(' > vektor1[' ,I, ']:=');
    readln(vektor1[I]);
    end;
for I:=1 to n do begin
    write(' > vektor2[' ,I, ']:=');
    readln(vektor2[I]);
    end;
{vypis vektorov}
writeln('> Vektor1:');
write(' '); for I:=1 to n do write(vektor1[I]:2:3, ' '); writeln(' ');
writeln('> Vektor2:');
write(' '); for I:=1 to n do write(vektor2[I]:2:3, ' '); writeln(' ');

{vypocet skal_sucinu}
skal_sucin:=0;
for I:=1 to n do skal_sucin:= skal_sucin + vektor1[I] * vektor2[I];
writeln('> Skalarny sucin vektorov je: ',skal_sucin:5:5);

{vypocet uhla}
if skal_sucin=0 then uhol:=90

    else begin uhol:=( skal_sucin / (abs_vektor(vektor1)*abs_vektor(vektor2)) );
        uhol:= ArcTan (sqrt (1-sqr (uhol)) /uhol);{vid. Pascal Help}
        uhol:=(180/PI) * uhol;{premena na stupne}
        end;
writeln('> Uhol vektorov je: ',uhol:5:5, ' deg');
readln;
end.

```

13. Urobte jednoduchý program na načítanie mien žiakov a ich rodných čísel (prvých 8 čísel). Potom z týchto údajov počítač zistí počet dievčat a chlapcov (chlapci majú druhé dvojčísle z rodného čísla od 01 do 12, dievčatá od 51 do 62), vyhľadá žiakov, ktorí sa narodili v danom mesiaci a všetky údaje bude môcť exportovať do súboru.

```

program mo2005_22;
const top_n=255;
{Datovy typ record je ako sablona a je
predzvestou komplikovanejsich dat. typov.
Jeho pochopenie vsak nieje az take problematicke.}
type record_ziak = record
    meno:string[40];
    priezvisko:string[40];
    rc{rodne cislo}:string[8];
    mesiac:byte;{mesiac narodenia}
end;
pole_ziakov=array[1..top_n] of record_ziak;

var   ziak: pole_ziakov;
      n:byte;{pocet ziakov}
      I:integer;
      poc_dievcat:byte;
      mesiac,input:string[100];{pomocna premenna}
      chyba:integer;{pomocna premenna}
      input_int:integer;

{exportuje zoznam ziakov}
procedure exportuj(path: string);
    var fw: text;
        I:integer;
    begin
        assign(fw,path);
        rewrite(fw);
        writeln(fw, 'meno':16, 'priezvisko':16, 'rc':10, ' mesiac');
        for I:=1 to n do

writeln(fw,ziak[I].meno:16,ziak[I].priezvisko:16,ziak[I].rc:10,ziak[I].mesiac:3);

        close(fw);
    end;

```

```

begin
repeat
    write('> Zadajte pocet ziakov v triede: ');
    readln(n);
until (n>0) and (n<=top_n);

for I:=1 to n do begin
    writeln('> Ziak c.',I);
    write(' - Meno (krstne): '); readln(ziak[I].meno);
    write(' - Priezvisko: '); readln(ziak[I].priezvisko);
    write(' - Rodne cislo (prvych 8 cisel): '); readln(ziak[I].rc);
end;

{teraz pouzijem nasledovny nazov: ziak[I].rc[1]
 |prvok pola pole_ziakov
 |      |rc a index vymedzuje prvý znak textoveho retazca rc
-----}
ziak[I].rc[1]
|
|bodka oddeluje prvok (rc) zaznamu record_ziak
}
poc_dievcat:=0;
for I:=1 to n do begin
    mesiac:=ziak[I].rc[3];
    if (ziak[I].rc[3]='5') then begin inc(poc_dievcat); mesiac:='0'; end;
    if (ziak[I].rc[3]='6') then begin inc(poc_dievcat); mesiac:='1'; end;

    mesiac:= mesiac + ziak[I].rc[4];{12, 06 ..}
    {val ako value}
    val(mesiac, ziak[I].mesiac, chyba);
    {chyba suvisi s faktom ze konvertovat
    muozme dat aj textovy retazec}
    if chyba<>0 then ziak[I].mesiac:=0;{pri zlom rc bude mesiac 0}
end;

writeln('> Pocet dievcat je: ',poc_dievcat);
writeln('> Pocet chlapcov je: ',n-poc_dievcat);
{Vyhľadavanie}
repeat
    writeln;
    write('> Prajete si vyhľadat ziaka podla mesiaca ["ano"/"nie"]: ');
    readln(input);
    if input='ano' then begin
        write(' > Zadajte mesiac [1,2,..,12]: ');
        readln(input_int);
        writeln('      meno      priezvisko rc.      mesiac');
        for I:=1 to n do
            if ziak[I].mesiac=input_int then
                writeln(' - ',ziak[I].meno:12,' ',ziak[I].priezvisko:12,'
',ziak[I].rc:10,' ',ziak[I].mesiac:2);
        end;
    end;
until input<>'ano';
{export dat}
write('> Napiste nazov suboru pre export: C:/');
readln(input);
input:= 'C:/' + input;
exportuj(input);
end.

```

14. Urobte program na výpočet ciferného súčtu kladného celého čísla.

```

program mo2005_23;
var cislo:longint;
    cislo_string:string;
    vyber:string;{pomocna premenná}
    hodnota,cifra,chyba:integer;
    I:integer;
begin
repeat
repeat
    write('> Zadajte cislo: ');

```

```

    readln(cislo);
until cislo>0;
{konvertuje cislo na string cislo_string}
str(cislo,cislo_string);
{ciferny sucelt
length(string) vracia dlzku stringu}
hodnota:=0;
for I:=1 to length(cislo_string) do begin
    val(cislo_string[I], cifra, chyba);
    hodnota:=hodnota+cifra;
end;
writeln('> Ciferny sucet je: ',hodnota);

write('> Prajete si opakovat? ["ano"/"nie"]: ');
readln(vyber);
if vyber='nie' then halt;

{tento nekonecny ciklus je isteny moznostou opustit
program pomocou halt}
until false;
end.

```

15. Simulujte hru m hráčov, ktorí odoberajú z kopy zápalič po 1, 2, 3 zápalkách. Hráč, ktorý zoberie poslednú zápalku vyhráva. Nezabudnite na mená hráčov.

```

program mo2005_25;
const top_m=255;
type pole_hracov=array[1..top_m] of string[20];
var m:byte; {pocet hracov}
    n:integer; {pocet zapaliek}
    hrac:pole_hracov; {mena hracov}
    I:integer;
    input:byte;
    koniec:boolean; {koniec hry}
    vyber:string; {pomocna premenna}
begin
repeat
koniec:=false;
repeat
    write('> Zadajte pocet hracov: ');
    readln(m);
    write('> Zadajte pocet zapaliek: ');
    readln(n);
until ((m>0) and (m<top_m)) and (n>0);
{vlozenie mien hracov}
for I:=1 to m do begin
    write(' > Hrac c.',I,' ma meno: ');
    readln(hrac[I]);
end;

I:=0;
repeat
    inc(I);
    if I>m then I:=1; {cyklovanie}
    repeat
        write(' - Ide hrac ',hrac[I],', beries 1,2 alebo 3 zapalky?: ');
        readln(input);
    until (input=1) or (input=2) or (input=3);
    dec(n,input); {odobratie z kopy zapaliek}
    if n<1 then koniec:=true;
until koniec;

writeln('Vytazom sa stal: ',hrac[I]);

write('> Prajete si opakovat hru? ["ano"/"nie"]: ');
readln(vyber);
if vyber='nie' then halt;

until false;
end.

```

16. Napíšte program na nájdenie najväčšieho spoločného deliteľa dvoch čísel pomocou Euklidovho algoritmu odčítania.

```

program mo2005_28;
var a,b: longint;{cisla}
    nsd_a,nsd_b: longint;
    I:integer;
    vyber:string;
begin
repeat
repeat
    writeln('> Zadajte dve cele, kladne cisla:');
    readln(a,b);
until (a>0) and (b>0);
{pouzijem euklidov algoritmus}
nsd_a:=a;
nsd_b:=b;
I:=0;
writeln('    a    b');
repeat
    inc(I);
    writeln(I:2, ', ', nsd_a:4, '| ', nsd_b:4);
    if nsd_a>nsd_b then nsd_a:= nsd_a - nsd_b
        else if nsd_b>nsd_a then nsd_b:= nsd_b - nsd_a;
until nsd_a=nsd_b;
writeln('> Najvacsi spolocny delitel cisel ',a,' a ',b,' je: ',nsd_a);

write('> Zadat nove cislo? ["ano"/"nie"]: ');
readln(vyber);
if vyber<>'ano' then halt;
until false;
end.

```

17. Program bude dávať n príkladov na súčet dvoch celých kladných čísel menších ako 101. Žiak napíše odpoveď alebo "n" ako neviem, kedy mu nepribudnú body.

```

program mo2005_29;
var n:integer;{pocet prikladov}
    a,b:byte;{dve cisla}
    sucet:integer;{skutocny sucet}
    odpoved:string;
    vysledok:integer;{odpoved hraca}
    body,chyba:integer;
    I:integer;
begin
randomize;
repeat
body:=0;
repeat
    write('> Zadajte pocet prikladov: ');
    readln(n);
until n>0;
for I:= 1 to n do begin
    a:=random(101);
    b:=random(101);

    sucet:=a+b;
    write(' - ',I,' ', 'a, '+' ,b, ' = ');
    readln(odpoved);
    val(odpoved,vysledok,chyba);
    writeln('    spravna odpoved: ',sucet);
{ak je chyba=0 tak string odpoved je cislo! a nie napr."n"}
    if chyba=0 then if vysledok=sucet then inc(body)
        else dec(body);
    end;
writeln('> Pocet ziskanych bodov: ',body);
    write('> Znovu? ["ano"/"nie"]: ');
    readln(odpoved);
    if odpoved<>'ano' then halt;
until false;
end.

```

mo ≈ master outline (hlavný náčrt)

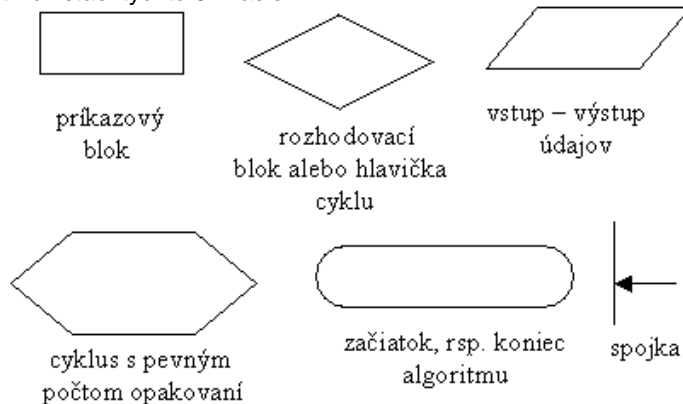
Na záver zhrniem pár rád:

- string je špeciálne jednorozmerné pole znakov
- v Borland Pascale fungujú tieto klávesové skratky (používať! zrychlí to prácu o 100%)
 - Ctrl + Insert – kopírovať
 - Ctrl + Delete – zmazať
 - Shift + Insert – vložiť
 - text sa označuje Shiftom
 - Ctrl + kurzorové šípky – rýchly pohyb v texte
 - F10 – menu hlavné
 - F3 – otvoriť súbor
 - Alt + F3 – zavrieť aktuálne okno
 - F6 – prepínanie medzi otvorenými oknami
 - F5 – maximalizovanie aktuálneho okna
 - Shift + F1 – slovník pojmov – stačí napísať pojem
 - Alt + X – zavrieť Borland Pascal

Zámerné som nepísal vývojové diagramy, tie sú v podstate takmer totožné so štruktúrou programu v Pascale ale vsadené do grafických diagramov. Tu je postup tvorby vývojových diagramov:

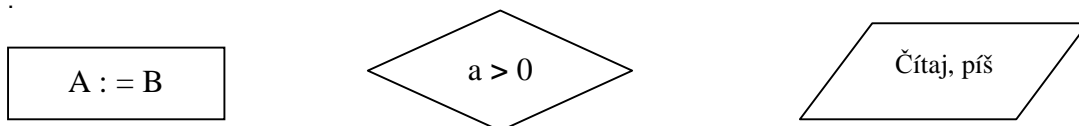
Zápis algoritmov :

1. Prírodný jazyk
2. Jazyk špecifikácie – zápis pomocou výrokových formúl a rôznych výrazov /aritmetické, logické, atď/, v prípade potreby doplnený poznámkami v prírodnom jazyku.
3. Vývojový diagram – najstaršia grafická forma zápisu, ktorá bola dokonca normalizovaná. Jedná sa o orientovaný graf, v ktorom jednotlivé uzly sú značky s presne určeným významom. Norma obsahuje viac ako 30 značiek, ale na zápis väčšiny algoritmov stačí týchto 6 značiek :



Celý algoritmus sa teda tvorí pomocou obdĺžnikov, kosoštvorcov a iných blokov. Do jednotlivých blokov zapisujeme príkazy a podmienky, jednotlivé bloky spájame čiarami a môžeme označiť šípkami a takto zvýrazniť tok riadenia.

Príklady :



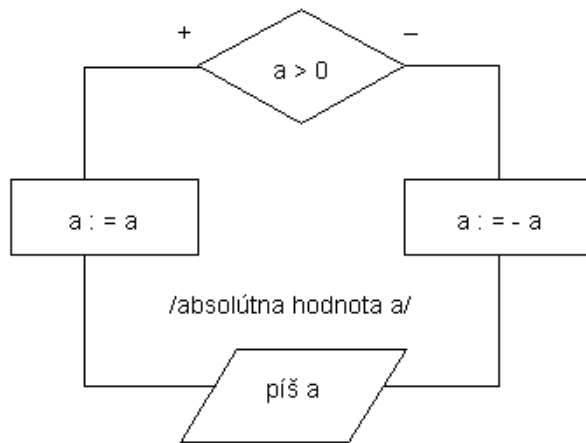
Pri kreslení vývojových diagramov by sme mali dodržať nasledovné pravidlá :

1. minimalizujeme kríženie hrán
2. kladné a záporné vetvy všetkých rozhodujúcich blokov v diagrame orientujeme rovnako /ak je kladná vetva raz na pravej, raz na ľavej strane, je možné, že sa pomýlime/
3. pri neúplnom vetvení nech prázdna vetva je záporná vetva
4. dlhé príkazy alebo zložité podmienky nepíšeme do diagramu, ale napíšeme ich osobitne, označíme a do diagramu napíšeme iba značku, odkaz.

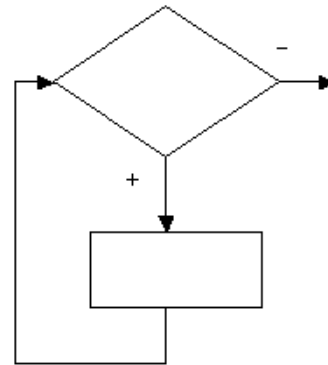
Príklady :

Postupnosť príkazov: ak máme viac príkazov, môžeme nahradiť skupinou príkazov :

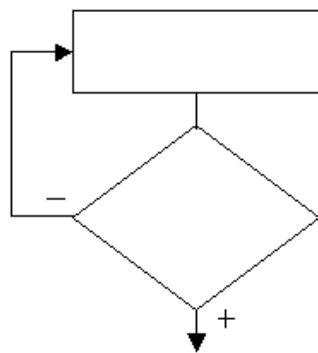
binárne vetvenie :



cyklus s podmienkou na začiatku



cyklus s podmienkou na konci



cyklus s pevným počtom opakovaní

