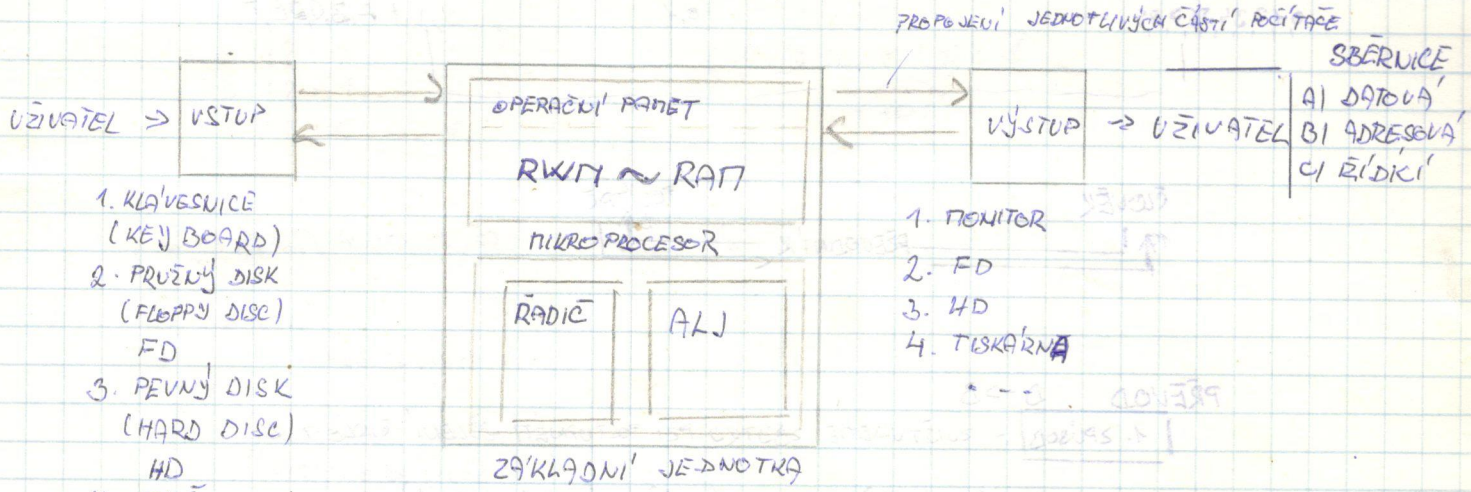
1.-2. 3.9.1992

SLOŽENÍ POČÍTAČE
[JOHN VON NEUMANN]



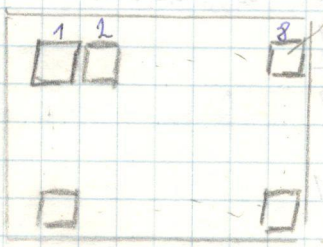

1. RWM
   - READ
   - WRITE
   - MEMORY

   - UMOŽŇUJE UKLÁDÁNÍ, ČTENÍ DAT, INFORMACÍ, PROGRAMŮ

2. ALJ
   - ARITMETICKÁ (ARITM. OPERACE - + - . /)
   - LOGICKÁ (LOGICKÉ OPERACE - NOT, OR, AND)
   - JEDNOTKA

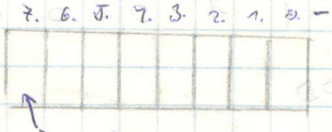
3. ŘADIČ - ŘADÍ INSTRUKCE DO POŘADÍ, VE KTERÉM JE POČ. PROVEDE

PŘEDSTAVA RAM




1BYT - NEJMENŠÍ ADRESOVATELNÁ JEDNOTKA*

PŘEDSTAVA 1BYTU




1BIT 0/1

1BIT - NEJMENŠÍ JEDNOTKA INFORMACE

KAPACITA RAM - POČET BYTŮ

1 BIT - 1b
1 BYT - 1B

JEDNOTKY KAPACITY RAM

1. BYT = 1B
1. kB = $2^{10}$B = 1024B ≐ $10^3$B
1 MB = $(2^{10})^2$B = $1024^2$B = ..... ≐ $10^6$B
1 GB = $(2^{10})^3$B = $1024^3$B = ..... ≐ $10^9$B

# ČÍSELNÉ SOUSTAVY – POČÍTAČ

1. DECIMÁLNÍ – DESÍTKOVÁ (10): 0 1 2 3 4 5 6 7 8 9
2. BINÁRNÍ, DVOJKOVÁ (2): 0, 1
3. HEXADECIMÁLNÍ, ŠESTNÁCTKOVÁ (16): + A, B, C, D, E, F

ČLOVĚK (19) ⟷ PŘEVODNÍK ⟷ POČÍTAČ (2)

PŘEVOD D → B

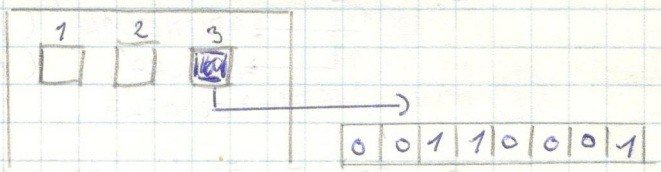
**1. ZPŮSOB** – ZJIŠŤUJEME ZBYTKY PŘI POSTUPNÉM DĚLENÍ ČÍSLEM 2.

$(49)_D = (\ ?\ )_B$ $\qquad$ $(49)_D = (110001)_B$

49 : 2 = 24 ... 1
24 : 2 = 12 ... 0
12 : 2 = 6 ... 0
6 : 2 = 3 ... 0
3 : 2 = 1 ... 1
1 : 2 = 0 ... 1

Příkaz – ulož č. $(49)_D$ do RAM do BYTU s adresou 3

UŽIVATEL → PŘEVODNÍK →
KLÁVESNICE $(49)_D \to (110001)_B$
[4] [9] [ENTER]




**2. ZPŮSOB** – TABULKA

| | $2^6$ | $2^5$ | $2^4$ | $2^3$ | $2^2$ | $2^1$ | $2^0$ | $2^{-1}$ | $2^{-2}$ |
|---|---|---|---|---|---|---|---|---|---|
| ... | 64 | 32 | 16 | 8 | 4 | 2 | 1 | 0,5 | 0,25 ... |
| | | 49 − 32 = 17 | 17 − 16 = 1 | | | | 1 − 1 = 0 | | |
| | | 1 | 1 | 0 | 0 | 0 | 1 | | |

PŘEVOD B → D

UVĚDOMIT SI:

$147 = 1 \cdot 10^2 + 4 \cdot 10^1 + 7 \cdot 10^0$ (210)

$(10101)_B = (\ ?\ )_D$ (4 3 2 1 0) $\qquad$ $(10101)_B = (21)_D$

$\to 1 \cdot 2^4 + 0 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 =$
$= 1 \cdot 2^4 + 1 \cdot 2^2 + 1 \cdot 2^0 = 16 + 4 + 1 = 21$

$(101)_D = (?)_B$ $(101)_D = (1100101)_B$

101 : 2 = 50 ... 1
50 : 2 = 25 ... 0
25 : 2 = 12 ... 1
12 : 2 = 6 ... 0
6 : 2 = 3 ... 0
3 : 2 = 1 ... 1
1 : 2 = 0 ... 1

KONTROLA :

1 1 0 0 1 0 1

64 32 16 8 4 2 1

101

C:\>DOS\FORMAT.I A: - FORMÁTOVÁNÍ DISKETY

TO TAM JE

LABEL - NÁVĚŠTÍ DISKU

AKTIVNÍ HD

A: BUDE AKTIVNÍ A

DIR CO JE ULOŽENO

A > C:\TG02\TG02 <ENTER> - spuštění editoru TG02 z aktivního disku A

CAPS LOCK - přepínač malá velká písmena

SHIFT - přesmykač

A> DIR/P -

C > CD\TG02

↓

C:\TG02>

C:\CD..

TG02 EXE

C:\HRY\PRINCE>

2.12.

ASCII kód - americký standartní kód (ZNAK → KÓD(číslo))
- mezinárodní norma

1. Ulož do RAM znak A

RAM

01000001

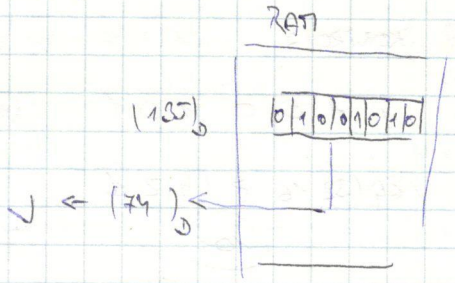
A => ASCII KÓD ZNAKU A

$(65)_D$

→ převod č. 65 do dvojkové čís. soustavy → $(01000001)_B$

PŘEVODNÍK

2. Přečti obsah bajtu s adresou $(135)_D$ v RAM




| | | |
|---|---|---|
| J | 74 | 0 1 0 0 1 0 1 0 |
| A | 65 | 0 1 0 0 0 0 0 1 |
| R | 82 | 0 1 0 1 0 0 1 0 |
| O | 79 | 0 1 0 0 1 1 1 1 |
| S | 83 | 0 1 0 1 0 0 1 1 |
| L | 76 | 0 1 0 0 1 1 0 0 |
| A | 65 | 0 1 0 0 0 0 0 1 |
| V | 86 | 0 1 0 1 0 1 1 0 |

SOUBOR - práce v TOOL -> J.TXT
(C, A, B)
- obsahuje informace (textové obrázek, tabulka...) nebo program (zpracování informací)
- každý soubor má jméno, může mít příponu

JMÉNO [. PŘÍPONA]
- 1-8 znaků (písmena, číslice, _ - podtržítko)
  NE ;!:
- 1-3 znaky

ZÁVAZNÉ PŘÍPONY SOUBORŮ

EXE (EXEcute) } PROGRAMY VE STROJOVÉM KÓDU – JSOU PŘÍMO SPUSTITELNÉ ZAPSÁNÍM JMÉNA SOUBORU
COM (COMmand) }
BAT (BATCH) - DÁVKOVÉ SOUBORY (AUTOEXEC.BAT)
SYS (SYSTEM) - SYSTÉMOVÉ SOUBORY
BAK (BAcKup) - ZÁLOŽNÍ SOUBORY
PAS (PASCAL) - ZDROJOVÝ PROGRAM PASCALU
BAS (BASIC) - BASICU
TXT (TEXT) - TEXTOVÝ SOUBOR
DAT (DATE) - DATOVÝ SOUBOR

10.12.

## HVĚZDIČKOVÁ KONVENCE

- způsob označení skupiny souborů
- používá 2 symboly

[*] ..... zastupuje libovolnou skupinu znaků od pozice * do konce jména (přípony) souboru

jak určím skupinu souborů?

JAN*.TXT ← MASKA NA SOUBORY

1
A1
A2
EK
OTKA

JAN*.T*

*.* - všechny soubory

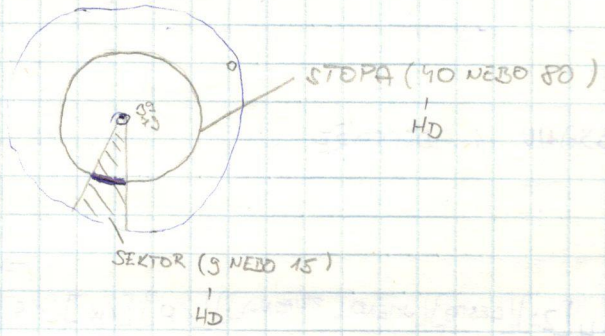
[?] ... zastupuje 1 libovolný znak

KAREL.TXT
PAVEL.TXT
JARDA.TXT
HAVEL.TXT
JIRI.TXT

?A?EL.TXT
?A?EL.*
*L.*

ADRESÁŘ - seznam (tabulka) souborů
DIRECTORY - obs. informace o souborech

① UKLÁDÁNÍ SOUBORU NA DISKU - STOPA A SEKTOR



2. jméno a přípona souboru

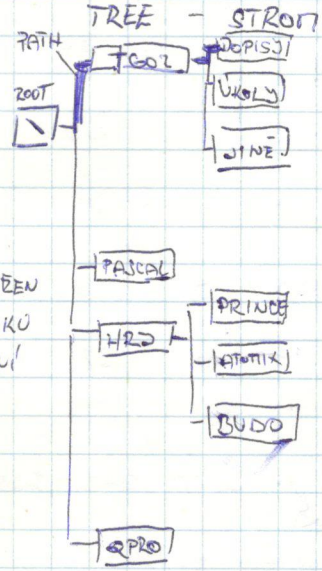
3. velikost souboru v B

4. datum vytvoření (aktualizace) souboru

5. čas ——— " ———

## STROMOVÁ STRUKTURA ADRESÁŘŮ

PATH - CESTA
ROOT - KOŘEN
TREE - STROM




KOŘEN JE VYTVOŘEN NA KAŽDÉM DISKU PŘI FORMÁTOVÁNÍ

16.12.

PŘÍKAZY MS-DOSu PRO PRÁCI S ADRESÁŘI

1. DIR (DIRECTORY-ADRESÁŘ)

VER(VERSION) - verze MS-DOSu < 4.01 / 5.00

↳ ON-LINE HELP

C> DIR /? <ENTER>

(PRINT SCREEN) - tištění obrazovky

VÝPIS OBSAHU 1 ADRESÁŘE

SYNTAXE

C>
B>
A>
:
} DIR ␣ [Z:] [CESTA] [JMÉNO [.PŘÍPONA]] [/P] [/W] [/S] → JEN U V. 5.00

↳ ZAŘÍZENÍ (A:, B:, C:)

↳ úplný popis cesty

/P ... pozastavení výpisu obsahu adresáře po zaplnění monitoru

/W ... vypíš jen jména (anebo přípony) souborů (podadresářů) adresáře
(nevypisuje se: velikost, datum, čas)

/S ... vypíš i obsahy všech podadresářů (SUB DIRECTORIES)

CESTA

```
Displays a list of files and subdirectories in a directory.

DIR [drive:][path][filename] [/P] [/W] [/A[[:]attributes]]
  [/O[[:]sortorder]] [/S] [/B] [/L]

  [drive:][path][filename]
                Specifies drive, directory, and/or files to list.
  /P            Pauses after each screenful of information.
  /W            Uses wide list format.
  /A            Displays files with specified attributes.
  attributes     D  Directories                  R  Read-only files
                 H  Hidden files                 A  Files ready for archiving
                 S  System files                 -  Prefix meaning "not"
  /O            List by files in sorted order.
  sortorder      N  By name (alphabetic)        S  By size (smallest first)
                 E  By extension (alphabetic)   D  By date & time (earliest first)
                 G  Group directories first     -  Prefix to reverse order
  /S            Displays files in specified directory and all subdirectories.
  /B            Uses bare format (no heading information or summary).
  /L            Uses lowercase.

Switches may be preset in the DIRCMD environment variable.  Override
preset switches by prefixing any switch with - (hyphen)--for example, /-W.

C>
```

DALŠÍ PŘÍKAZY MS-DOSu PRO PRÁCI S ADRESÁŘI

2. MD ... vytvoř adresář
   MAKE DIRECTORY

A> MD B:\MESTA
A> MD B:\MESTA\PRAHA
RD - zrušit

! A> MD B:\DATA\DOPISY\MAMA  NE
MÁME  NEMÁME

3. TREE ... vypíš stromovou strukturu adresářů

[/F]

↳ File(s)

Soubory)

A> TREE/F | MORE

↳ PŘÍKAZ MS-DOSu

FILTR

C> TREE\A:/F | MORE

4. CD ... změň pracovní (aktivní) adresář na adr., který je uveden na konci ce

↳ CHANGE

- to zapamatuje poč. je aktivní adresář kořen

A> CD ... otázka - který adr. je aktivní

A:\

A> CD\ mesta\ praha

A:> MESTA\PRAHA> { nastavení výzvy nálezí na použití příkazu PROMPT

PROMPT JARDA $P$G

5. RD - - - - - zruš adresář

↳ REMOVE DIRECTORY

Nelze zrušit adresář: 1) KOŘENOVÝ

2) AKTIVNÍ

3) KTERÝ OBS. PODADRESÁŘ NEBO SOUBORY (NEPRÁZDNÝ ADRESÁŘ)

Adresář je prázdný tehdy, když obsahuje pouze položky

. <DIR> - v každém adresáři

.. <DIR> s výjimkou kořenového

↳ tento samý adresář

↳ nadřízený adresář

A:\MESTA\PRAHA>CD..

..

A:\MESTA>

DELETE

ERASE - ZRUŠENÍ SOUBORU

# NORTON COMMANDER (NC)

nadstavba operačního systému počítače

UŽIVATEL <-> NC <-> OS <-> POČÍTAČ <-> PERIFÉRIE

M 602

HH MANAŽER

WINDOWS 3.1 ... grafické rozhraní (ikony)

Instalace NC - instalační program INSTALL

A> INSTALL <ENTER>

nainstaluje NC na C do adresáře NC (NC.EXE)

A> C:\NC\NC

6. DEL ... zruš soubor(y)
   ↳ DELETE
7. COPY ..... zkopíruj soubor(y)
8. REN ... přejmenuj soubor
   ↳ RENAME

---

MS-DOS

6. DEL [DRIVE:][PATH] FILENAME [/P]

LZE POUŽÍT HVĚZDIČKOVOU KONVENCI

PŘED ZRUŠENÍM KAŽDÉHO SOUBORU BUDE DOTÁZÁNI, ZDA ZRUŠIT ČI NEZRUŠIT

A> DEL ␣ \MESTA\PRAHA\P1.TXT
C> DEL ␣ A:\MESTA\BRATISLAVA*.*

RUŠ. ADRES. - RD
SOUB - DEL

7. COPY

Co chceme kopírovat?
Kam chceme kopírovat?

A> COPY ␣ \MESTA\PRAHA\P1.TXT ␣ \DATA

SOURCE (co má být kop.) — DESTINATION (umístění)

ZDROJ

C> COPY ␣ A:\MESTA\PRAHA\P2.TXT ␣ A:\INFO

A> COPY ␣ \MESTA\BRATISLAVA\B1.TXT ␣ \INFO\BRAT1.*

KOPÍROVÁNÍ SE ZMĚNO JMÉNA KOP. SOB.

zkopíruje a přejmenuje

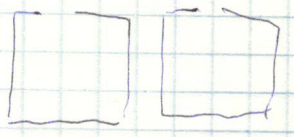
NC

F8 DELETE
- zruší adresáře i soubory

+ - SELECT - výběr

- - UNSELECT

F5 COPY



CO
A:
A:

KAM
A:
C:

8, REN ... přejmenování souboru

REN [DRIVE:][PATH] FILENAME1 ␣ FILENAME 2

↓ STARÉ JMÉNO ↓ NOVÉ JMÉNO

SHRNUTÍ MS-DOSU

1. Příkazy pro práci s adresáři
   1 DIR
   2 MD
   3 CD
   4 RD
   5 TREE
1. Příkazy pro práci se soubory
   1 DEL
   2 COPY
   3 REN
   — — —
3. ostatní příkazy
   TIME
   DATE
   CLS
   TYPE ... zobr. textových souborů na monitoru

ATTRIB
- R - pouze ke čtení
- A - archivační
- S - systémový
- H - neviditelný

→ má ho vytv. soubor

ATTRIB ␣ ± R ␣ SOUBOR(Y)
A
S
H

ATTRIB ␣ SOUBOR
- OTÁZKA

NC projevuje se ?

FILES
ATR.
[X]
[X]

COMMANDS
- FIND FILE - vyhledej soubor(y)
- NCD TREE -

— NCD TREE —

↑ →

↓

SPEED SEARCH : -
RYCHLÉ VYHLEDÁVÁNÍ
ADRESÁŘE

# TURBO PASCAL v 6.0 (BORLAND)

Pascal - vyšší programovací jazyk, N. Wirth (1971)

PROBLÉM => POSTUP ŘEŠENÍ PROBLÉMU ALGORITMUS => PŘEPIS ALGORITMU POMOCÍ PROGRAM. JAZYKA PASCAL => PRÁCE NA POČÍTAČI V INTEGROVANÉM PROSTŘEDÍ T.P.

① ② ③

① ALGORITMIZACE

② ZNALOST PROG. JAZ.

③ ZNALOST PRÁCE V INTEGROVANÉM PROSTŘEDÍ T.P.

≡ FILE . . . . . HELP — HLAVNÍ MENU

DESK TOP

F10 MENU — ON LINE HELP

OTEVŘENÍ SOUBORU
OPEN FILE

ČÍSLO OKNA

(SYMBOL UZAVŘENÍ (MYŠ)) =1= [↕] ZOOM - ZMĚNA VELIKOSTI OKNA (MYŠ)

UKAZATEL

1:1 = SYMBOL ROLOVÁNÍ

ŘÁDEK SLOUPEC

ODSAZENÍ

ULOŽENÍ ZDROJOVÉHO PROGRAMU NA DISK

↓

SAVE - F2

DOČASNÉ OPUŠTĚNÍ TP

<F10> FILE <ENTER> } <ALT><F>

DOS SHELL

# PŘEKLAD (KOMPILACE - KOMPILÁTOR) ZDROJOVÉHO PROGRAMU V TP DO STROJOVÉHO KÓDU

P1.PAS -> [COMPILE] -> P1.EXE

(2) COMPILE Alt-F9 <ENTER>

(1) UMÍSTĚNÍ <- DESTINATION [redacted]

1. DISK ... umístění zkomp. prog. na disk (C, A, B, D)

2. MEMORY ... RAM

WRITELN (' ') ... USES - připojení jednotek (i vše najed

TP — CRT, GRAPH, ...

CLRSCR - smaže obr.

## SLOŽENÍ PROGRAMU V PASCALU

má 3 části

1. část - HLAVIČKA PROGRAMU

[redacted] program ␣ [P1] ;

JMÉNO PROG.

~~P1.PAS~~ DUPLICITNÍ IDENTIFIKÁTOR

2. část - ČÁST DEFINICÍ A DEKLARACÍ

DEKLARACE - ZVEŘEJNĚNÍ PROMĚNNÝCH

! Nejdříve musíme proměnnou deklarovat, potom ji můžeme použí

VARiable - PROMĚNNÁ

var ␣

(1) Víme, že v programu budu používat proměnné a, b, c, na které budou dosazována celá čísla.

var ␣ A, B, C : integer;

(2) Víme, že v programu budu potřebovat proměnné x, y, na které budou dos. reálná čísla

INTEGER ... standardní označení datového typu celých čísel

REAL .. stand. označení datového typu reálných čísel

var ␣ x, y : real;

③ var ␣ A, B, C : integer;
X, Y : real;

3. část - PŘÍKAZOVÁ ČÁST

begin { BEGIN - ZAČÁTEK PŘÍK. ČÁSTI PROGRAMU }
zápis jednotlivých příkazů Paskalu,
každý příkaz je ukončen " ; "

end. { END - KONEC PŘÍKAZOVÉ ČÁSTI PROGRAMU }

- J
  - real - E
  - ordinální
    - celá č. - E
    - char
    - boolean
    - výčtový
- S
  - pole
  - záznam
  - množina
  - ŘETĚZEC - string
- UKAZATEL

PROGRAMOVÁNÍ ZNÁMÝCH VZORCŮ

① Napište prog., který vypočítá a vytiskne na obr. pouze kvádru o hranách a, b, c

S = 2(ab + bc + ac)
S = 2 · (15 + 35 + 21)
S = 2 · 71
S = 142 $cm^2$

a = 3 cm
b = 5 cm
c = 7 cm
← VSTUPNÍ DATA

→ VÝSTUP
$S, a, b, c \in R^+$

```
program ␣ OBSAH;
var ␣ S,A,B,C : real;
begin
  writeln ('ZADEJ DÉLKU HRAN KVÁDRU');
  write   ('A␣=␣');
  readln  (A);
  write   ('B␣=␣');
  readln  (B);
  write   ('C␣=␣');
  readln  (C);
  S:= 2*(A*B + A*C + B*C);
  writeln ('POVRCH KVÁDRU JE␣', S:4:2);
end.
```

Napište program pro výpočet objemu kvádru o hr. a, b, c.

```
program OBJEM;
var V, A, B, C: real;
begin
 writeln('Zadej délku stran');
 write('A = ');
 readln(A);
 write('B = ');
 readln(B);
 write('C = ');
 readln(C);
 V := A * B * C;
 writeln('Objem kvádru je ', V:4:2);
end.
```

Napište program pro výp. obsahu kruhu o poloměru r.

$S = \pi r^2$ $\pi = 3{,}14159\ldots$

výpočty vstup → KONSTANTA

DEKLARACE KONSTANTANT

```
const PI = 3.14;
```

PI → JMÉNO KONSTANTY

3.14 → HODNOTA KONSTANTY

```
const PI = 3.14;
      R = 'KONEC';
```

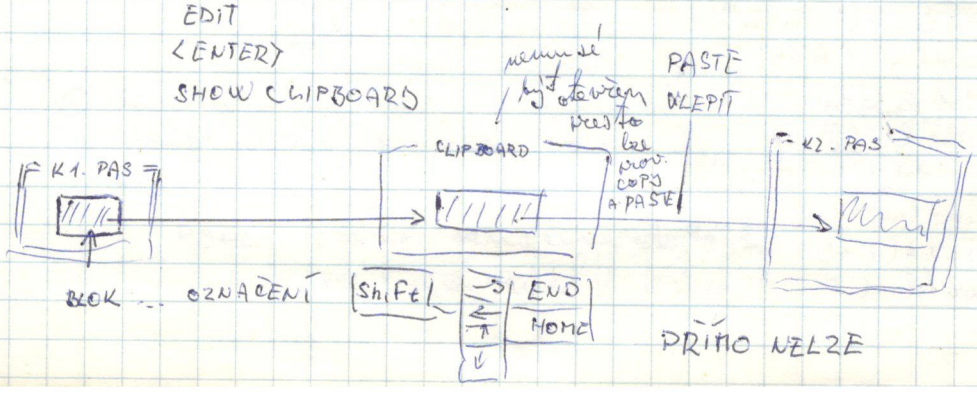
R → řetězcová konstanta

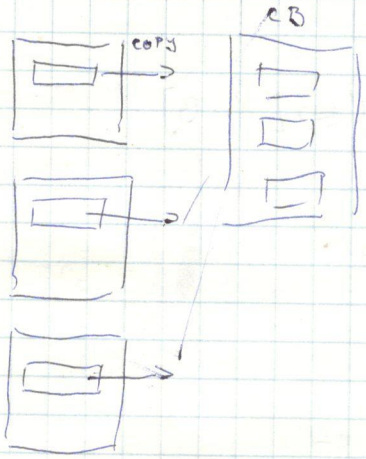
```
program KRUH;
uses CRT;
const PI = 3.14;
var S, R: real;
begin
  CLRSCR;
  writeln('Zadej pol. kruhu');
  write('R = ');
  readln(R);
  S := PI * R * R;
  writeln('Obsah kruhu je ', S:4:2);
end.
```

CLIPBOARD - ODKLÁDACÍ OKÉNKO (SCHRÁNKA)

1. otevření - <F10>
   EDIT
   <ENTER>
   SHOW CLIPBOARD

2.



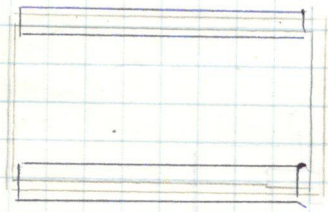



## Použití ASCII kódu při psaní programu

ASCII kód – 1 BYTE (0 – 255) | ZNAK | KÓD (číslo)

Chceme napsat program, který vytiskne násl. rámeček:

WRITELN('



');

Pro nakr. rám. budu potř. znaky:

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| ═ | 205 | ─ | 196 | á | 160 | ó | 162 | Š | 155 |
| ╒ | 201 | ┌ | 218 | Á | 143 | Ó | 149 | š | 168 |
| ╕ | 187 | └ | 192 | ä | 142 | ú | 163 | č | 128 |
| ╚ | 200 | ┐ | 191 | ä | 132 | Ú | 151 | ř | 135 |
| ╝ | 188 | ┘ | 217 | é | 130 | ů | 150 | Ř | 158 |
| ╓ | 186 | │ | 179 | É | 144 | Ů | 166 | Ť | 169 |
| ╟ | 206 | ┤ | 180 | ě | 137 | ž | 145 | ť | 134 |
| ╟ | 204 | ├ | 196 | ě | 136 | Ž | 146 | ť | 159 |
| ╟ | 185 | ┴ | 193 | í | 139 | ď | 131 | ý | 157 |
| ╟ | | ┬ | 194 | Í | 161 | Ď | 133 | ý | 152 |
| ╟ | | ┼ | 197 | | | | | | |
| ╟ | 202 | | | | | | | | |
| ╟ | 203 | | | | | | | | |

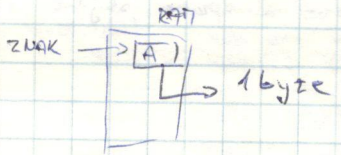
PROMĚNNÁ V PASCALU

- místo v operační paměti počítače, jehož obsah se může při běhu programu měnit

- var A: integer; — RAM: A

- var ZNAK: char; {CHARAKTER} — ZNAK

- var X: real; — RAM: X



boolean

| | | |
|---|---|---|
| $\wedge$ | and | |
| $\vee$ | or | |
| $'$ | not | |

$[A \Rightarrow B]' \Leftrightarrow [A \wedge B']$

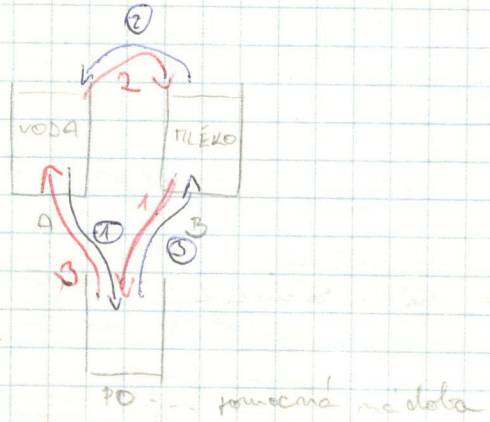
$[[A \Rightarrow B']' \Leftrightarrow [A \wedge B']' \Leftrightarrow [A' \vee B]$

$[A \Rightarrow B] \Leftrightarrow [A' \vee B]$

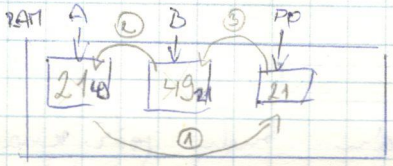
PRAVDA — NEPRAVDA

TRUE — FALSE

## ALGORITMUS ZÁMĚNY HODNOT DVOU PROMĚNNÝCH




var A,B,PO: integer;



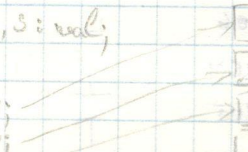

PŘÍKAZ PŘIŘAZENÍ

- zápis v Pascalu

:= - nelze psát jen : nebo =
- nedělitelný symbol

var A: integer;
begin
A:=48;
end.

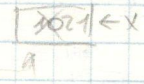
var A,B,C,S: real;
begin
A:=2.4;
B:=2.6;
C:=2.2;
S:=(A+B+C)/3;
end.




NEJDŘÍVE JE VYP. HODNOTA VÝRAZU (A+B+C)/3

2,4

var X: integer;
begin
X:=10;
X:=2*X+1; - vlevo a vpravo stejná prom.
end.



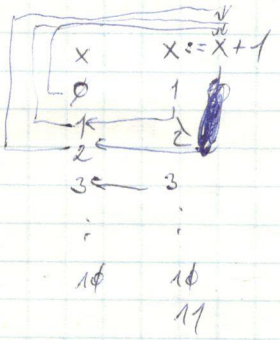

PASCAL — MATEM.

X:=X+1 — X=X+1, 0=1 NŘ

- stará hodn. X
- nová hodnota X

X:=X+1 10X
↳ 0

Pro hodn. X, počínaje hodn. 0, po hodn. 10 v kroku 1 pokračuj.




napište program, který zamění hod. 2 prom. A,B typu real.

```
program ZAMENA;
var A,B,X:real;
begin
  A:=23,4;
  B:=71,6;
  X:=A;      X:=B;
  A:=B;      B:=A;
  B:=X;      A:=X;
  writeln('A=',A,'B=',B);
end.
```

PŘÍKAZ VSTUPU

V= abc

} vstupní data

- k zadávání vstupních dat. Vstupní data tvoří vstupní soubor.
  ↓
  INPUT

- pokud zadáváním vstupní data z klávesnice ——→

1) READ( )
   seznam proměnných oddělených čárkou

- přečti hodnoty a ulož je v souboru INPUT na tom řádku, z kterého byly hodnoty načteny

2) READLN( )
   ČTI LINE

- přečti hodnoty a ulož na začátek následujícího řádku souboru INPUT

INPUT

| | | |
|---|---|---|
| -98 ␣ 128 ␣ -64 | | |
| 25 ␣ 68 ␣ 1 | | |
| 11 ␣ 27 | | |
| | | |
| | | |
| | | |
| | | |

```
var A,B,C: integer;
  :
  read(A);   -98
  read(B);   128
  read(C);   -64
  :
```

```
read(A,B,C);
-98,128,-64
```

```
readln(A);   -98
read(B);     25
read(C);     68
readln(A,B);  -98, 128
read(C);     25
```

```
readln(A,B,C);
-98 128 -64
```

## PŘÍKAZ VÝSTUPU

soubor VYSTUP1.AS

| | |
|---|---|
| (1) | DANA␣JANA␣HANA⊠ |
| (2) | DANA␣JANA<br>HANA⊠ |
| (3) | DANA<br>JANA<br>HANA⊠ |

- slouží k výstupu výstupních dat. Výstupní data tvoří výstupní soubor.

↓

OUTPUT ... MONITOR

1) WRITE ( ) - zapíš a zůstaň na řádku
PIŠ

seznam proměnných odd. čárkou - write (A,B,C);
a řetězců - write('A␣=␣', A, 'B␣=␣', B);

2) WRITELN

- zapíš a přejdi na začátek násl. řádku

## PARAMETRY PŘÍKAZU VÝSTUPU

řetězec - 'JANA'

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | ... | 79 | 80 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| I=4 | | | | | | | | | | | | | | | |
| I=12 | | | | | | | | | | | | | | | |
| I=2 | | | | | | | | | | | | | | | |
| I=3 | | | | | | | | | | | | | | | |
| I=10 | | | | | | | | | | | | | | | |
| I=6 | | | | | | | | | | | | | | | |
| * I=8 | | | | | | | | | | | | | | | |
| I=1 | | | | | | | | | | | | | | | |
| I=80 | | | | | | | | | | | | | | | |
| I=81 | | | | | | | | | | | | | | | |

## PODMÍNĚNÉ PŘÍKAZY

1, neúplný podmíněný příkaz

if [ ] then [ ];
podmínka příkaz

jestliže platí PODMÍNKA, potom proveď PŘÍKAZ;

2, úplný podmíněný příkaz

if [ ] then [ ]
podmínka příkaz 1
else [ ];
příkaz 2

jestliže platí podmínka, potom proveď příkaz 1
jinak proveď příkaz 2.
Příkaz 2 se provádí, když podmínka není

## 3) Příkaz CASE

PODMÍNKA
- výraz typu boolean (typ logických hodnot)

↓

hodnota výrazu buď platí, nebo neplatí.
1 > 0
TRUE > FALSE

-2
5
↓
A>0

→ podmínka A>0 je true. (pod. platí)
false.

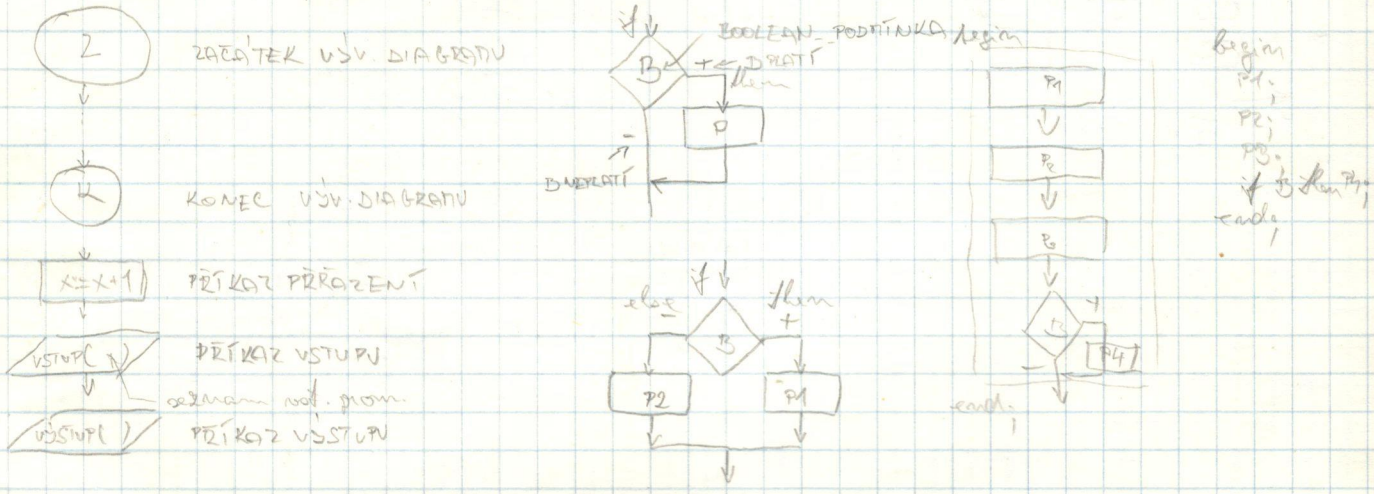
Např. program, který vytiskne informaci "ČÍSLO KLADNÉ", v případě je buch zadané kladné celé číslo, jinak se neprovede nic.

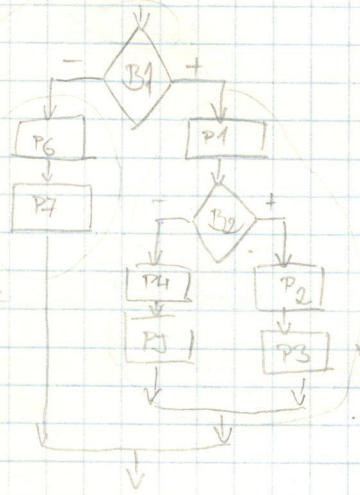
PODM1.PAS

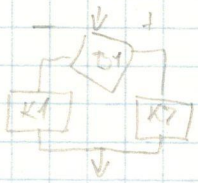
SLOŽENÝ PŘÍKAZ
je to více než jeden příkaz

```
begin { ZAČÁTEK SL. P.}
  P1;
  P2;
  :
  PN;
end; { KONEC SL. P.}
```

VÝVOJOVÉ DIAGRAMY
jsou to značky, pomocí kterých zapisujeme algoritmy - postupy řešení úlohy.





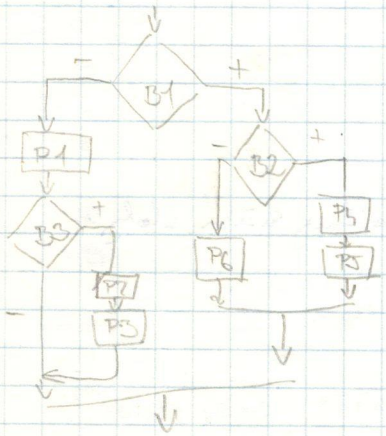




```
if B1 then K2
      else K1;
```

```
K1   begin
       P6;
       P7;
     end;

K2   begin
       P1;
       if B2 then begin
                    P2;
                    P3;
                  end
                else begin
                    P4;
                    P5;
                  end
     end
```

```
if B1 then begin
             P1;
             if B2 then begin
                          P2;
                          P3;
                        end
                      else begin
                          P4;
                          P5;
                        end
           end
      else begin
             P6;
             P7;
           end;
```




```
if B1 then if B2 then begin
                        P4;
                        P5;
                      end
                    else P6
      else begin
             P1;
             if B3 then begin
                          P2;
                          P3;
                        end
           end;
```

Sestavte výv. diagr. a mat. program řešení rovnice

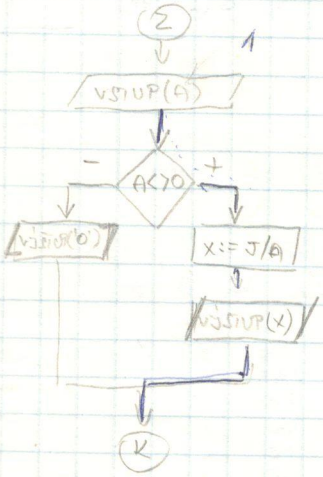
$a \cdot x = 5$ , $a, x \in R$

a ... parametr

x ... neznámá

$a \neq 0$ $\left.\begin{array}{ll} 5x = 5 & x = 1 \\ -10x = 5 & x = -\frac{1}{2} \\ 6x = 5 & x = \frac{5}{6} \end{array}\right\}$ 1 řešení $x = \frac{5}{a}$

$a = 0$ $0x = 5$ - - - - 0 řešení

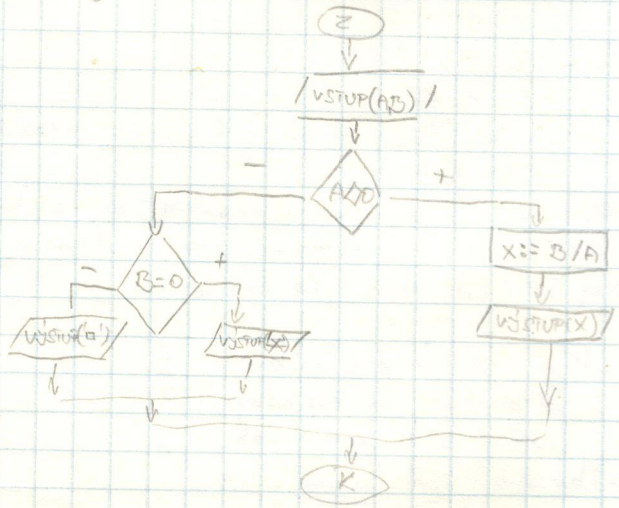



čím později se začně program psát, tím obtížnější je ladění.

$ax = b$

$A \neq 0$ $\left\{\begin{array}{ll} 2x = 4 \\ -3x = 17 \\ 0x = 0 \end{array}\right\}$ 1 řešení $x = \frac{b}{a}$

$a = 0$ $\left\{\begin{array}{ll} 0x = 0 & \text{nek. mn. ř.} \\ 0x = 5 & \underbrace{0 \text{ ř.}}_{0} \quad \times \end{array}\right.$ $B = 0$ $B \neq 0$




IF2.PAS

Kv. r. v R.

$ax^2 + bx + c = 0$ $\quad a, b, c \in R$

$a \neq 0$

$x^2 - 3x + 2 = 0$

$x^2 + 6x + 10 = 0$

$D = b^2 - 4ac = 36 - 4 \cdot$

$= 36 - 40 = \underline{-4}$

if A=0 then writeln('R. není kvd')

else 'ŘEŠENÍ KV. ROVNICE'

22.9.

$\rightarrow |x| = a$ $\quad$ x ... reálná neznámá

a ... reálný parametr

$$\begin{cases} |x| = 4 \ldots x_1 = 4,\ x_2 = -4 \\ |x| = 0 \ldots x = 0 \\ |x| = -3 \ldots \text{nemá reál. kořen} \end{cases}$$

jestli $a > 0 \Rightarrow$ 2 kořeny

$x_1 = a$

$x_2 = -a$

jinak $\{a \leq 0\}$ jestliže $a = 0$, pak má r. 1 kořen

$x = 0$

jinak nemá r. řešení

```
if A>0 then begin
            x1:= A;
            x2:= -A;
            writeln('Rovnice má 2 řešení:', x1:6:3, 'a', x2:6:3);
          end;
  else if A=0 then begin
                   x1:=0
                   writeln('R. má 1 řešení:' x1:6:3);
                   end;
                 else writeln('R. nemá řeš');
```

Příkaz case v Borealu

- podmíněný příkaz
- úloha: napsat př., který vytiskne název měsíce v roce podle čísla měsíce.

```
read(n); {n.. číslo měsíce}
if n=1 then writeln('LEDEN');
```

12 PROSINEC

=>

CELEKTOR (nesmí být typu n: real;)

```
case n of
  1: writeln('Leden');
  2: writeln('Únor');
  else writeln('VOLE?!)
end;
```

úloha na př. CASE

např. př. který po zadání čísla měsíce vytiskne počet dní v tomto měsíci.

| | |
|---|---|
| 1 | 31 |
| 2 | 28 |
| 3 | 31 |
| 4 | 30 |
| 5 | 31 |
| 6 | 30 |
| 7 | 31 |
| 8 | 31 |
| 9 | 30 |
| 10 | 31 |
| 11 | 30 |
| 12 | 31 |

některé měsíce mají 31 – 1, 3, 5, 7, 9, 10, 12
30 – 4, 6, 9, 11
28 – 2

CASE2.PAS

```
case (n) of
  1,3,5,7,8,10,12: writeln('31 dní');
  4,6,9,11 : writeln('30 dní');
  2: writeln('28 dní');
  else: writeln('?');
end;
```

FOR cyklus

- cyklus se známým počtem opakování
- syntaxe for I:=P to K do PRIKAZ;
  pro hodnoty I od h. P až po h. K (s krokem 1) provádej PRIKAZ.

  I – řídící proměnná cyklu
  P – počáteční hodnota
  K – konečná hodnota
  I, P, K – musí být ordinálního typu – integer...
  - char
  - boolean
  - výčtový

~~for I,P,K: real;~~

př. 1.

Prg., který sečte prvních N přirozených čísel.

$n = 7$ $\quad 1+2+3+4+5+6+7$

SOUČET ① SOUCET := 0;

1. sčítance = 1 $\quad$ SOUCET := SOUCET + 1

2. sčítance = 2 $\quad$ SOUCET := SOUCET + 2

3 $\quad$ 3 + 2

7. sčítance = 7 $\quad$ SOUCET := 5 + 7 $\quad$ I

28

② for I := 1 to N do SOUCET := SOUCET + I

př. 2

součin prvních n přir. čísla děl. 5

$n = 6$ $\quad$ 5, 10, 15, 20, 25, 30

| 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|
| 1·5 | 2·5 | 3·5 | 4·5 | 5·5 | 6·5 |

I

5 * I

① SOUČIN := 1;

② for I := 1 to N do SOUČIN := SOUČIN * I * 5

simulační tabulka:

| N | I | SOUCIN |
|---|---|---|
| 6 | | 1 |
| | 1 ≤ 6 | 1 * 5 = 5 |
| | 2 ≤ 6 | 5 * 5 * 2 = 50 |
| | 3 ≤ 6 | 50 * 5 * 3 = 750 |
| | 4 ≤ 6 | 750 * 5 * 10 = 15000 |
| | 5 ≤ 6 | 15000 * 5 * 5 = 375000 |
| | 6 ≤ 6 | 375000 * 5 * 6 = 11250000 |
| | 7 ≤ 6 | |

Napište program pro určení součtu prvních n přir. čísel dělitelných 2

```
program SCITANI;
uses CRT;
var I, SOUCET, N: integer;
begin
clrscr;
writeln('Zadej číslo?');
readln(N);
SOUCET:=0;
for I:=1 to N do
  SOUCET:=SOUCET+(I*2);
writeln('Soucet je', SOUCET);
end.
```

DATOVÝ TYP BOOLEAN (TYP LOGICKÝCH HODNOT)
↳ G. BOOL

var B: boolean;
↳ proměnná B může mít hodnotu → true
↳ false

- operace: and ... ($\wedge$) .. log. součin
  or ... ($\vee$) .. součet - nevylučovací nebo
  xor ... ($\bar{\vee}$) .. vylučovací nebo
  not ( ' ) .. negace

? $\Rightarrow$
$\Leftrightarrow$

Vyjádřete implikaci $A \Rightarrow B$ pomocí operací $\wedge, \vee, '$.

1 .... $[A \Rightarrow B]' \Leftrightarrow [A \wedge B']$
2 .... $(X')' \Leftrightarrow X$
3 .... $[A \wedge B]' \Leftrightarrow [A' \vee B']$

$$[A \Rightarrow B] \underset{2}{\Leftrightarrow} [(A \Rightarrow B)']' \underset{1}{\Leftrightarrow} [A \wedge B'] \underset{3}{\Leftrightarrow} [A' \wedge (B')'] \underset{2}{\Leftrightarrow} [A' \wedge B]$$

(not(A)) and (B)

true 1
False 0

False<true

| A | B | A and B | A or B | A xor B | not(A) |
|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 0 | 0 |
| 1 | 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 1 | 1 | 1 |
| 0 | 0 | 0 | 0 | 0 | 1 |

- relace < > <= >= <>

- výrazy typu boolean

jsou výrazy, které zapisujeme pomocí proměnných, operací and, or, xor, not, relací a závorek.

true nebo false

```
var A,B : integer;
    BOOL: boolean;
    :
    :
read (A,B);
BOOL:= A<=B
```

J 10
J 1

J<= 7
true

```
if BOOL then writeln ('ANO')
        else writeln ('NE')
```

BOOL = true

Napište program, který zobrazí ANO v případě, že trojice celých čísel A,B,C je uspořádána sestupně (A >= B >= C), jinak se vytiskne NE

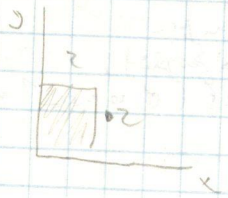
(A >= B) and (B >= C)
7 3 3 5
TRUE FALSE
FALSE

△

a = 7 b = 5 c = 4

A+B > C
A+C > B
B+C > A
podm. musí platit současně

```
var A,B,C : integer;
    BOOL: boolean;
begin
    read (A,B,C);
    BOOL:= (A+B>C) and (A+C>B) and (B+C>A);
    if BOOL then
        :
    else ... ;
```



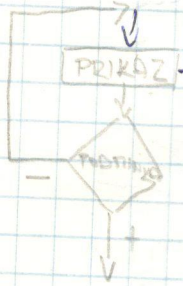

X
[x, y]
[3, 1]
[1, 1]

$(x \leq 2)$ and $(y \leq 2)$ and $(x \geq 0)$ and $(y \geq 0)$

```
read (x, y);
BOOL := (x>=0) and (y>=0) and (x<=2) and (y<=2);
if BOOL then writeln ('ZASAH')
        else writeln ('VEDLE');
```

Repeat cyklus v Pascalu

- cyklus s podmínkou na konci cyklu
  - ↳ výraz typu boolean




→ příkaz se provede alespoň jednou

```
repeat
   PRIKAZ
until PODMINKA
```

opakuj příkaz dokud neplatí podmínka

je dána řada čísel (kladných) zakončená číslem 0.

A 4 3 2 1 0

↳ označuje libovolné číslo v dané řadě

čti čísla A vždy tak dlouho, dokud je přečtené číslo kladné (různé od 0)

je-li A 0 čtení čísel ukonči.

↳ read (A);

```
repeat
   read (A);
until A=0;
```

4 3 2 1 0

NE ← 4=0 / 3=0 / 2=0 / 1=0

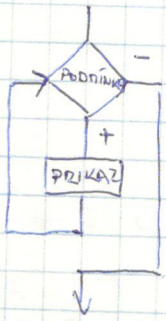
... AND – PODMÍNKA UKONČENÍ CYKLU

Máme schované účtenky z několika nákupů a chceme zjistit, kolik jsme průměrně při jednom nákupu. Je dána řada kladných celých čísel (ceny jedn. nákupů), zakončená nulou. Vypočítejte ar. průměr těchto čísel (0 čísla 1 nákup)

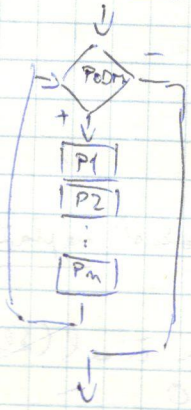
1. 20
2. 100
3. 80
4. 70

```
program UCTY;
uses CRT;
var SOUCET, KOLIK, PORADI : integer;
    PRUMER : real;
begin
  clrscr;
  writeln('Zadávej ceny nákupů');
  SOUCET:=0;
  PORADI:=0;
  repeat
    readln(KOLIK);
    SOUCET:=SOUCET+KOLIK;
    PORADI:=PORADI+1;
  until KOLIK=0;
  PRUMER:=SOUCET/PORADI;
  writeln('Utratil jsi průměr.', PRUMER:6:2);
end.
```

WHILE cyklus – cyklus řízený podmínkou na začátku cyklu.
↳ výraz typu boolean

① 


PŘÍKAZ ... jednoduchý nebo složený

② 


Popis v Pascalu:

① while PODMÍNKA do PŘÍKAZ;

② while PODMÍNKA do
```
begin
  P1;
  P2;
  :
  Pn;
end;
```

Ve WHILE cyklu se příkaz nemusí provést.

na vstupu je přir. číslo n. Např. program, který zjistí a postupně vytiskne všechny číslice pomocí kterých je ~~[illegible]~~ číslo n zapsané.

Př. n = 3726 . . . . mod  10 mod 2 = 0
div  10 div 2 = 5

7456 div 10 = 745 ; 745 div 10 = 74 ; 74 div 10 = 7
7456 mod 10 = 6 ; 745 mod 10 = 5 ; 74 mod 10 = 4
=> 7 div 10 = 0
=> 7 mod 10 = 7

n = 37 | 6
↓ nové N  ↳ CISLICE

1, CISLICE := N mod 10;
↓ 6  ↑ 3726

N := N div 10;
↑ 372 | 6

2, CISLICE := N mod 10;
↓ 2  ↑ 372

N := N div 10;
↑ 37 | 2

3, CISLICE := N mod 10;
↓ 7  ↑ 37

N := N div 10;
↑ 3 | 7

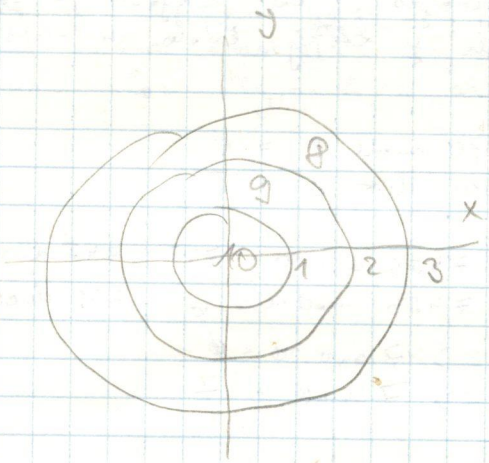
4, CISLICE := N mod 10;
↓ 3  ↑ 3

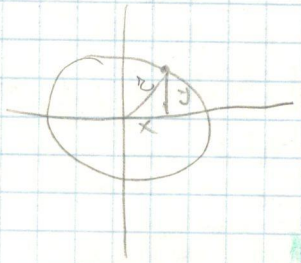
N := N div 10;
! ∅ ↑ 0 | 3

Dokud bude platit N <> ∅, provádej příkazy:

```
CISLICE := N mod 10;
N := N div 10;
write(' ', CISLICE);
```

```
while N <> ∅ do
begin
  CISLICE := N mod 10;
  N := N div 10;
  write(' ', CISLICE);
end;
```




$x^2 + y^2 \leq r^2$

```
function DESET (x1, y1 : real) : boolean;
begin
  DESET := (SQR(x1) + SQR(y1)) <= 1;
end;

function DEVET (x2, y2 : real) : boolean;
begin
  DEVET := (SQR(x2) + SQR(y2) > 1) and
           (SQR(x2) + SQR(y2) <= 2);
end;

function OSM
```

alg: vzestupně seřadit trojici celých čísel A, B, C.

1. KROK vzestupně setř. dvojici čísel A, B

| A | B | C |
|---|---|---|
| 2 | 7 | 4 |
| 2 | 4 | 7 |
| 4 | 2 | 7 |
| 4 | 7 | 2 |
| 7 | 2 | 4 |
| 7 | 4 | 2 |

2 7 4 ①

2 4 7 *

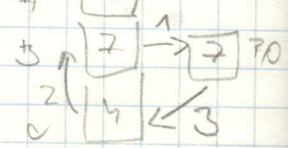
2 4 7 *

4 7 2 ②

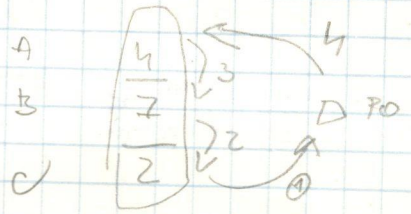
2 7 4

4 7 2

① (A<C) and (C<B)



(2) C<A pak




| VSTUP | VÝSTUP |
|---|---|
| 7 4 2 | 2 4 7 |
| A B C | |

```
procedure SORT( A,B,C : integer; var X,Y,Z : integer);
                    VSTUP                  VÝSTUP
var PO : integer;
begin
  if A>B then begin
                PO := A;
                A := B;
                B := PO;
              end;
  if (A<C) and (C<B) then begin
                          PO := B;
                          B := C;
                          C := PO;
                          end;
  if (C<A) then begin
                PO := C;
                C := B;
                B := A;
                A := PO;
                end;
  X := A;
  Y := B;
  Z := C;
end;
```

Zač. do pr:

```
var AA, BB, CC, XX, YY, ZZ : integer;
[ PR ]
begin
  :
  readln (AA, BB, CC);
  SORT (AA, BB, CC, XX, YY, ZZ);
  writeln ( XX, YY, ZZ);
end.
```

△ABC  9 = 5  7  4  7 5 4

△A'B'C'  8  10  14  14 10 8

## třídění pole

A [24 | -7 | 9 | 11 | 37 | -3 | 6 | 9]
1 2 3 4 5 6 7 8

- neostupné

$A[1] \leq A[2] \ldots \leq A[n]$

- nestupné

obr.

třídicí algoritmy (uměle vnitřních souborů)

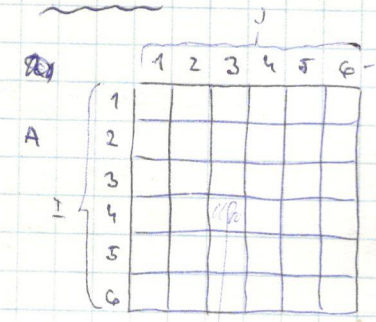
1) BUBLESORT - bublinkové třídění

[4 7 -3 2 -1 5 -6 9] - DANÉ NESOUTĚ. POLE

① zjistit počet nedělení prvků v poli A o N = 10 prvcích

② První vnější číslo přemístit přemístit na poslední pozici v poli A o N = 10 prvcích

# MATICE




type MATICE : array [1..6, 1..6] of Integer;
var A : matice;

A [4, 3]

[6, 6] typ matice A
[M, N] —"—
M = N => čtvercová matice

ZÁKLADNÍ OPERACE S MATICEMI

1, SČÍTÁNÍ

A je typu [M, N]
B je typu [M, N]
C = A + B

$$c_{i,j} = a_{i,j} + b_{i,j}$$

Dána matice typu [6, 6]
Zjistěte v kolika sloupcích matice je stejný součet všech prvků jako v 1 sloupci.

SOUČIN KONSTANTY A MATICE

DÁNO $k \in R - \{0\}$
matice A [m, n]

B = k . A

$b_{i,j} = k \cdot a_{i,j}$

SOUČIN 2 MATIC

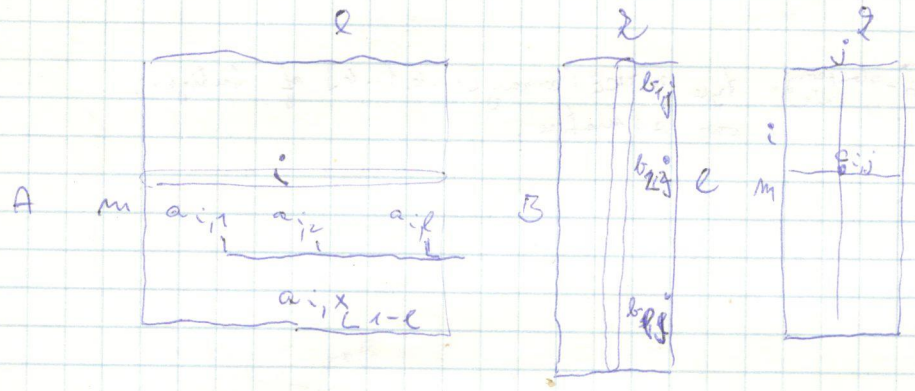
A [m, l]
B [l, k] musí být stejné

C = A · B
[2, 4]

A
2 3 1
1 1 -1
[2, 3]

B
1 1 1 1
-1 1 -1 1
1 0 1 1
[3, 4]

$c_{1,1} = a_{1,1} \cdot b_{1,1} + a_{1,2} \cdot b_{2,1} + a_{1,3} \cdot b_{3,1}$
$c_{1,2} = a_{1,1} \cdot b_{1,2} + a_{2,2} \cdot b_{2,2} + a_{1,3} \cdot b_{3,2}$
$c_{2,1} = a_{2,1} \cdot b_{1,1} + a_{2,2} \cdot b_{2,2} + a_{2,3} \cdot b_{?}$



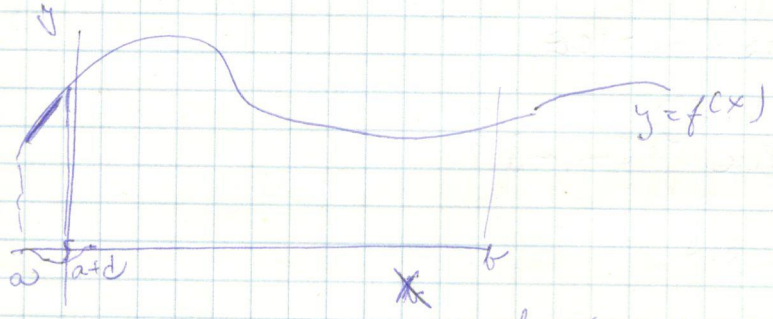

A · B = C

$$c_{i,j} = a_{i,1} \cdot b_{1,j} + a_{i,2} \cdot b_{2,j} + \dots + a_{i,l} \cdot b_{l,j} = \boxed{\sum_{x=1}^{l} a_{i,x} \cdot b_{x,j}}$$

procedure SOUCIN_M (A: Matice1; B: Matice2; var C: Matice3);

Sest. pr. na výpočet délky grafu fce

$f(x) = \dfrac{e^x}{x^2+1}$ v intervalu $\langle a, b \rangle$




$\langle a, b \rangle$ rozdělíme na n stejných dílů

1 díl $\dfrac{|b-a|}{n} = d$

type REAL_FCE = function F(x: real): real;
function MOJE (X1: real): real;
begin
MOJE :=
end;

procedure DELKA (A,B: real; N: int; F1: REAL_FCE
F1
DELKA(AA, BB, NN, MOJE);

# REKURZIVNÍ FCE A PROCEDURY

R = PODOBNOST

REK. FCE

PŘÍMÁ REKURZE - FCE A VYVOLÁ FCI A

Př:

① $n! = n \cdot (n-1)(n-2) \dots 3 \cdot 2 \cdot 1$

$n! = n \cdot (n-1)!$

$n=1 : 1! = 1$

$n=0 : 0! = 1$

$4! = 4 \cdot 3!$

$3! = 3 \cdot 2!$

$2! = 2 \cdot 1!$

$1! = 1$

| | |
|---|---|
| 2 | 1 |
| 6 | 2 |
| 24 | 3 |
| | 4 |

```
function Faktorial (N: Integer): Integer;
begin
  if (N=0) or (N=1) then Faktorial := 1   { nerekurzivní větev }
                    else Faktorial := N x Faktorial (N-1); { rekurzivní větev }
end;
```

ukončení rekurze

jestliže Podmínka -> KONEC REKURZE
jinak

N=5

120 = Faktorial (a)

5 * Fakt (4)

4 * Fakt (3)

3 * Fakt (2)

2 * Fakt (1)

1

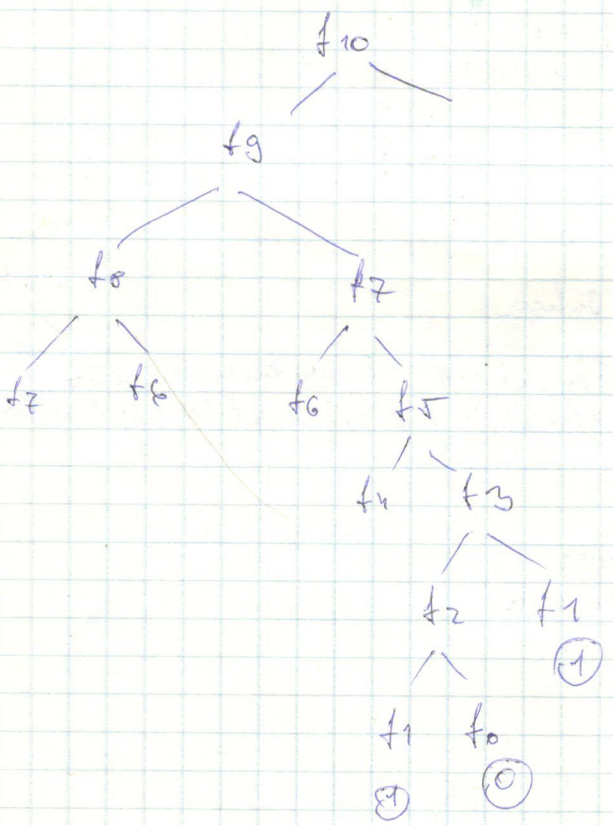
② FIBONACCIMO POSLOUPNOST

$$f_n \begin{cases} n=0 & f_0=0 \\ n=1 & f_1=1 \\ n>1 & f(n-1)+f(n-2) \end{cases}$$

$n=2 \quad f_2 = f_1 + f_0 = 1+0 = 1$

$n=3 \quad f_3 = f_2 + f_1 = 1+1 = 2$

0 1 1 2 3 5 8 13



```
function FIB (N: Integer): Integer;
begin
  if N = 0 then FIB := 0
           else if N = 1 then FIB := 1
                         else FIB := FIB(N-1) + FIB(N-2);
end;
```

REKURZE

| | | | | | |
|---|---|---|---|---|---|
| 0 | 1 | 1 | 2 | 3 | 5 |
| | P | D | T | | |
| | | P | D | T | |
| | | | P | D | T |
| | | | | P | D | T |

ITERACE

① D := 0; T := 1;

② cyklus P := D
D := T
T := P + D

③ NSD - EUKLEIDŮV ALGORITMUS

? NSD(x, y)

x = y  NSD(x, y) = x = y
x > y  NSD(x, y) = NSD(x - y, y)
x < y  NSD(x, y) = NSD(x, y - x)

? NSD(15, 9)

15 > 9 ... NSD(15, 9) = NSD(6, 9)
NSD(6, 9) = NSD(6, 3)
NSD(6, 3) = NSD(3, 3)

MOCNINA

$x^2$ ... Sqr(x)

Rekurze: $a^n = a \cdot a^{n-1}$

HANOJSKÁ VĚŽ

procedure PD(O, K: Integer);
{ přenesení disku }
begin
  Write(O, '=>', K);
end;

procedure PV(Výška, Odkud, Kam, Pomocná);
① PV(VÝŠKA-1, Odkud, Pomocná, Kam)
② PD(Odkud, Kam)
③ PV(Výška-1, Pomocná, Kam, Odkud)

## Řešení soustav rovnic

2 možnosti

1) Determinanty

2) matice - iterační metody → GEM - gaussova eliminační metoda
JEM jordanova

## Determinanty

$$2x - y = 1$$
$$x + y = 3$$

$$x = \frac{4}{3}$$
$$y = \frac{5}{3}$$

SOUSTAVA REGULÁRNÍ - 1 ŘEŠENÍ

$$2x - y = 1$$
$$-4x + 2y = -2 \quad / : (-2)$$

$$\left[\frac{y+1}{2}, y\right] \; y \in R$$
$$[x, 2x+1] \; x \in R$$ - SINGULARITA

$$x + 2y = 1$$
$$x + 2y = 3$$

$$0 = -2$$

NŘ.

$$a_{1,1}x + a_{1,2}y = b_1$$
$$a_{2,1}x + a_{2,2}y = b_2$$

D

$$x = \frac{b_1 - a_{1,2}y}{a_{1,1}} \qquad a_{1,1} \neq 0$$

$$a_{21} \cdot \frac{b_1 - a_{12}y}{a_{1,1}} + a_{22}y = b_2$$

$$a_{2,1} \cdot b_1 - a_{2,1} \cdot a_{1,2}y + a_{2,2}a_{1,1}y = b_2 \cdot a$$

$$y \cdot (a_{1,1} \cdot a_{2,2} - a_{2,1} \cdot a_{1,2}) = b_2 a_{11} - a_{21}$$

$$y = \frac{b_2 a_{1,1} - a_{2,1} b_1}{a_{1,1} \cdot a_{2,2} - a_{2,1} a_{2,2}} \neq 0$$

determinant neznámé y - $D_y$

→ determinant soustavy D

$$x = \frac{b_1 a_{2,2} - a_{1,2} b_2}{a_{1,1} a_{2,2} - a_{2,1} a_{2,2}} \neq 0$$

determinant neznámé x - $D_x$

| | | |
|---|---|---|
| $a_{1,1}$ | $a_{1,2}$ | $b_1$ |
| $a_{2,1}$ | $a_{2,2}$ | $b_2$ |

| | 1 | |
|---|---|---|
| 2 | -1 | 1 |
| 1 | 1 | 3 |

| | 2 | |
|---|---|---|
| 2 | -1 | 1 |
| -4 | 2 | -2 |

3

1. 2 1.

1 2 3

$$D = \begin{vmatrix} a_{1,1} & a_{1,2} \\ a_{2,1} & a_{2,2} \end{vmatrix}$$

VEDLEJŠÍ DIAGONÁLA

hlavní diagonála

−

+

$$D_x = \begin{vmatrix} b_1 & a_{1,2} \\ b_2 & a_{2,2} \end{vmatrix}$$

hl. d.

+

VEDL. DIA.

$$D_y = \begin{vmatrix} a_{1,1} & b_1 \\ a_{2,1} & b_2 \end{vmatrix}$$

$D \neq 0$ → 1 REG. ŘEŠENÍ

| | 1 | 2 | 3 |
|---|---|---|---|
| D | 3 | 0 | 0 |
| $D_x$ | 4 | 0 | 4 |
| $D_y$ | 5 | 0 | 0 |

$$a_{11}x_1 + a_{12}x_2 + a_{13}x_3 = b_1$$
$$a_{21}x_1 + a_{22}x_2 + a_{23}x_3 = b_2$$
$$a_{31}x_1 + a_{32}x_2 + a_{33}x_3 = b_3$$

(1)

$$D = \begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix}$$

$$\begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \end{vmatrix}$$

připsat 1. a 2. řádek

$$:= (1,1 * 2,2 * 3,3) + (2,1 * 3,2 * 1,3) + (3,1 * 1,2 * 2,3) - (1,3 * 2,2 * 3,1) - (2,3 * 3,2 * 1,1) - (3,3 * 1,2 * 2,1);$$

$$D_2 \begin{vmatrix} a_{11} & a_{12} & b_1 \\ a_{21} & a_{22} & b_2 \\ a_{31} & a_{32} & b_3 \\ a_{11} & a_{12} & b_1 \\ a_{21} & a_{22} & b_2 \end{vmatrix}$$

SARRUSOVO PRAVIDLO

$$D_{x_1} = \begin{vmatrix} b_1 & a_{12} & a_{13} \\ b_2 & a_{22} & a_{23} \\ b_3 & a_{32} & a_{33} \\ b_1 & a_{12} & a_{13} \\ b_2 & a_{22} & a_{23} \end{vmatrix}$$

$$D_y \begin{vmatrix} a_{11} & b_1 & a_{13} \\ a_{21} & b_2 & a_{23} \\ a_{31} & b_3 & a_{33} \\ a_{11} & b_1 & a_{13} \\ a_{21} & b_2 & a_{23} \end{vmatrix}$$

(2) ROZVOJ DETERMINANTU

indexy prvku / vynechat sl. a řad. prvku

$$D = \underbrace{(-1)^{1+1} a_{1,1} \begin{vmatrix} a_{22} & a_{23} \\ a_{32} & a_{33} \end{vmatrix}}_{a_{11}a_{22}a_{33} - a_{23}a_{32}a_{11}} + (-1)^{1+2} \cdot a_{1,2} \begin{vmatrix} a_{21} & a_{23} \\ a_{31} & a_{33} \end{vmatrix} + (-1)^{1+3} \cdot a_{1,3} \begin{vmatrix} a_{21} & \\ a_{31} & \end{vmatrix}$$

maticové řešení soustav

$x_1$ — $2x_1 + x_2 + 3x_3 = 1$

$0x_1 \; 2x_2 - x_3 = 2$

$x_2$ — $0x_1 + 0x_2 \; 2x_3 = 4 \rightarrow x_3 = 2$

algoritmus zpětné substituce

→ zobecnění

$$a_{11}x_1 + a_{12}x_2 + \quad a_{1n}x_n = b_1$$
$$0 \cdot x_1 + a_{22}x_2 \quad a_{2n}x_n = b_2$$
$$0x_1 + 0x_2 \quad a_{n-1,n-1}x_{n-1} + a_{n-1,n}x_n = b_{n-1}$$
$$\vdots$$
$$a_{nn}x_n = b_n$$

$$\left(\begin{array}{cccc|c} a_{11} & a_{12} & \cdots & a_{1n} & b_1 \\ 0 & a_{22} & & a_{2n} & b_2 \\ 0 & 0 & \cdots & a_{3n} & b_3 \\ & & & \vdots & \\ 0 & 0 & \cdots & a_{nn} & b_n \end{array}\right)$$

sloupec pravých stran

rozšířená matice soustavy

matice soustavy

$a_{i,i} \neq 0 \quad i = 1 \dots n$

$\Downarrow$

regulární matice

$[x_1, x_2, x_3, \dots, x_n]$

$$x_n = \frac{1}{a_{nn}} \cdot b_n$$

$$x_{n-1} = \frac{1}{a_{n-1,n-1}} \cdot \left(b_{n-1} - a_{n-1,n} \cdot x_n\right)$$

$$\vdots$$

$$x_2 = \frac{1}{a_{22}} \cdot \left(b_2 - \left(a_{23}x_3 + a_{24}x_4 + \dots + a_{2,n}x_n\right)\right)$$

$$x_1 = \frac{1}{a_{1,1}} \cdot \left(b_1 - \left(a_{12}x_2 + a_{13}x_3 + \dots + a_{1n}x_n\right)\right)$$

jak vyjádřit $x_i$?

$$\boxed{x_i = \frac{1}{a_{ii}} \cdot \left( b_i - \sum_{j=i+1}^{n} a_{ij} x_j \right)}$$

## algoritmus přímé substituce

$$\begin{aligned} a_{11} x_1 \qquad\qquad\qquad\qquad &= b_1 \\ a_{21} x_1 + a_{22} x_2 \qquad\qquad &= b_2 \\ a_{n1} x_1 + \dots + a_{nn} x_n &= b_n \end{aligned}$$

$x_1 = \frac{b_1}{a_{11}}$ $\quad a_{11} \neq 0$

$x_2 = \frac{1}{a_{22}} \cdot (b_2 - a_{21} x_1)$

$x_n = \frac{1}{a_{nn}} \cdot (b_n - (a_{n1} x_1 + \dots a_{nn-1} x_{n-1}))$

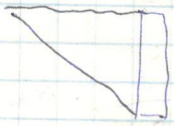
$$\boxed{x_i = \frac{1}{a_{ii}} \cdot \left( b_i - \sum_{j=1}^{i-1} a_{ij} \cdot x_j \right)}$$

$a_{i,i} \neq 0 \quad i = 1, \dots, n$

## gaussova eliminační metoda (GEM)

PRINCIP:

$$\begin{aligned} 2x_1 - x_2 + 3x_3 - x_4 &= 7 \\ x_1 - x_2 + 4x_3 - 2x_4 &= 5 \\ 3x_1 + 2x_2 + x_3 + 4x_4 &= 31 \\ 4x_1 - 3x_2 + 3x_3 - 3x_4 &= -5 \end{aligned}$$

$\rightarrow$ 

1. fáze eliminace

$$\begin{array}{cccc|c} 2 & -1 & 3 & -1 & 7 \\ 1 & -1 & 4 & -2 & 5 \\ 3 & +2 & 1 & 4 & 31 \\ 4 & -3 & 3 & -3 & -5 \end{array} \sim \begin{array}{ccccc} 1 & -1 & 4 & -2 & 5 \\ \hline 2 & -1 & 3 & -1 & 7 \\ 3 & 2 & 1 & 4 & 31 \\ 4 & -3 & 3 & -3 & -5 \end{array} \begin{array}{l} \\ (-2) \\ (-3) \\ (-4) \end{array}$$

$$\begin{array}{ccccc} 1 & -1 & 4 & -2 & 5 \\ \boxed{0} & 1 & 5 & 3 & -3 \\ \hline \boxed{0} & 5 & -11 & 10 & 16 \\ \boxed{0} & 1 & -13 & 5 & -25 \end{array} \begin{array}{l} \\ \\ (-5) \\ (-1) \end{array} \sim \begin{array}{ccccc} 1 & -1 & 4 & -2 & 5 \\ 0 & 1 & 5 & 3 & -3 \\ 0 & 0 & 14 & -5 & 31 \\ 0 & 0 & -8 & 2 & -22 \end{array} \sim \begin{array}{ccccc} 1 & -1 & 4 & -2 & 5 \\ 0 & 1 & 5 & 3 & -3 \\ 0 & 0 & 14 & -5 & 31 \\ 0 & 0 & 4 & -1 & 11 \end{array} \left(-\frac{4}{14}\right)$$

$$\begin{array}{cccccc} 1 & -1 & 4 & -2 & 5 & 12 \\ 0 & 1 & 5 & 3 & -3 & 1-3 \\ 0 & 0 & 14 & -5 & 31 & 15 \\ 0 & 0 & 0 & \frac{0}{14} & \frac{28}{14} & \\ & & & 1 & 5 & \end{array}$$

zpětná substituce

JORDAN

$$\begin{array}{ccccc} 1 & -1 & 4 & 0 & 15 \\ 0 & 1 & -5 & 0 & -18 \\ 0 & 0 & 14 & 0 & 56 \\ 0 & 0 & 0 & 1 & 5 \end{array} \sim \begin{array}{ccccc} 1 & -1 & 4 & 0 & 15 \quad (\cdot 4) \\ 0 & 1 & -5 & 0 & -18 \quad (\cdot 5) \\ 0 & 0 & 1 & 0 & 14 \\ 0 & 0 & 0 & 1 & 5 \end{array} \sim \begin{array}{ccccc} 1 & -1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 2 \\ 0 & 0 & 1 & 0 & 14 \\ 0 & 0 & 0 & 1 & 5 \end{array} \sim$$

$$\sim \begin{array}{ccccc} 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 2 \\ 0 & 0 & 1 & 0 & 3 \\ 0 & 0 & 0 & 1 & 4 \end{array}$$

$$\text{If. } K = \frac{a_{21}}{a_{11}}$$

$$K = \frac{a_{31}}{a_{11}}$$

$$\begin{array}{cccc} a_{11} & a_{12} & a_{13} & b_1 \\ 0 & a_{22}^{(1)} & a_{23}^{(1)} & b_2^{(1)} \\ 0 & a_{32}^{(1)} & a_{33}^{(1)} & b_3^{(1)} \\ 0 & a_{32} & & \end{array} \qquad a_{22}^{(1)} = a_{22} \cdot \left(\frac{a_{21}}{a_{11}}\right) + a_{22}$$

$$\begin{array}{cccc} a_{11} & a_{12} & a_{13} & b_1 \\ 0 & a_{22}^{(1)} & a_{23}^{(1)} & b_2^{(1)} \\ 0 & 0 & a_{33}^{(2)} & b_3^{(3)} \end{array}$$

$$a_{33}^{(2)} = a_{23}^{(1)} \cdot \left(-\frac{a_{32}^{(1)}}{a_{22}^{(1)}}\right) + a_{33}^{(1)}$$

FÁZE ŘÁDEK $i$ SLOUPEC $j$ MULTIPLIKÁTOR ČLEN

$\vdots$

$k$. $\quad k+1, \dots, n \quad k+1, \dots, n+1 \qquad m_{ik} = \dfrac{a_{ik}^{(k-1)}}{a_{kk}^{(k-1)}} \qquad a_{ij}^{(k)} = a_{ij}^{(k-1)} + m_{ik} \cdot a_{kj}^{(k-1)}$

GEM 3

$k = 1, 2, \dots, n-1$ – fáze eliminace

$i = k+1, \dots, n$ – řádek matice

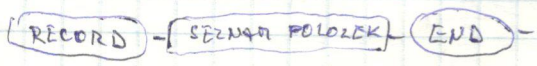
$$m_{ik} = -\frac{a_{ik}^{(k-1)}}{a_{kk}^{(k-1)}}$$ multiplikátor

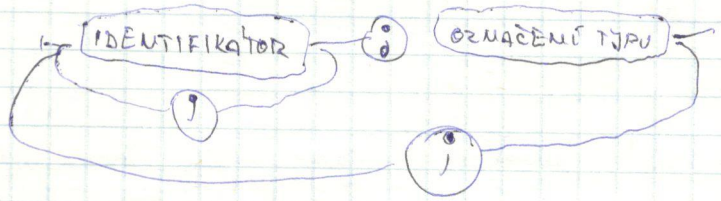
$j = k+1, \dots, n, n+1$

$$a_{ij}^{(k)} = a_{ij}^{(k-1)} + m_{i,k} \cdot a_{k,j}^{(k-1)}$$

TYP ZÁZNAM

- nehomogenní
- obl. ze 2 položek záznamu




WITH

- strukturovaný příkaz

with A do
with B do <=> with A, B do

① nat. př. prov. čtení 2 zlomku z celoč. čit. a jm. a vytváření zlomek = součinu.

② + − : zlomku

③ Proc. + − · : zompl. čísel

④ Proc a alg. do bin.

$\pm$ $(a+bi)+(c+di) = (a+c)\pm(b+d)i$

$\cdot$ $(a+bi)\cdot(c+di) = (ac-bd)+(ad+bc)i$

$: \quad \frac{a+bi}{c+di}\cdot\frac{c-di}{c-di} = \frac{(ac+bd)+(bc-ad)i}{c^2+d^2} = \frac{ac+bd}{c^2+d^2}+\frac{bc-ad}{c^2+d^2}i$

$\frac{4}{5}+\frac{3}{8} = \frac{\overset{32}{4\times8}\quad\overset{+15}{3\times5}}{40} = \frac{47}{40}$

$\frac{2}{3}:\frac{5}{8} = \frac{2\cdot8}{3\cdot5}$

$\frac{8}{48} = \frac{1}{6}$

# TŘÍDÍCÍ ALGORITMY

BUBLESORT

HOFSORT

-3 7 16 -9 0

-9

-9 7 16 -3 0

-9 -3 16 7 0

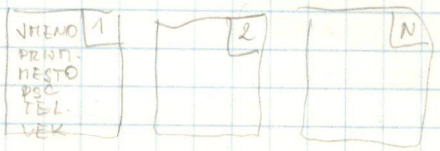
-9 -3 0 7 16

```
for I := 2 to N-1 do
  for J := I to N do
    if x[J] < x[I-1] then prehod (x[J], x[I-1])
```

① Tm. V setr. pole (pole zázn. ~~Eff~~ podle třídící pol.) výběrem Min/Max a jeho uzl. na 1./posl. pozici v poli

② DUBLESORT + příznak SETRIDENO

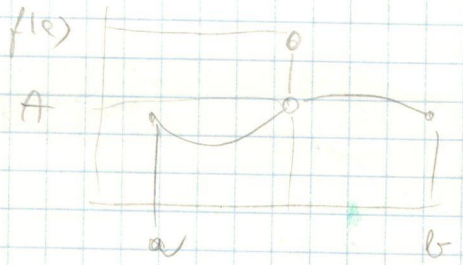
Pole záznamů - výběr
- SORT




Setřídit pole záznamů podle položky VEK

1) tisk celé sestavy na monitor
2) ↑ n. ↓ VĚK => tisk sestavy
3) ↑ n. ↓ PŘÍJMENÍ => tisk sestavy
4) vytisknout všechny záznamy podle uživatelem určené položky jména

## Přibližné řešení rovnic




$\lim_{x \to e} f(x) = f(e)$ – spoj.

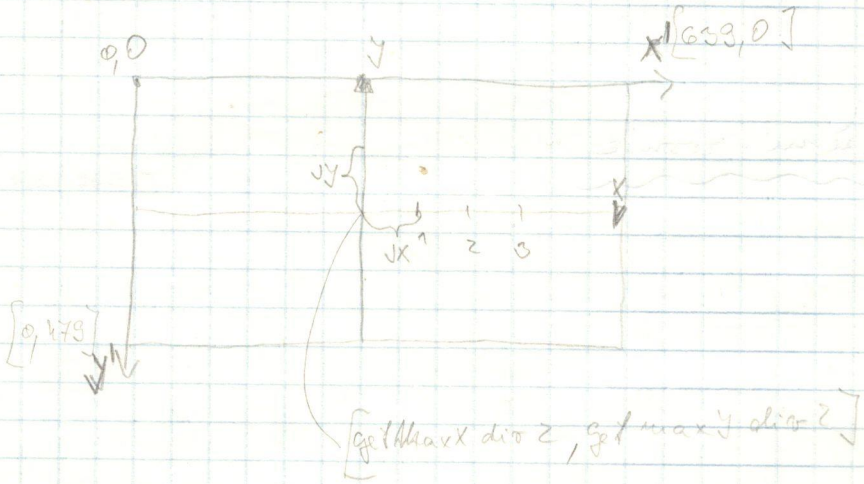
$\ln x = \sin x$ $\qquad f(x) = 0$

$\ln x - \sin x = 0$

$y = \ln x - \sin x$

# /GRAPH/

V níž vytiskl sestavit souřadné zobrazení △ABC, jestliže budou zadány souřadnice bodů A, B, C.




vx := 20
vy := 20

1. OSA X ... postupně nakreslí onu X
2. ŠIPKA X ... orientuje X
3. POPIS X ... popíše X
4. OSA Y
5. ŠIPKA Y
6. POPIS Y
7. JEDNOTKY NA X ... nakreslí i popíše
8. JEDNOTKY NA Y
9. ~~[illegible]~~ PŘEPOČTI JEDNOTKY
   XA, XB

   XAP := B div 2 + XA * vx

10. ZANESBOD

---

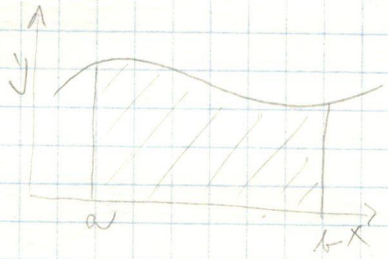
VYBARVENÍ PLOCHY

SET FILL STYLE
FLOOD FILL ( )
barva
těžiště

# NUMERICKÉ METODY VÝPOČTU URČITÝCH INTEGRÁLŮ

1) OBDÉLNÍKOVÁ
2) LICHOBĚŽNÍKOVÁ
3) SIMPSONOVO PRAVIDLO

1) Fce $y = f(x)$ je spojitá v $\langle a, b \rangle$



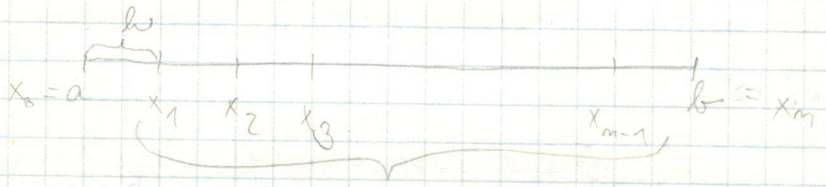

vyšr. obsah obl. vymezené fcí v $\langle a, b \rangle$

$\int_a^b f(x)\,dx$ .... číslo určitý integrál

$\left( \int f(x)\,dx \text{ ... fce neurčitý int.} \right)$

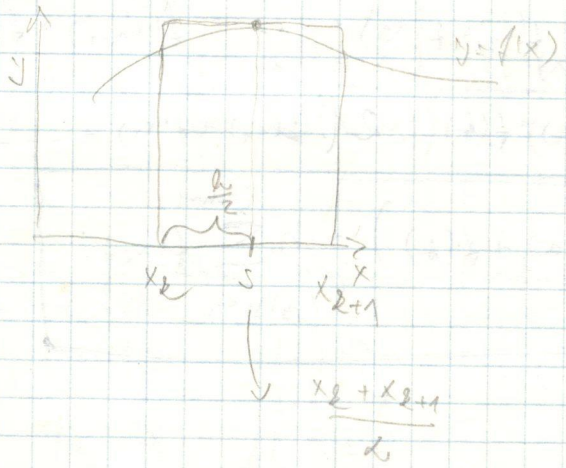
$\langle a, b \rangle$ rozdělíme na $n$ stejných částí (ekvidistantní dělení)

$h = \frac{b-a}{n}$




$x_k = x_0 + kh \qquad k = 0, 1, \dots n$

ZÁKLADNÍ VZOREC

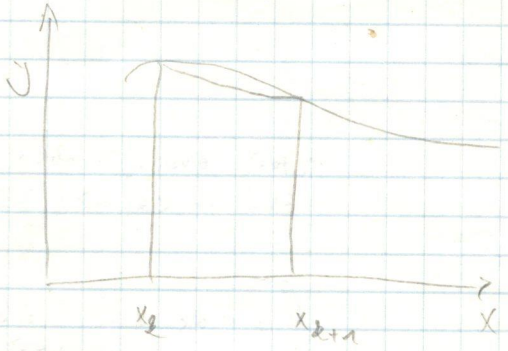



$$S = \boxed{h \cdot f\left(\frac{x_k + x_{k+1}}{2}\right) \approx \int_k^{k+1} f(x)\,dx}$$

ZOBECNĚNÍ VZORCE

$$\int_a^b f(x)\,dx = \sum_{i=0}^{i=m-1} h \cdot f\left(\frac{x_{i+1}+x_i}{2}\right)$$

2) ZÁKL. VZOREC




$$\boxed{S = \frac{(f(x_k)+f(x_{k+1}))\cdot h}{2} \approx \int_{x_k}^{x_{k+1}} f(x)\,dx}$$

ZOBECNĚNÝ VZ.

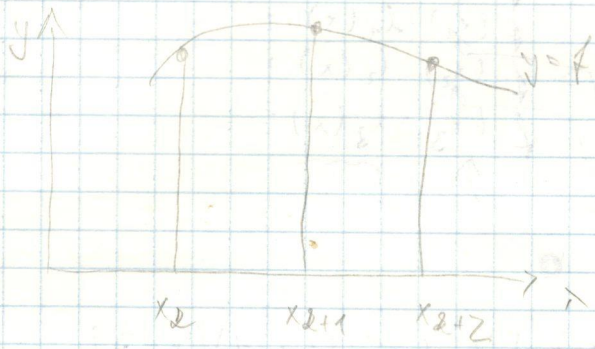
$$\int_a^b f(x)\,dx \approx \sum_{i=0}^{i=m-1} h \cdot f\left(\frac{f(x_i)+f(x_{i+1})}{2}\right) = \frac{h}{2}\Big[f(x_0)+f(x_1)+f(x_1)+f(x_2)$$

$$+ f(x_2)+f(x_3)+\dots+f(x_{m-1})+f(x_{m-1})+f(x_m)\Big] =$$

$$= \frac{h}{2}\cdot\Big[f(a)+f(b)+2\big(f(x_1)+f(x_2)+\dots f(x_{m-1})\Big] =$$

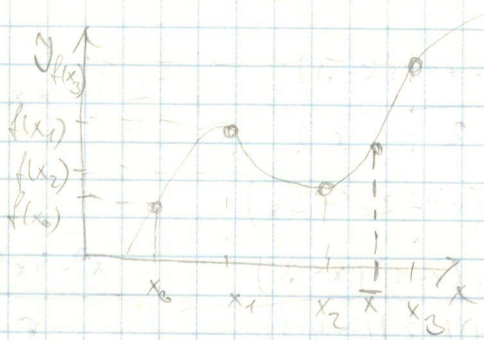
$$\boxed{= \frac{h}{2}\big(f(a)+f(b)\big) + h\sum_{i=1}^{i=m-1} f(x_i)}$$

3) SIMP. MET.




$y = f(x)$ .... nahradit kvad. fcí

INTERPOLACE FCE




$\varphi(x)$ ... polynom

$f(\bar{x}) \doteq \varphi(\bar{x})$

$f(x_0) = \varphi(x_0)$
$f(x_1) = \varphi(x_1)$
$f(x_2) = \varphi(x_2)$
$f(x_i) = \varphi(x_i)$
$i = 0, 1 \dots n$

$x_0, x_1 \dots x_n$ ... $n+1$ uzlů interpolace

? $\varphi(x)$

LAGRANGEŮV INTERPOLAČNÍ POLYNOM

$$L_n(x) = \sum_{i=0}^{n} f(x_i) \cdot l_i(x)$$

$x - x_i$ ! NE

$$l_i(x) = \frac{(x-x_0)(x-x_1)\cdot\ldots\cdot(x-x_{i-1})(x-x_{i+1})\cdot\ldots\cdot(x-x_n)}{(x_i-x_0)(x_i-x_1)\cdot\ldots\cdot(x_i-x_{i-1})(x_i-x_{i+1})\cdot\ldots\cdot(x_i-x_n)}$$

$x_i - x_i$ ! NE

nechť je fce $f$ dána svými hodnotami v bodech

| | |
|---|---|
| $x_0 = 0$ | $f(0) = 1$ |
| $x_1 = 1$ | $f(1) = 2$ |
| $x_2 = -1$ | $f(-1) = 2$ |
| $x_3 = 3$ | $f(3) = 0$ |

vyj. př. hodn. $f(2)$ podle Lagr. pol.

$n = x_3 = 3$

$$L_3(x) = \sum_{i=0}^{3} f(x_i) \cdot l_i(x) = \overset{1}{f(x_0)} \cdot l_0(x) + f(x_1) \cdot l_1(x) + \overset{2}{f(x_2)} \cdot l_2(x) + \overset{2}{f(x_3)} \cdot \underset{0}{l_3(x)}$$

$$l_0(x) = \frac{(x-x_1)(x-x_2)(x-x_3)}{(x_0-x_1)(x_0-x_2)(x_0-x_3)} = \frac{(x-1)(x+1)(x-3)}{(-1)(1)(-3)} = \frac{1}{3}(x^2-1)(x-3) =$$

$$= \frac{1}{3}(x^3-3x^2-x+3)$$

$$l_1(x) = \frac{(x-x_0)(x-x_2)(x-x_3)}{(x_1-x_0)(x_1-x_2)(x_1-x_3)} = \frac{x(x+1)(x-3)}{1 \cdot 2 \cdot (-2)} = \frac{-1}{4}(x^2+x)(x-3) =$$

$$= -\frac{1}{4}(x^3+x^2-3x^2-3x) = -\frac{1}{4}(x^3-2x^2-3x)$$

$$l_2(x) = \frac{(x-x_0)(x-x_1)(x-x_3)}{(x_2-x_0)(x_2-x_1)(x_2-x_3)} = \frac{x \cdot (x-1)(x-3)}{(-1)(-2)(-4)} = \frac{x^3-4x^2+3x}{-8}$$

$$L_3(x) = -\frac{5}{12}x^3 + x^2 + \frac{7}{12}x + 1$$

$L_3(x_i) = f(x_i)$

$L_3(0) = 1 = f(0)$

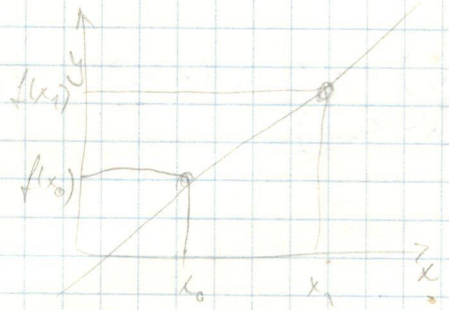
$L_3(1) = 2 = f(1)$

$\vdots$

$L_3(3) = 0 = f(3)$

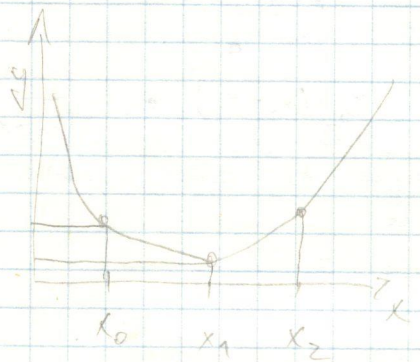
$f(2) \approx L_3(2) =$

LINEÁRNÍ LAGRANGEOVA INTERPOLACE




$$L_1(x) = \sum_{i=0}^{1} f(x_i) \cdot l_i(x) = f(x_0)\,\frac{x - x_1}{x_0 - x_1} + f(x_1)\,\frac{x - x_0}{x_1 - x_0}$$

KVADRATICKÁ L. I.




$$L_2(x) = \sum_{i=0}^{2} f(x_i) \cdot l_i(x) = f(x_0)\,\frac{(x-x_1)(x-x_2)}{(x_0-x_1)(x_0-x_2)} +$$

$$+ f(x_1)\,\frac{(x-x_0)(x-x_2)}{(x_1-x_0)(x_1-x_2)} +$$

$$+ f(x_2)\,\frac{(x-x_0)(x-x_1)}{(x_2-x_0)(x_2-x_1)}$$

# NEWTONŮV INTERPOLAČNÍ POLYNOM

$$N_n(x) = a_0 + a_1(x-x_0) + a_2(x-x_0)(x-x_1) + a_3(x-x_0)(x-x_1)(x-x_2) + \dots$$
$$\dots + a_n(x-x_0)(x-x_1)\cdot\dots\cdot(x-x_{n-1})$$

známe: $x_0, x_1 \dots x_n$

$f(x_0) \dots f(x_n)$

$N_n(x_i) = f(x_i)$

↳ $i = 0, 1 \dots n$

určit $N_n(x)$ znamená najít hodnoty ~~ale~~ koeficientů

$a_0, a_1 \dots a_n$

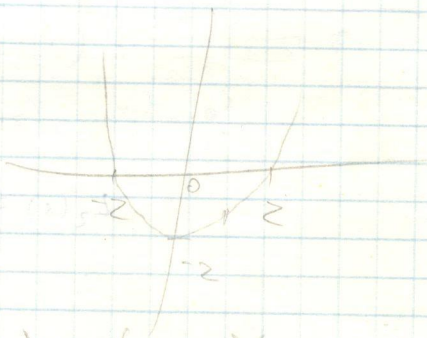
$N_n(x_0) = f(x_0) = a_0$

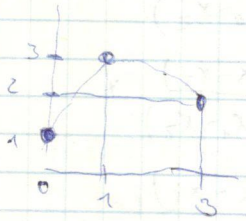
$N_n(x_1) = f(x_1) = a_0 + a_1(x_1 - x_0)$ $\quad a_1 = \dfrac{f(x_1) - a_0}{x_1 - x_0}$

$N_n(x_2) = f(x_2) = a_0 + a_1(x_2 - x_0) + a_2(x_2 - x_0)(x_2 - x_1)$ $\quad a_2 = \dfrac{f(x_2) - a_0 - a_1(x_2 - x_0)}{(x_2 - x_0)(x_2 - x_1)}$

Použití algoritmu přímé substituce

? $L_2(x)$ $\qquad N_2(x)$

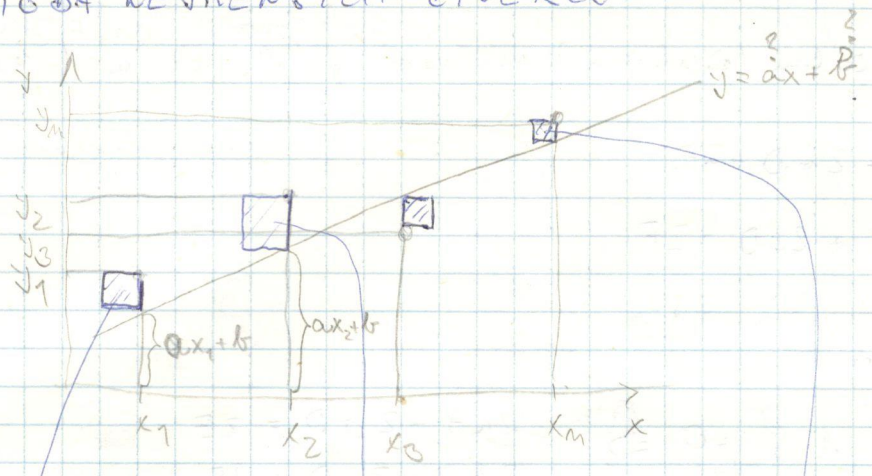



$$N_n(x_3) = f(x_3) = a_0 + a_1(x_3 - x_0) + a_2(x_3 - x_0)(x_3 - x_1)$$
$$+ a_3(x_3 - x_0)(x_3 - x_1)(x_3 - x_2)$$

$$a_3 = \frac{f(x_3) - a_0 - a_1(x_3 - x_0) - a_2(x_3 - x_0)(x_3 - x_1)}{(x_3 - x_0)(x_3 - x_1)(x_3 - x_2)}$$

# METODA NEJMENŠÍCH ČTVERCŮ




$y_1 \dots y_n$ – nejsou přesné hodnoty (empiricky získané)

$S = (y_1 - ax_1 - b)^2 = (ax_1 + b - y_1)^2$ $\quad S = (y_2 - ax_2 - b)^2$ $\quad S = (ax_n + b - y_n)^2$

Hledáme $a, b$ tak, aby <u>součet obsahů</u> všech čtverců byl co nejmenší.

$$S = (ax_1 + b - y_1)^2 + (ax_2 + b - y_2)^2 + \dots + (ax_n + b - y_n)^2 =$$

$$= \sum_{i=1}^{n} (ax_i + b - y_i)^2$$

$$= a^2x_1^2 + b^2 + y_1^2 + 2ab\,x_1 - 2ax_1y_1 - 2by_1 +$$

$$+ \dots$$

$$+ a^2x_n^2 + b^2 + y_n^2 + 2ab\,x_n - 2ax_ny_n - 2by_n$$

$$= a^2\underbrace{\boxed{(x_1^2 + \dots + x_n^2)}}_{A} + \underbrace{\boxed{n}}_{B}b^2 + \underbrace{\boxed{(y_1^2 + \dots y_n^2)}}_{C} + 2ab\underbrace{\boxed{(x_1 + \dots x_n)}}_{D} - 2a\underbrace{\boxed{(x_1y_1 + \dots + x_ny_n)}}_{E} - 2b\underbrace{\boxed{(y_1 + \dots y_n)}}_{F}$$

$$= a^2A + Bb^2 + C + 2abD - 2aE - 2bF \quad \dots \text{ fce 2 proměnných } a, b$$

$S'_{(a)}$ ... parciální (částečná) derivace fce $S$ dle $a$ ( $b$ je konstanta

$S'_{(a)} = 2aA + 2bD - 2E$

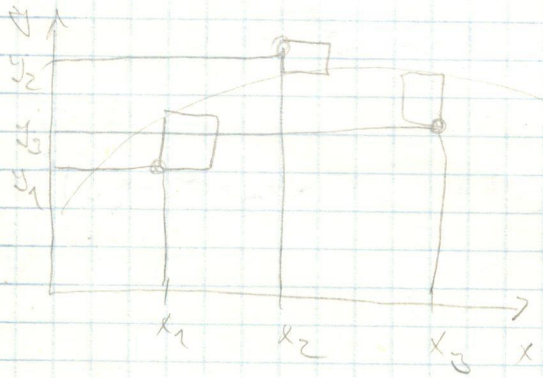
$S'_{(b)} = 2bB + 2aD - 2F$

Extrém fce:

$S'_{(a)} = 0 \wedge S'_{(b)} = 0 \Rightarrow$

$$\begin{aligned} 2aA + 2bD &= 2E \\ 2bB + 2aD &= 2F \end{aligned}$$

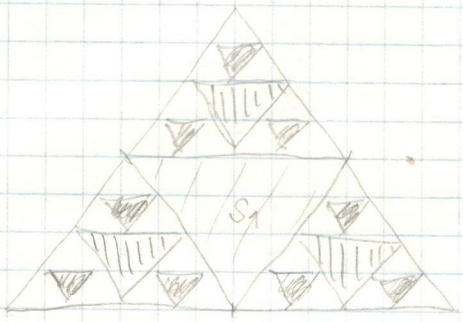
$$\begin{aligned} aA + bD &= E \\ bB + aD &= F \end{aligned} \Rightarrow \begin{aligned} aA + bD &= E \\ aD + bB &= F \end{aligned}$$




$$S = \sum_{i=1}^{n} (ax_i^2 + bx_1 + c - y_i)^2$$

# FRAKTÁLNÍ GEOMETRIE

Dán rovnostranný $\triangle ABC$, $a$




$$S_{\triangle ABC} = \frac{a^2\sqrt{3}}{4}$$

$$S_1 = \frac{a^2\sqrt{3}}{16}$$

$$S_2 = 3 \cdot \frac{\left(\frac{a}{4}\right)^2\sqrt{3}}{4} = \frac{3a^2\sqrt{3}}{64}$$ – 3 malý $\triangle$

$$S_3 = 9 \cdot \frac{\left(\frac{a}{8}\right)^2\sqrt{3}}{4} = \frac{9a^2\sqrt{3}}{2\sqrt{6}}$$ – 9 malejch $\triangle$

nekonečný proces

$$\frac{S_2}{S_1} = \frac{\frac{3a^2\sqrt{3}}{64}}{\frac{a^2\sqrt{3}}{16}} = \frac{3}{4}$$

$S_1, S_2, S_3 \dots S_n$

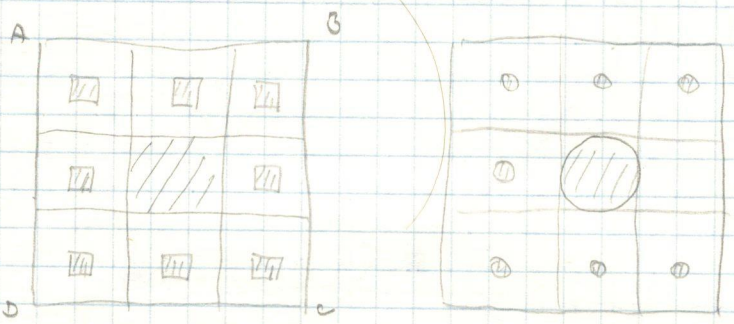
$S_2 = \frac{3}{4} S_1$

$S_3 = \frac{3}{4} S_2$ nekonečná geometrická řada

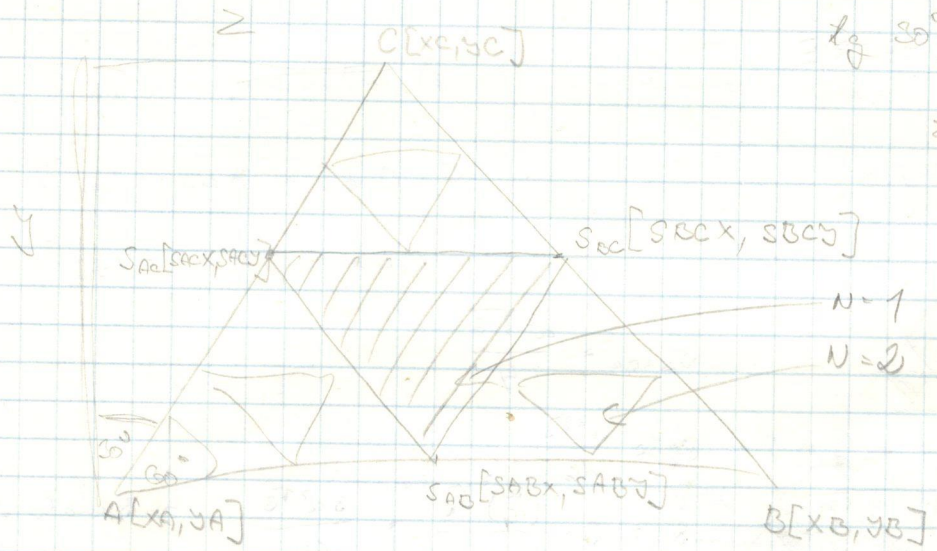
je konvergentní $\Leftrightarrow |q| < 1$

$q = \frac{3}{4}$

$$\boxed{S = \frac{a_1}{1-q}} = \frac{\frac{a^2\sqrt{3}}{16}}{\frac{1}{4}} = \frac{a^2\sqrt{3}}{4}$$




SIERPINSKÉHO KOBEREC



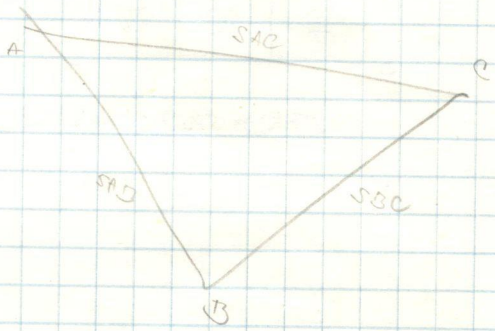

$tg\ 30° = \frac{z}{y}$

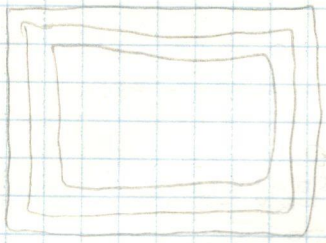
$z = y \cdot \frac{\sin 30°}{\cos 30°}$

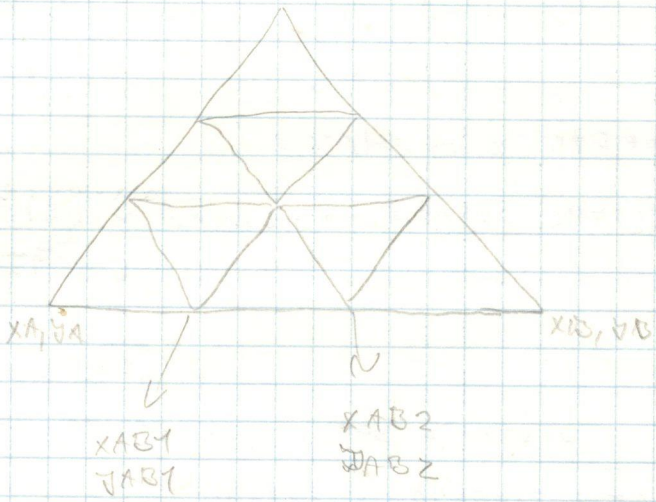
$\boxed{z\ 1D\ \text{vždy}\ 3}$

```
procedure Trojuhelnik (N:Byte; xA,yA, xB,yB, xC,yC : Word);
var   SABX , SABY
      SBCX , SBCY
      SACX , SACY : Word;
begin
      Round ((xA+xB)/2);
SABX := (xA + xB) div 2;
SABY :=
SBCX :=
SBCY :=
SACX :=
SACY :=
Set Color (      );
Move To (SABX , SABY);
LineTO (SBCX , SBCY);
Line To (SCA x , SCAY);
LineTo (SABX , SABY);
Set Fill Style
Flood Fill
Trojuhelnik ( N-1
Trojuhelnik ( N-1
Trojuhelnik ( N-1
end;
```

if N>0







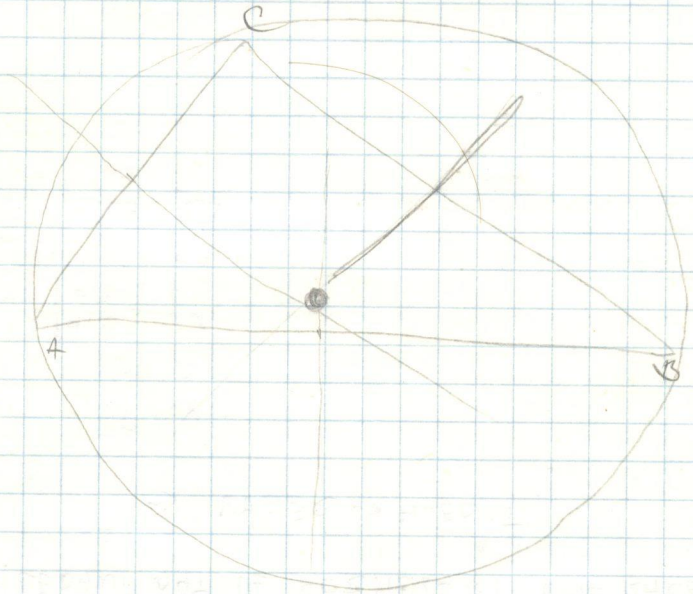

① Zobrazit v soustavě souřadné kružnici se zadaným středem a poloměrem.

② Zobr. k, vnější bod k => spojnice středu k a vnějšího bodu a nechat vytvořit vzdálenost vnějšího bodu od kružnice

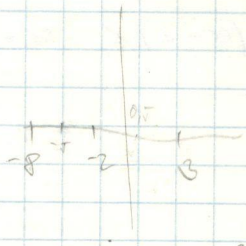
③ Dána k a vnější bod. Konstruovat k. se středem ve vnějším bodě kt. se dotýká dané k.

④ je daný △ABC v sous. souř. Narýsovat k opsanou tomto △

⑤ Vnějším bodem k vést tečny k dané k.




1 AB
2 AC
3 BC




$\vec{s_1} = (Bx - Ax, By - Ay) = \vec{n_1}$

~~$ax + by + c = 0$~~

$a_1 sx1 + b_1 sy1 + c_1 = 0$

$c_1 = -a_1 sx1 - b_1 sy1$

$c_2 = -a_2 sx2 - b_2 sy2$

$c_3 = -a_3 sx3 - b_3 sy3$

$a_1 x + b_1 y + c_1 = 0$

$a_2 x + b_2 y + c_2 = 0$ stačí 2

$a_3 x + b_3 y + c_3 = 0$

4. TVORBA FORMULÁŘŮ (PROJEKCE) - VÝBĚR DAT

NOVÝ FORMULÁŘ

5. ÚPRAVA FORULÁŘE -> VÝBĚR -> FORMULÁŘ } PŘEPÍNÁNÍ
LISTOVAT

6. VYHLEDÁNÍ DAT VE FORMULÁŘI (FILTRACE)

-> VÝBĚR -> HLEDAT