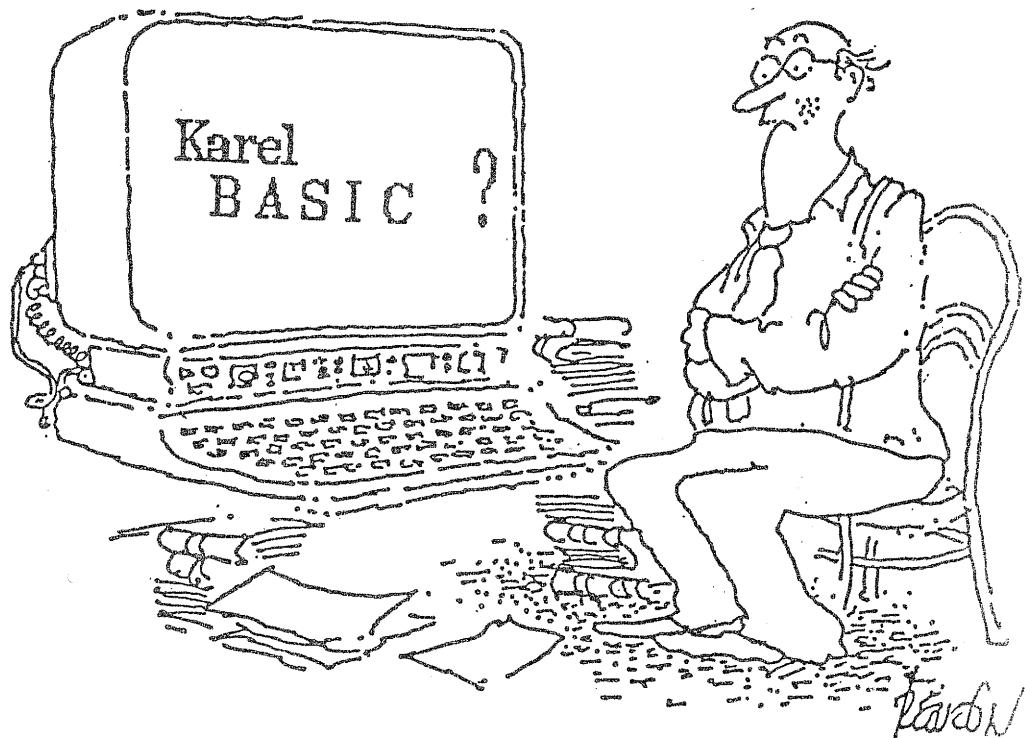


Soutěž dětí a mládeže v programování
Přebor České republiky
BRNO 18.- 20. května 1990



Soubor řešených úloh a teoretických testů

**Soutěž dětí a mládeže v programování
Přebor České republiky
BRNO 18. - 20. května 1990**

Karel - mladší žáci

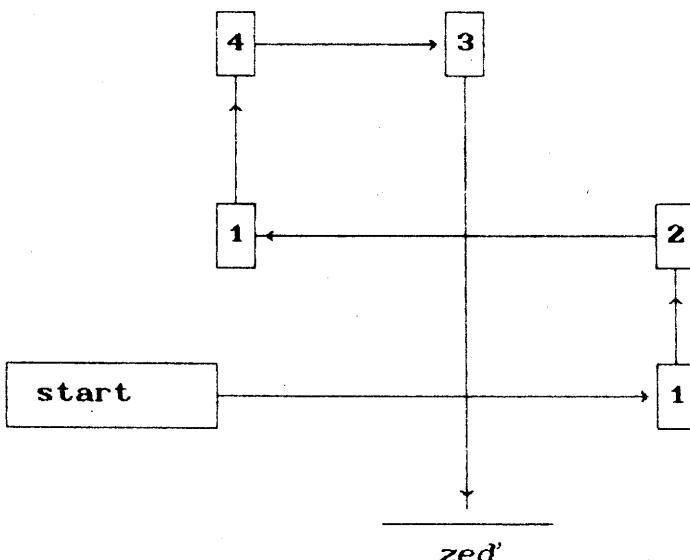
Příklad 1

Koeficient obtížnosti: 1

Karel stojí na startu a je nastaven ve směru trasy. Po spuštění trasu projde. Trasu tvoří body, mezi nimiž jde Karel rovně. Body jsou značeny značkami, na nichž Karel mění směr:

-
- 1..... sever
2..... západ
3..... jih
4..... východ
-

Karel skončí, narazí-li na zed' bez značky.



Soutěž dětí a mládeže v programování
Přebor České republiky
BRNO 18.- 20. května 1990

R E S E N I

Karel - mladší žáci

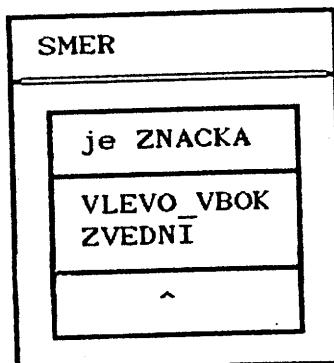
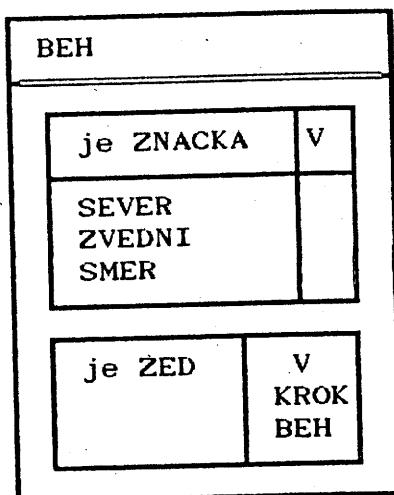
Příklad 1

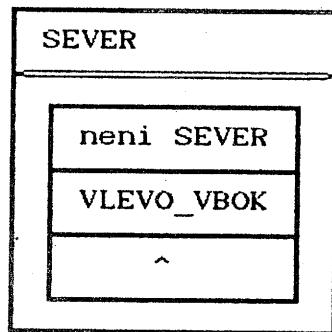
Koefficient obtížnosti: 1

BEH znamena
kdyz je ZNACKA
 SEVER
 ZVEDNI
 SMER
 konec, jinak
 konec
 kdyz je ZED
 konec, jinak
 KROK
 BEH
 konec
 konec

SMER znamena
dokud je ZNACKA
 VLEVO_VBOK
 ZVEDNI
 konec
konec

SEVER znamena
dokud není SEVER
 VLEVO_VBOK
 konec
konec





**Soutěž dětí a mládeže v programování
Přebor České republiky
BRNO 18.- 20. května 1990**

Karel - mladší žáci

Příklad 2

Koeficient obtížnosti: 1,5

Vyplň značkami obdélník, u jehož JZ vrcholu je Karel a jehož SV vrchol je SV vrchol města. Město je prázdné.

Příklad 3

Koeficient obtížnosti: 2

Karel v prázdném městě pod sebe položí kolik značek, kolik kroků ho dělí od zdi, kníž je natočen (je to méně než 6 kroků).

Příklad 4

Koeficient obtížnosti: 3

U jižní a západní zdi je umístěna značka. Karel dojde na políčko, v němž se kříží řádek a sloupec obsahující značku.

**Soutěž dětí a mládeže v programování
Přebor České republiky
BRNO 18. - 20. května 1990**

ŘEŠENÍ

Karel - mladší žáci

Příklad 2

Koefficient obtížnosti: 1,5

OBDELNIK znamena

KSEVERU

RADAАЗPET

dokud není ZED

KROK

RADAАЗPET

konec

konec

KSEVERU znamena

dokud není SEVER

VLEVO_VBOK

konec

konec

RADAАЗPET znamena

VPRAVOVBOK

RADAKEZDI

CELEMVZAD

POZNACZPET

VPRAVOVBOK

konec

CELEMVZAD znamena

VLEVO_VBOK

VLEVO_VBOK

konec

VPRAVOVBOK znamena

opakuj 3 krát

VLEVO_VBOK

konec

konec

RADAKEZDI znamena

dokud není ZED

POLOZ

KROK

konec

POLOZ

konec

POZNACZPET znamena

dokud je ZNACKA

KROK

konec

CELEMVZAD

KROK

CELEMVZAD

konec

OBDELNIK

KSEVERU

RADAАЗPET

není ZED

KROK

RADAАЗPET

^

KSEVERU

neni SEVER

VLEVO_VBOK

^

RADAKEZDI

neni ZED

POLOZ
KROK

^

POLOZ

RADA AZPET

VPRAVOVBOK
RADAKEZDI
CELEMVZAD
POZNACZPET
VPRAVOVBOK

CELEMVZAD

VLEVO_VBOK
VLEVO_VBOK

VPRAVOVBOK

3 krat

VLEVO_VBOK

^

POZNACZPET

je ZNACKA

KROK

^

CELEMVZAD
KROK
CELEMVZAD

**Soutěž dětí a mládeže v programování
Přebor České republiky
BRNO 18.- 20. května 1990**

ŘEŠENÍ

Karel - mladší žáci

Příklad 3

MERIC znamena

POLOZ

KEZDI

CELEMVZAD

SPOCITEJ

ZVEDNI

konec

CELEMVZAD znamena

VLEVO_VBOK

VLEVO_VBOK

konec

SPOCITEJ znamena

kdyz je ZNACKA

konec, jinak

KROK

SPOCITEJ

POLOZ

konec

konec

KEZDI znamena

dokud není ZED

KROK

konec

konec

Koefficient obtížnosti: 2

MERIC

POLOZ

KEZDI

CELEMVZAD

SPOCITEJ

ZVEDNI

CELEMVZAD

VLEVO_VBOK

VLEVO_VBOK

KEZDI

není ZED

KROK

~

SPOCITEJ

je ZNACKA

V

KROK
SPOCITEJ
POLOZ

**Soutěž dětí a mládeže v programování
Přebor České republiky
BRNO 18.- 20. května 1990**

ŘEŠENÍ

Karel - mladší žáci

Příklad 4

Koeficient obtížnosti: 3

KRIZ znamena

DOMU
KKRIZ
ZVEDNI
ZKROK

konec

KKRIZ znamena

ZKROK
VLEVO_VBOK
opakuj 9 krat
POLOZ
KROK

konec

• VLEVO_VBOK

KEZDI

VLEVO_VBOK

opakuj 9 krat

 kdyz je ZNACKA

 konec, jinak

 KROK

 konec

 konec

 VLEVO_VBOK

konec

ZKROK znamena

dokud není ZNACKA

 KROK

 konec

 konec

DOMU znamena

NAZAPAD
KEZDI
VLEVO_VBOK
KEZDI

 VLEVO_VBOK
konec

CELEMVZAD znamena

 VLEVO_VBOK
 VLEVO_VBOK

 konec

KEZDI znamena

 dokud není ZED
 KROK

 konec

 konec

NAZAPAD znamena

 dokud není ZAPAD
 VLEVO_VBOK

 konec

 konec

KKRIZ

ZKROK
VLEVO_VBOK

9 krat

POLOZ
KROK

^

VLEVO_VBOK
KEZDI
VLEVO_VBOK

9 krat

je ZNACKA

v

KROK

^

VLEVO_VBOK

ZKROK

neni ZNACKA

KROK

^

KRIZ

DOMU
KKRIZ
ZVEDNI
ZKROK

DOMU

NAZAPAD
KEZDI
VLEVO_VBOK
KEZDI
VLEVO_VBOK

CELEMVZAD

VLEVO_VBOK
VLEVO_VBOK

KEZDI

neni ZED

KROK

^

NAZAPAD

neni ZAPAD

VLEVO_VBOK

^

**Soutěž dětí a mládeže v programování
Přebor České republiky
BRNO 18.- 20. května 1990**

Karel - starší žáci

Příklad 1

Koeficient obtížnosti: 1

Na řádce, na jejímž začátku stojí Karel, jsou značky. Změň jejich počet na každém políčku na 5-z, kde z je původní počet značek na políčku.

Příklad 2

Koeficient obtížnosti: 2

Na druhém a třetím řádku od spodního okraje jsou binární čísla o maximální délce 9 bitů. Karel je sečeť a výsledek uloží do prvního řádku u spodního okraje.

Příklad 3

Koeficient obtížnosti: 2,5

Od horního okraje města visí několik krápníků (maximálně 5) šířky (zn.). Karel je spočítá a výsledek uloží do JZ rohu města. (Políčko, ve kterém je část krápníku, obsahuje maximálně 1 značku).

Příklad 4

Koeficient obtížnosti: 3

Na spodní hraně města jsou dvě políčka se značkami. Karel zjistí, zda je na obou políčkách stejný počet značek. Výsledek oznámí svým otočením: na sever - stejný počet jiným směrem - různý počet

**Soutěž dětí a mládeže v programování
Přebor České republiky
BRNO 18.- 20. května 1990**

ŘEŠENÍ

Karel - starší žáci

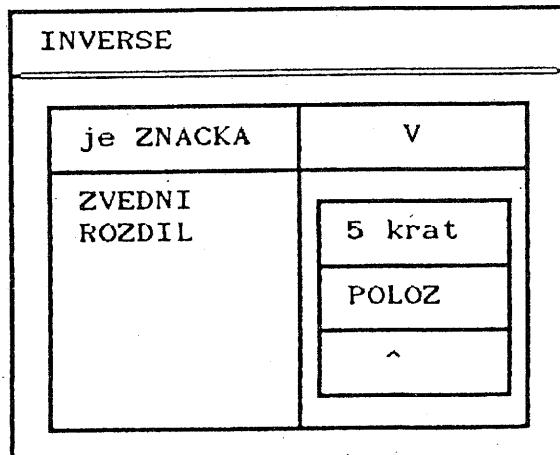
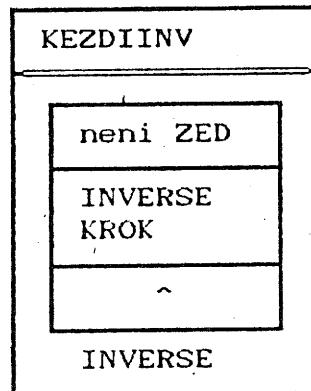
Příklad 1

Koeficient obtížnosti: 1

KEZDIINV znamena
dokud není ZED
 INVERSE
 KROK
konec
 INVERSE
konec

INVERSE znamena
když je ZNACKA
 ZVEDNI
 ROZDIL
konec, jinak
 opakuj 5 krát
 POLOZ
 konec
konec
konec

ROZDIL znamena
když je ZNACKA
 ZVEDNI
 ROZDIL
konec, jinak
 opakuj 5 krát
 POLOZ
 konec
konec
 ZVEDNI
konec



ROZDIL

je ZNACKA	V			
ZVEDNI ROZDIL	<table border="1"><tr><td>5 krat</td></tr><tr><td>POLOZ</td></tr><tr><td>~</td></tr></table>	5 krat	POLOZ	~
5 krat				
POLOZ				
~				

**Soutěž dětí a mládeže v programování
Přebor České republiky
BRNO 18. - 20. května 1990**

R E S E N I

Karel - starší žáci

Příklad 2

Koefficient obtížnosti: 2

PLUS znamena

JV

opakuj 9 krát

TOP

LOW

CARRY

konec

konec

CARRY znamena

ZVEDNI

kdyz je ZNACKA

ZVEDNI

KROK

POLOZ

konec, jinak

POLOZ

KROK

konec

VPRAVO

konec

LOW znamena

KROK

CE

kdyz je ZNACKA

ZVEDNI

KROK

POLOZ

CE

LOW

konec, jinak

KROK

VPRAVO

konec

konec

TOP znamena

KROK

KROK

CE

kdyz je ZNACKA

ZVEDNI

KROK

KROK

POLOZ

CE

TOP

konec, jinak

KROK

KROK

CE

konec

konec

JV znamena

dokud není VYCHOD

VLEVO_VBOK

konec

dokud není ZED

KROK

konec

VPRAVO

dokud není ZED

KROK

konec

CE

konec

CE znamena
VLEVO_VBOK
VLEVO_VBOK
konec

VPRAVO znamena
VLEVO_VBOK
VLEVO_VBOK
VLEVO_VBOK
konec

VPRAVO

VLEVO_VBOK
VLEVO_VBOK
VLEVO_VBOK

CE

VLEVO_VBOK
VLEVO_VBOK

PLUS

JV

9 krat

TOP
LOW
CARRY

^

CARRY

ZVEDNI

je ZNACKA	V
ZVEDNI KROK POLOZ	POLOZ KROK

VPRAVO

JV

neni VYCHOD

VLEVO_VBOK

^

neni ZED

KROK

^

VPRAVO

neni ZED

KROK

^

CE

LOW

KROK
CE

je ZNACKA	V
ZVEDNI KROK POLOZ CE LOW	KROK VPRAVO

TOP

KROK

KROK

CE

je ZNACKA	V
ZVEDNI KROK KROK POLOZ CE TOP	KROK KROK CE

**Soutěž dětí a mládeže v programování
Přebor České republiky
BRNO 18.- 20. května 1990**

ŘEŠENÍ

Karel - starší žáci

Příklad 3

Koefficient obtížnosti: 2,5

KRAPNIK znamena

JZ

dokud je ZNACKA

ZVEDNI

konec

JEDEN

konec

JEDEN znamena

dokud není ZED

KROK

konec

VPRAVO

když je ZNACKA

ZVEDNI

JZ

POLOZ

JEDEN

konec, jinak

když je ZED

konec, jinak

KROK

VLEVO_VBOK

JEDEN

konec

konec

konec

JZ znamena

dokud není JIH

VLEVO_VBOK

konec

dokud není ZED

KROK

konec

VPRAVO

dokud není ZED

KROK

koněc

VPRAVO

konec

VPRAVO znamena

VLEVO_VBOK

VLEVO_VBOK

VLEVO_VBOK

konec

KRAPNIK

JZ

je ZNACKA

ZVEDNI

^

JEDEN

VPRAVO

VLEVO_VBOK

VLEVO_VBOK

VLEVO_VBOK

JZ

neni JIH

VLEVO_VBOK

^

neni ZED

KROK

^

VPRAVO

neni ZED

KROK

^

JEDEN

neni ZED

KROK

^

VPRAVO

je ZNACKA

V

ZVEDNI
JZ
POLOZ
JEDEN

je ZED

V

KROK
VLEVO_VBOK
JEDEN

**Soutěž dětí a mládeže v programování
Přebor České republiky
BRNO 18.- 20. května 1990**

R E Š E N I

Karel - starší žáci

Příklad 4

Koefficient obtížnosti: 3

POROVNEJ znamena

ZACATEK

TEST

konec

TEST znamena

DRUHA

ZVEDNI

kdyz je ZNACKA

DRUHA

ZVEDNI

kdyz je ZNACKA

TEST

konec, jinak

konec

konec, jinak

DRUHA

ZVEDNI

kdyz je ZNACKA

konec, jinak

dokud není SEVER

VLEVO_VBOK

konec

konec

konec

DRUHA znamena

KROK

dokud není ZNACKA

KROK

konec

CELEM

konec

ZACATEK znamena

dokud není ZAPAD

VLEVO_VBOK

konec

dokud není ZED

KROK

konec

VLEVO_VBOK

dokud není ZED

KROK

konec

VLEVO_VBOK

dokud není ZNACKA

KROK

konec

konec

CELEM znamena

VLEVO_VBOK

VLEVO_VBOK

konec

POROVNEJ

ZACATEK

TEST

CELEM

VLEVO_VBOK

VLEVO_VBOK

TEST

DRUHA
ZVEDNI

je ZNACKA	V
DRUHA ZVEDNI	DRUHA ZVEDNI
je ZNACKA V TEST	je ZNACKA V neni SEVER VLEVO_VBOK ^

DRUHA

KROK

neni ZNACKA

KROK

^

CELEM

ZACATEK

neni ZAPAD

VLEVO_VBOK

^

neni ZED

KROK

^

VLEVO_VBOK

neni ZED

KROK

^

VLEVO_VBOK

neni ZNACKA

KROK

^

**Soutěž dětí a mládeže v programování
Přebor České republiky
BRNO 18.- 20. května 1990**

BASIC - starší žáci

Příklad 1

Koefficient obtížnosti: 1

Je dán textový řetězec. Vytvořte program, který vypíše sloupcový graf četnosti výskytu písmen v řetězci. Přitom nerozlišujte malá písmena od velkých. (V řetězci se nevyskytují písmena s háčky a čárkami.)

Příklad 2

Koefficient obtížnosti: 1

Jsou dána dvě kladná desítková čísla, která mají maximálně 30 platných míst. Napište program, který tato čísla sečte a výsledek vypíše. Předpokládá se, že čísla budou zadána jako řetězce znaků.

Příklad 3

Koefficient obtížnosti: 2

Vytvořte program, který vypíše všechna prvočísla ležící v intervalu od 1 do 1000 včetně.

Příklad 4

Koefficient obtížnosti: 2

Napište program, který zjistí, z kterých prostorových pyramid o výškách 1 až 100, lze sestavit plošnou pyramidu tak, aby nezbyla žádná kostka.

Soutěž dětí a mládeže v programování
Přebor České republiky
BRNO 18.- 20. května 1990

ŘEŠENÍ

BASIC - starší žáci

Příklad 1

Koefficient obtížnosti 1

```
10 DIM G[256]
20 INPUT S$
30 FOR I=1 TO LEN(S$)
40 C$=MID$(S$,I,1)
50 IF (ASC(C$)<ASC("a")) OR
(ASC(C$)>ASC("z")) GOTO 70
60 C$=CHR$(ASC(C$)-ASC("a")+ASC("A"))
70 IF (ASC(C$)<ASC("A")) OR
(ASC(C$)>ASC("Z")) GOTO 90
80 G(ASC(C$)-ASC("A")+1)=G(ASC(C$)-ASC("A")+1)+1
90 NEXT I
100 M=G(1)
110 FOR I=1 TO 26
120 IF G(I)<=M GOTO 140
130 M=G(I)
140 NEXT I
150 IF M>0 GOTO 180
160 PRINT "NENI ZADNE PISMENO"
170 END
180 FOR J=10 TO 1 STEP -1
190 FOR I=1 TO 26
200 IF 10*G(I)/M>=J GOTO 230
210 PRINT " ";
220 GOTO 240
230 PRINT "*";
240 NEXT I
250 PRINT
260 NEXT J
270 FOR I=1 TO 26
280 PRINT CHR$(I+ASC("A")-1);
290 NEXT I
300 PRINT
900 END
```

**Soutěž dětí a mládeže v programování
Přebor České republiky
BRNO 18.- 20. května 1990**

R E S E N I
BASIC - starší žáci

Příklad 2

Koefficient obtížnosti 1

```
10 INPUT "N1:",N1$  
20 INPUT "N2:",N2$  
30 C=0  
40 V$=""  
50 IF LEN(N1$)>LEN(N2$) GOTO 120  
60 L=LEN(N2$)  
70 N$=N1$  
80 GOSUB 1000  
90 N1$=N$  
100 GOTO 180  
120 L=LEN(N1$)  
130 N$=N2$  
140 GOSUB 1000  
150 N2$=N$  
180 FOR I=1 TO L  
190 P=ASC(MID$(N1$,L-I+1,1))-ASC("0")  
200 P=P+ASC(MID$(N2$,L-I+1,1))-ASC("0")+C  
210 C=INT(P/10)  
220 V$=CHR$(ASC("0")+P-10*C)+V$  
230 NEXT I  
240 IF C=0 GOTO 300  
250 V$="1"+V$  
300 PRINT V$  
900 END  
1000 IF LEN(N$)=L GOTO 1100  
1010 FOR I=LEN(N$)+1 TO L  
1020 N$="0"+N$  
1030 NEXT I  
1100 RETURN
```

**Soutěž dětí a mládeže v programování
Přebor České republiky
BRNO 18. - 20. května 1990**

ŘEŠENÍ

BASIC - starší žáci

Příklad 3

Koefficient obtížnosti 2

```
10 DIM P[1000]
20 J=2
30 I=J
40 IF 2*I>1000 GOTO 120
50 FOR I=2*I TO 1000 STEP I
60 P(I)=1
70 NEXT I
80 J=J+1
90 IF J>1000 GOTO 120
100 IF P(J)=1 GOTO 80
110 GOTO 30
120 FOR I=1 TO 1000
130 IF P(I)=1 GOTO 150
140 PRINT I;" ";
150 NEXT I
160 PRINT
170 END
```

**Soutěž dětí a mládeže v programování
Přebor České republiky
BRNO 18.- 20. května 1990**

ŘEŠENÍ

BASIC - starší žáci

Příklad 4

Koefficient obtížnosti 2

```
10 DIM A(100)
20 S=0
30 FOR I=1 TO 100
40 S=S+I*I
50 A(I)=S
60 NEXT I
70 I=1
80 S=0
90 L=1
100 S=S+I
110 I=I+1
120 IF A(L)>S THEN 100
130 IF A(L)=S THEN PRINT "Jde to pro vysku";L
140 L=L+1
150 IF L<=100 THEN 100
160 PRINT "To je všechno"
170 END
```

**Soutěž dětí a mládeže v programování
Přebor České republiky
BRNO 18. - 20. května 1990**

BASIC - mládež

Příklad 1

Koeficient obtížnosti: 1

Napište program, který vypíše všechny prvočíselné dvojice, které leží v intervalu od 1000 do 2000. Prvočíselná dvojice čísel, která se od sebe liší o 2 a jsou obě prvočíslly (například 3 a 5, 11 a 13 atd.)

Příklad 2

Koeficient obtížnosti: 2

*Napište program, který po zadání roku (1900-2100) vypíše všechny pátky třináctého v tomto roce.
Pozn. 1.1.1900 bylo pondělí.*

Příklad 3

Koeficient obtížnosti: 2

Dominové kostky jsou na každém ze svých dvou konců označeny číslicí od 1 do 6 (včetně). Sestavte program, který zjistí, zdali lze najít takové uspořádání daných kostek, aby jednotlivé kostky spolu sousedily konci označenými stejnou číslicí. Kostek může být maximálně šest. Kostky budou zadány jako dvojice čísel.

**Soutěž dětí a mládeže v programování
Přebor České republiky
BRNO 18.- 20. května 1990**

BASIC - mládež

Příklad 4

Koeficient obtížnosti 3

*V příkazech DATA je napsán program v programova-
cím jazyce KONVALINKA. Napište program, který jej
vykoná.*

Příkazy jazyka KONVALINKA jsou:

ECHO *Výstup proměnné, nebo textového řetězce
na obrazovku*

ECHO I1

ECHO 'Retezec'

INPUT.... *Vstup čísla z klávesnice*

INPUT I5

LET..... *Přiřazení hodnoty proměnné. Ve výrazu
se smí vyskytovat: celočíselné konstan-
ty, proměnné a operátory + a -.*

LET I1 = 23

LET I1 = -4

LET I1 = I5

LET I1 = I6+1

LET I1 = I0-7

LET I1 = I2+I1

LET I1 = I4-I8

DO..... *Příkaz cyklu. Je obdobou příkazu FOR
v BASICu.*

DO I2 = 16(*význam jako FOR I2 = 1 TO
16 STEP 1*)

DO I1 = I5(*význam jako FOR I1 = 1 TO
I5 STEP 1*)

ENDDO.... *Příkaz konce cyklu. Je obdobou příkazu
NEXT v BASICu.*

ENDDO

END..... *Příkaz ukončení programu. Musí být uždy
nakonci programu. Nesmí být uveden více-
krát.*

END

- Pozn.*
- Počet vnořených cyklů DO je omezen na 50, maximální počet řádků v programu je 100
 - Jazyk KONVALINKA dovoluje pracovat pouze s deseti celočíselnými proměnnými, jejichž jména jsou I0 až I9.
 - Na jednom řádku smí být pouze jeden příkaz
 - Řádek smí obsahovat nadbytečné mezery, ale nikoli u klíčových slovech.

**Soutěž dětí a mládeže v programování
Přebor České republiky
BRNO 18.- 20. května 1990**

ŘEŠENÍ

BASIC - mládež

Příklad 1

Koeficient obtížnosti 1

```
10 DIM P[2000]
20 J= 2
30 I=J
40 IF 2*I>2000 GOTO 120
50 FOR I=2*I TO 2000 STEP I
60 P(I)=1
70 NEXT I
80 J=J+1
90 IF J>2000 GOTO 120
100 IF P(J)=1 GOTO 80
110 GOTO 30
120 FOR I=1002 TO 2000
130 IF (P(I)=1) OR (P(I-2)=1) GOTO 150
140 PRINT I-2;",";I;";"
150 NEXT I
160 PRINT
170 END
```

**Soutěž dětí a mládeže v programování
Přebor České republiky
BRNO 18.- 20. května 1990**

ŘEŠENÍ

BASIC - mládež

Příklad 2

Koeficient obtížnosti 2

```
10 DIM L(12)
20 FOR I=1 TO 12
30 READ L(I)
40 NEXT I
50 GOSUB 340
60 D=1
70 M=1
80 R=1900
90 T=1
100 INPUT "Rok: ",RR
110 IF R=RR THEN 170
120 GOSUB 360
130 R=R+1
140 T=T+LD
150 IF T>7 THEN T=T-7
160 GOTO 110
170 IF T<6 THEN D=6-D else D=13-T
180 T=5
190 GOSUB 250
200 IF R>RR THEN 230
210 IF D=13 THEN ??"Patek 13.";M;";";RR
220 GOTO 190
230 ??"To je všechno"
240 END
250 D=D+7
260 IF D<L(M) THEN RETURN
270 D=D-L(M)
280 M=M+1
290 IF M<13 THEN RETURN
300 M=1
```

```
310 R=R+1
320 GOSUB 340
330 RETURN
340 IF (R/4=INT(R/4)) AND (R/100<>INT(R/100))
    THEN L(2)=29 ELSE L(2)=28
350 RETURN
360 IF (R/4=INT(R/4)) AND (R/100<>INT(R/100))
    THEN LD=2 ELSE LD=1
370 RETURN
380 DATA 31,28,31,30,31,30,31,31,30,31,30,31
```

Soutěž dětí a mládeže v programování
Přebor České republiky
BRNO 18.- 20. května 1990

ŘEŠENÍ

BASIC - mládež

Příklad 3

Koefficient obtížnosti 2

```
10 DIM D(6,2),S(6),M(6)
20 INPUT "N";N
30 FOR I=1 TO N
40 PRINT "A(";I;"), B(";I;")";
50 INPUT "";D(I,1), D(I,2)
60 NEXT I
80 I=0
90 I=I+1:REM DALSI KOSTKA
95 IF I>N GOTO 600:REM POSLOUPNOST NALEZENA
100 P=1
110 J=1
120 IF J=I GOTO 300
130 IF S(J)=P GOTO 200
140 J=J+1
150 GOTO 120
200 P=P+1
205 Q=1
210 IF P>N GOTO 500
220 GOTO 110
300 S(I)=P
310 M(I)=Q
320 IF I=1 GOTO 90:REM PRVNI KOSTKA SE NETESTUJE
330 AA=1
340 IF M(I-1)=1 GOTO 360
350 AA=2
```

```
360 BB=2
370 IF Q=1 GOTO 390
380 BB=1
390 IF D(S(I-1),AA)=D(P,BB) GOTO 90
400 Q=Q+1
410 IF Q>2 GOTO 200
420 GOTO 300
500 I=I-1
510 IF I<1 GOTO 700:REM POSLOUPNOST NENALEZENA
520 P=S(I)
530 Q =M(I)
540 IF I<>1 GOTO 200
550 GOTO 400
600 PRINT "POSLOUPNOST NALEZENA"
610 FOR I=1 TO N
615 FOR J=1 TO 3*I:PRINT " ",:NEXT J
620 IF M(I)=1 GOTO 650
630 PRINT D(S(I),1);D(S(I),2)
640 GOTO 660
650 PRINT D(S(I),2);D(S(I),1)
660 NEXT I
670 END
700 PRINT "POSLOUPNOST NENALEZENA"
710 END
```

Soutěž dětí a mládeže v programování
Přebor České republiky
BRNO 18.- 20. května 1990

ŘEŠENÍ

BASIC - mládež

Příklad 4

Koefficient obtížnosti 3

```
10  DIM P$(100)
20  DIM I(9)
30  DIM SP(50,3)
40  I=1
50  READ A$
60  GOSUB 670
70  P$(I)=A$
80  IF A$="END" THEN 120
90  I=I+1
100 IF I>50 THEN ?"Chyba,KONEC":END
110 GOTO 50
120 N=I
130 PC=1
140 SPP=0
150 GOSUB 180
160 GOTO 150
170 END
180 X$=P$(PC)
190 X=LEN(X$)
200 IF X<5 THEN 230
210 IF LEFT$(X$,5)="INPUT" THEN 630
220 IF LEFT$(X$,5)="ENDDO" THEN 380
230 IF X<4 THEN 250
240 IF LEFT$(X$,4)="ECHO" THEN 590
250 IF X<3 THEN 290
260 IF LEFT$(X$,3)="LET" THEN 440
270 IF LEFT$(X$,3)="END" THEN END
280 IF LEFT$(X$,2)="DO" THEN ,310
290 PRINT "Chyba"
300 END
```

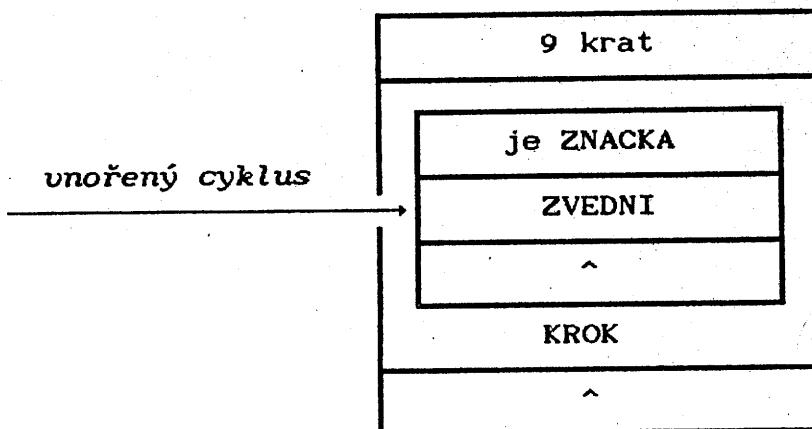
310 SPP=SPP+1
320 SP(SPP,1)=PC
330 SP(SPP,2)=VAL(MID\$(X\$,4,1))
340 I(SP(SPP,2))=1
350 IF MID\$(X\$,6,1)="I" THEN SP(SPP,3)=
 I(VAL(MID\$(X\$,7,1))):GOTO 650
360 SP(SPP,3)=VAL(MID\$(X\$,6,LEN(X\$)-5))
370 GOTO 650
380 I(SP(SPP,2))=I(SP(SPP,2))+1
390 IF I(SP(SPP,2))>SP(SPP,3) THEN 420
400 PC=SP(SPP,1)
410 GOTO 650
420 SPP=SPP-1
430 GOTO 650
440 NV=VAL(MID\$(X\$,5,1))
450 IF MID\$(X\$,7,1)="I" THEN NO=
 VAL(MID\$(X\$,8,1)):GOTO 490
460 NO=VAL(MID\$(X\$,7,LEN(X\$)-6))
470 I(NV)=NO
480 GOTO 650
490 IF MID\$(X\$,9,1)="+" THEN K=1:GOTO 530
500 IF MID\$(X\$,9,1)="-" THEN K=-1:GOTO 530
510 PRINT"Chyba,KONEC"
520 END
530 IF MID\$(X\$,10,1)="I" THEN NP=
 VAL(MID\$(X\$,11,1)):GOTO 570
540 NP=VAL(MID\$(X\$,10,LEN(X\$)-9))
550 I(NV)=I(NO)+K*NP
560 GOTO 650
570 I(NV)=I(NO)+K*I(NP)
580 GOTO 650
590 IF MID\$(X\$,5,1)=""" THEN PRINT
 MID\$(X\$,6,LEN(X\$)-6):GOTO 650
600 NV=VAL(MID\$(X\$,6,1))
610 PRINT I(NV)
620 GOTO 650
630 NV=VAL(MID\$(X\$,7,1))
640 INPUT I(NV)
650 PC=PC+1
660 RETURN

```
670 B$=A$  
680 A$="" "  
690 P=1  
700 FOR JJ=1 TO LEN(B$)  
710 C$=MID$(B$,JJ,1)  
720 IF C$="" THEN P=ABS(P-1)  
730 IF C$="" AND P=1 THEN 750  
740 A$=A$+C$  
750 NEXT JJ  
760 RETURN  
770 DATA "ECHO'Ukazkovy program'"  
780 DATA "ECHO'Kolik cisel chces scitat: '"  
790 DATA "INPUT I2"  
800 DATA "LET I1=0"  
810 DATA "DO I3=I2"  
820 DATA "ECHO'Zadej cislo'"  
830 DATA "INPUT I4"  
840 DATA "LET I1=I1+I4"  
850 DATA "ENDDO"  
860 DATA "ECHO'Soucet je: '"  
870 DATA "ECHO I1"  
880 DATA "END"
```

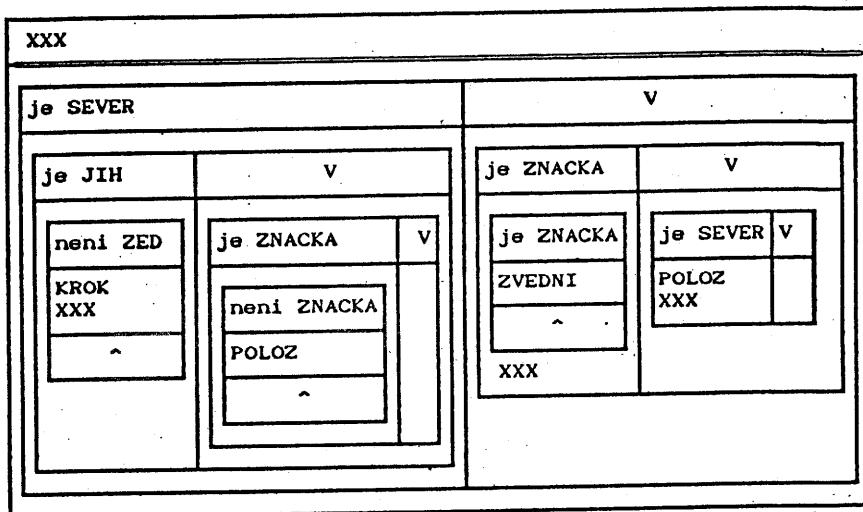
Teoretický test kategorie A,B - Karel

1. Jmenuj strukturované (složené) příkazy jazyka Karel.
kdyz, opakuj, dokud, příkaz procedury
2. Jmenuj jednoduché příkazy jazyka Karel.
POLOZ, ZVEDNI, KROK, VLEVO_VBOK, prázdný příkaz
3. Uveď základní rozdíl mezi jednoduchými a složenými příkazy.
Složené příkazy mohou ve svém těle obsahovat další příkazy.
4. Uveď rozdělení cyklů v jazyku Karel.
 - a) cyklus s daným počtem opakování - opakuj
 - b) cyklus se vstupní podmínkou - dokud
5. Co je to vnořený cyklus. Uveď příklad.

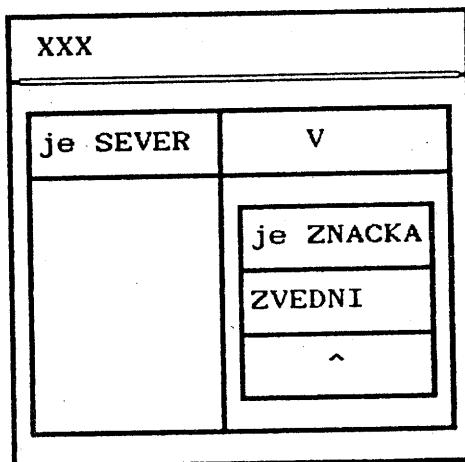
Vnořený cyklus je takový cyklus, který je uvnitř jiného cyklu



6. Zjednoduš, v případě, že to jde, následující proceduru:



řešení



7. Napiš, co dělá procedura z otázky 6. V případě, že existuje situace, při níž by nastala chyba běhu, napiš, jaká chyba nastane a přesně popiš situaci, kdy nastane.

V případě, že Karel stojí natočen na sever, nic neudělá, jinak vysbírá značky na poli, kde stojí (jsou-li tam nějaké). Chyba běhu nenastane v žádném případě.

8. Spočítej a uveď, z kolika příkazů se skládá procedura XXX z otázky 6.

Procedura XXX se skládá z jednoho příkazu - příkazu když, všechny ostatní jsou uvnitř tohoto příkazu.

9. Napiš, v jaké situaci je nevhodné používat v jazyku Karel rekurzi a jakým příkazem ji lze přímo nahradit. (To jest ve kterém případě lze rekurzi nahradit přímo jiným příkazem jazyka Karel, aniž by bylo nutné měnit algoritmus.)

Vždy tam, kde je rekurze použita ve funkci cyklu se vstupní podmínkou a lze ji tedy přímo nahradit příkazem dokud.

Příklad:

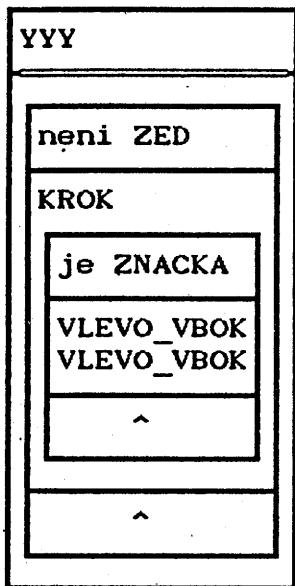
nevhodné řešení

KEZDI	
je ZED	V
	KROK KEZDI

vhodné řešení

KEZDI
neni ZED
KROK

10. Proč není následující procedura správná



Ve vnořeném cyklu jsou příkazy, které v žádném případě nemají vliv na jeho vstupní podmíinku. Z tohoto cyklu se tedy může stát nekonečný cyklus

Teoretický test kategorie C - BASIC

- 1.** Počet vnořených cyklů
 - a) není omezen
 - b) může být maximálně 256
 - > c) záleží na konkrétním BASICu
- 2.** **BUFFER** je
 - > a) vyrovnávací paměť
 - b) podpora BASICu pro rychlé zotavení po chybách
(Basic Unit For Fast Error Recovery)
 - c) doplňkový kód

- 3.** Zjistěte co dělá následující program:

```
10 INPUT X,Y
20 Z=1
30 IF X<0 THEN X=ABS(X):Z=-1
40 IF Y<0 THEN Y=ABS(Y):Z=Z*-1
50 IF X>Y THEN S=X:X=Y:Y=S
60 S=0
70 IF X=0 THEN 110
80 FOR I=1 TO X
90 S=S+Y
100 NEXT I
110 PRINT Z*S
120 END
```

Tento program počítá a tiskne hodnotu $X \cdot Y$ pro libovolná ...
celá čísla

.....
.....
.....

- 4.** Popište stručně pravidla, která platí pro název proměnné v jazyce BASIC.

Název proměnné v BASICu je tvořen pouze z písmen a číslic a musí začínat písmenem. (V některých verzích BASICu smí obsahovat i znak podtržení.) Jako název proměnné nelze ... použít klíčové slovo např. THEN, FOR nebo AUTO. BASIC ... nerozlišuje malá a velká písmena. Délka názvu proměnné je u různých verzích BASICu různá.

.....
.....

5. Napište úryvek programu, který převede celé číslo uložené v řetězci A\$ jako text na číslo do proměnné A.

```
10 A=0 .....  
20 FOR I=1 TO LEN(A$) .....  
30 A=A+MID$(A$,I,1) .....  
40 NEXT I .....  
.....  
.....
```

6. Použití příkazu GOTO v jazyce BASIC

- a) je nesprávné
- b) není nutné

-> c) nelze se bez něj obejít

7. Popište stručně pravidla, která platí pro název proměnné v jazyce BASIC.

Název proměnné v BASICu je tvořen pouze z písmen a číslic a musí začínat písmenem. (V některých verzích BASICu smí obsahovat i znak podtržení.) Jako název proměnné nelze použít klíčové slovo např. THEN, FOR nebo AUTO. BASIC nerozlišuje malá a velká písmena. Délka názvu proměnné je v různých verzích BASICu různá.

```
.....  
.....
```

8. Následující úryvek programu obsahuje chyby. Jaké?

```
10 FOR I=1 TO 30
20 FOR J=0 TO 15
30 X=INT(I/2+J)
40 S$=MID$(A$, I, X)
50 IF S$>"C" THEN 130
60 IF S$>"O" THEN 100
70 S$="JEDNA"
80 X=I
90 GOTO 150
100 S$="DVA"
110 X=2
120 GOTO 150
130 S$="TRI"
140 X=3
150 PRINT S$, X
160 NEXT I
170 NEXT J
```

1. Funkce MID\$ je volána s délkou podřetězce 0
2. Podmínka na řádku 60 nemůže být nikdy splněna
3. Cykly jsou nesprávně ukončeny
4.

9. Zymurgyho brána je
 - a) algoritmus pro řazení metodou průměrných mediánů
 - b) algoritmus pro vícekriteriální zatřídování položek
(obvykle nazývaný SHELL SORT)
 - > c) nesmysl

10. Mikroprocesory Z80 a i8080 mohou přímo adresovat
 - a) maximálně 48 kB paměti
 - > b) maximálně 64 kB paměti
 - c) maximálně 80 kB paměti

Teoretický test kategorie D - BASIC

- x.** Erastothenovo sítí slouží k
 a) hledání společných dělitelů
-> b) hledání prvočísel
 c) rýžování zlata

- 2.** Zjistěte co dělá následující program:

```
10 INPUT X,Y
20 Z=I
30 IF X<0 THEN X=ABS(X):Z=-I
40 IF Y<0 THEN Y=ABS(Y):Z=Z*-I
50 IF X>Y THEN S=X:X=Y:Y=S
60 S=0
70 IF X=0 THEN 110
80 FOR I=I TO X
90 S=S+Y
100 NEXT I
110 PRINT Z*S
120 END
```

Tento program počítá a tiskne hodnotu $X \times Y$ pro libovolná ...
celá čísla
.....
.....

- 3.** Počet vnořených cyklů
 a) není omezen
 b) může být maximálně 256
-> c) záleží na konkrétním BASICu

- 4.** BUFFER je
-> a) vyrovnávací paměť
 b) podpora BASICu pro rychlé zotavení po chybách
 (Basic Unit For Fast Error Recovery)
 c) doplňkový kód

5. Napište úryvek programu, který převede celé číslo uložené v řetězci A\$ jako text na číslo do proměnné A.

```
10 A=0 .....  
20 FOR I=1 TO LEN(A$) .....  
30 A=A+MID$(A$,I,1) .....  
40 NEXT I .....  
.....
```

6. Použití příkazu GOTO v jazyce BASIC

a) je nesprávné

b) není nutné

-> c) nelze se bez něj obejít

7. Následující úryvek programu obsahuje chyby. Jaké?

```
10 FOR I=1 TO 30  
20 FOR J=0 TO 15  
30 X=INT(I/2+J)  
40 S$=MID$(A$,I,X)  
50 IF S$>"C" THEN 130  
60 IF S$>"O" THEN 100  
70 S$="JEDNA"  
80 X=1  
90 GOTO 150  
100 S$="DVA"  
110 X=2  
120 GOTO 150  
130 S$="TRI"  
140 X=3  
150 PRINT S$,X  
160 NEXT I  
170 NEXT J
```

1. Funkce MID\$ je volána s délkou podřetězce 0

2. Podmínka na řádku 60 nemůže být nikdy splněna

3. Cykly jsou nesprávně ukončeny

4.

8. Jeden kilobyte má

- a) 1000 bitů
- b) 1024 bitů
- c) 8000 bitů
- > d) 8192 bitů

9. Mikroprocesor je

- a) jednopřůchodový kompilátor mikroBASICu
- > b) řídící jednotka mikropočítace
- c) knihovna mikroprogramů ve strojovém kódu

10. Myš je

- a) slangové označení pro digitizer
- b) slangové označení pro scanner
- > c) drobný hlodavec