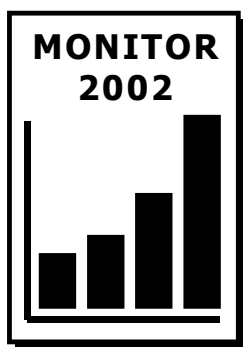


# **M O N I T O R 2002**

## **pilotné testovanie maturantov**





# **Informatika**

## **1. časť**

**Odborný garant projektu:** Štátny pedagogický ústav, Bratislava

**Realizácia projektu:** EXAM<sup>®</sup>, Bratislava

© (2002) Štátny pedagogický ústav

<b>01</b>	Pod lokalizáciou softvéru rozumieme
	<p>(A) uloženie softvéru do lokálneho adresára.          (B) uloženie softvéru na server do lokálnej počítačovej siete.          (C) vytvorenie národnej verzie softvéru.          (D) vytvorenie novej kódovacej tabuľky pre národnú abecedu.</p>
<b>02</b>	Zápis čísla 11C v číselnej pozičnej sústave s najmenším možným základom predstavuje v desiatkovej sústave číslo
	<p>(A) 155.                      (B) 194.                      (C) 195.                      (D) 284.</p>
<b>03</b>	Pri skenovaní obrázkov je potrebné nastaviť parameter dpi. Čím je parameter dpi väčší, tým sa zoskenovaný obrázok skladá
	<p>(A) z menšieho počtu bodov.                      (B) z väčšieho počtu bodov.          (C) z menšieho počtu farebných odtieňov.                      (D) z väčšieho počtu farebných odtieňov.</p>
<b>04</b>	Ktoré z uvedených tvrdení o vstupno-výstupných zariadeniach je <u>nepravdivé</u> ?
	<p>(A) Skener je výstupné zariadenie slúžiace na digitalizáciu predlohy (obrázka, textu, negatívu).          (B) Mikrofón je vstupné zariadenie slúžiace na digitalizáciu zvuku.          (C) Tlačiareň je výstupné zariadenie dávajúce výstup na papier.          (D) Grafický tablet je vstupné zariadenie umožňujúce kresliť špeciálnym perom.</p>
<b>05</b>	Koľko z uvedených tvrdení (1) – (4) je pravdivých?
	<p>(1) Mikroprocesor vykonáva inštrukcie programu uloženého v pamäti a riadi činnosť ostatných častí počítača.          (2) Úlohou pevného disku je uchovávať údaje aj po vypnutí počítača a vyznačuje sa veľkou záznamovou kapacitou.          (3) Pamäť ROM umožňuje vykonávať iba operáciu čítania, údaje sa po vypnutí počítača stratia.          (4) Pamäť RAM umožňuje vykonávať operácie čítania aj zápisu, poskytuje údaje procesoru.</p>
	<p>(A) 4                      (B) 3                      (C) 2                      (D) 1</p>
<b>06</b>	V hertzoch a jeho násobkoch sa vyjadruje
	<p>(A) uhlopriečka monitora.                      (B) rýchlosť prenosu údajov.          (C) rýchlosť procesora.                      (D) kapacita pamäte.</p>
<b>07</b>	Na disku máme súbor TEST.EXE a spustíme ho. Operačný systém preniesie súbor do  a  začne príkazy zo súboru postupne vykonávať.
	Aké slová treba doplniť na vynechané miesta, aby tvrdenie bolo pravdivé?
	<p>(A) mikroprocesora, operačný systém                      (B) RAM, operačný systém          (C) mikroprocesora, mikroprocesor                      (D) RAM, mikroprocesor</p>

**08** Počítačová zostava má parametre: Pentium 2 GHz; 512 MB RAM; 40 GB HDD; 1,44 MB FDD; CD-ROM 42x; 15" monitor; VGA 32 MB. Znamená to, že

- (A) pamäť na grafickej karte počítača je 512 MB, frekvencia procesora je 2 GHz, operačná pamäť má veľkosť 32 MB.
- (B) operačná pamäť má 40 GB, pevný disk má kapacitu 512 MB, súčasťou je 42-rýchlostná CD ROM.
- (C) počítač umožňuje čítať kompaktné disky, veľkosť jeho operačnej pamäte je 512 MB, procesor má frekvenciu 2 GHz.
- (D) obvod monitora je 15", súčasťou je malá disketová jednotka a 42-rýchlostná CD ROM.

**09** V ktorej z možností sú uvedené iba súčasti hardvéru?

- (A) myš, skener, tlačiareň, procesor
- (B) myš, skener, procesor, operačný systém
- (C) operačný systém, procesor, tlačiareň, tabuľkový kalkulačtor
- (D) operačný systém, tabuľkový kalkulačtor, textový editor, počítačová hra

**10** Obrázok v pamäti je zložený zo 600 x 800 farebných bodov. Najmenej koľko KB pamäte zaberie tento 6-farebný neskomprimovaný obrázok?

- |                                                     |                                                     |
|-----------------------------------------------------|-----------------------------------------------------|
| (A) $\frac{600 \cdot 800}{8 \cdot 1024}$ KB         | (B) $\frac{600 \cdot 800 \cdot 3}{8 \cdot 1024}$ KB |
| (C) $\frac{600 \cdot 800 \cdot 6}{8 \cdot 1024}$ KB | (D) $\frac{600 \cdot 800}{1024}$ KB                 |

**11** Súbor s veľkosťou 10 MB bude pri komunikačnej rýchlosti 56 kbps prenášaný počítačovou sieťou približne

- (A) 15 minút.                      (B) 20 minút.                      (C) 25 minút.                      (D) 30 minút.

**12** Ktoré z tvrdení (1) – (4) sú pravdivé?

- (1) E-mailová adresa nesmie obsahovať skratku www.
- (2) V e-mailovej adrese je pred znakom @ názov poštovej schránky, za znakom @ je doménová adresa servera.
- (3) E-mailová správa sa nedá súčasne poslať na viac adries.
- (4) Pri odosielaní a prijímaní správ e-poštový klient komunikuje s e-poštovým serverom.

- (A) Iba (1), (2).                      (B) Iba (1), (2), (4).  
 (C) Iba (2), (4).                      (D) Iba (2), (3), (4).

**13** Ktoré z uvedených tvrdení o internete je nepravdivé?

- (A) Služba FTP umožňuje umiestňovať súbory na FTP-serveroch.
- (B) Medzi interaktívnu komunikáciu patrí IRC, ICQ, e-mail a videokonferencia.
- (C) Každý počítač pripojený na internet má pridelenú jednoznačnú IP adresu.
- (D) Na jednom serveri môže byť súčasne spustených viac internetových služieb.

Nasledujúci obrázok sa vzťahuje k úlohám číslo 14 – 17.

Zobrazený výrez z tabuľkového procesora obsahuje cenník, v ktorom sme zadali DPH (daň z pridanej hodnoty), názov tovaru, počet kusov jednotlivých druhov tovaru a jednotkovú cenu každého druhu tovaru bez DPH.

	A	B	C	D	E	F
1	<b>Firemný cenník ponúkaného tovaru</b>					
2						
3	DPH:	23%				
4						
5	<b>Tovar</b>	<b>Počet kusov</b>	<b>Cena bez DPH za kus</b>	<b>Cena s DPH za kus</b>	<b>Spolu s DPH</b>	
6	CPU	7	5 200,00 Sk	6 396,00 Sk	44 772,00 Sk	
7	HDD	12	4 700,00 Sk	5 781,00 Sk	69 372,00 Sk	
8	FDD	5	912,00 Sk	1 121,76 Sk	5 608,80 Sk	
9	CD ROM	13	1 419,00 Sk	1 745,37 Sk	22 689,81 Sk	
10	<b>Súčet</b>	<b>37</b>	<b>-</b>	<b>-</b>	<b>142 442,61 Sk</b>	
11	<div style="border: 1px solid black; padding: 10px;"> <p style="text-align: center;"><b>Počet kusov</b></p> <div style="display: flex; align-items: center;"> <div style="border: 1px solid black; padding: 5px; margin-right: 20px;"> <p>■ CPU</p> <p>▨ HDD</p> <p>□ FDD</p> <p>■ CD ROM</p> </div> </div> </div>					
12						
13						
14						
15						
16						
17						
18						
19						
20						
21						
22						

**14** Aký tvar má vzorec v bunke B10, kde je vypočítaný celkový počet kusov tovaru?  
 (A) SUM(B6,B9)      (B) =SUM(B6,B9)      (C) =SUM(B6:B9)      (D) SUM[B6:B9]

**15** Ktorý z uvedených vzorcov po kopírovaní z bunky D6 smerom nadol do riadkov 7 – 9 dá správne výsledky?  
 (A) =C6+B\$3\*C6      (B) =\$C\$6+B3\*\$C\$6  
 (C) =C6+\$B3\*C6      (D) =\$C\$6+\$B\$3\*\$C\$6

**16** Hodnotu v bunke B7 zmeníme z 12 na 15. Ako sa zmenia hodnoty v ostatných častiach tabuľky?  
 (A) Zmení sa iba hodnota v bunke B7.  
 (B) Zmenia sa všetky hodnoty súvisiace s bunkou B7.  
 (C) Zmenia sa iba hodnoty v riadku 7.  
 (D) Zmenia sa iba hodnoty v stĺpci B.

**17** Na vytvorenie grafu pod tabuľkou boli použité zdrojové údaje  
 (A) A5:B10.      (B) B5:B9.      (C) A6:B9.      (D) A5:B9.

**18** V akom poradí sa vykonávajú činnosti (1) – (6) pri práci s elektronickou poštou?

- (1) klientovo prevzatie správy od servera
- (2) otvorenie správy
- (3) spustenie poštového klienta na počítači
- (4) overenie prístupového hesla do zadanej poštovej schránky
- (5) nadviazanie spojenia klienta s vybraným poštovým serverom
- (6) zadanie mena používateľa

(A) (3), (5), (6), (4), (1), (2)

(B) (5), (6), (4), (3), (1), (2)

(C) (3), (6), (5), (4), (1), (2)

(D) (5), (4), (6), (3), (1), (2)

### Text k úlohám 19 – 20.

Program *Poloha* počíta vzdialenosť bodu od stredu kruhu v rovine.

```

program Poloha;
  type tBod = record
    x,y:real;
  end;
  tKruh = record
    S:tBod;
    r:real;
  end;

  var A:tBod;
      k:kruh;

  procedure Vstup;
  begin
    write('Zadaj súradnice bodu A ');
    readln(A.x,A.y);
    write('Zadaj súradnice stredu kruhu a jeho polomer ');
    readln ( {1} );
  end;

  function Vzdial(M,N:tBod):real;
  begin
    Vzdial:=sqrt(sqr(M.x-N.x)+sqr(M.y-N.y));
  end;

begin
  Vstup;
  writeln('Bod A má od stredu vzdialenosť ',Vzdial( {2} ));
end;

```

**19** V príkaze `readln ( {1} );` je prekrytý zápis

(A) k.S,k.r

(B) S.x,S.y,k.r

(C) S.x,S.y,r

(D) k.S.x,k.S.y,k.r

**20** Zápis v prekrytej časti `Vzdial ( {2} )` obsahuje zápis skutočných parametrov

(A) A,k

(B) A,k.S

(C) A.x,A.y,k.S.x,k.S.y

(D) A.x,A.y,k.S.x,k.S.y

## Text k úlohám 21 – 24.

Procedúra *Uprav\_subor* prečíta textový súbor s názvom Text.in, upraví ho a výstup uloží do súboru s názvom Text.out.

```

procedure Uprav_subor;
  var Text,NovyText:text;
      znak:char;
      prva:Boolean;
begin
  {1}
  prva:=true;
  while {2} do
  begin
    read(Text,znak);
    if (znak<>' ')or((znak=' ') and prva) then
      begin write({3});
            prva:=znak<>' ';
          end;
    end;
  close(Text);
  close(NovyText);
end;

```

**21** Aké príkazy na otváranie súborov chýbajú v časti {1} ?

- |                                                                                                   |                                                                                                   |
|---------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------|
| (A) assign(Text,'Text.in');<br>assign(NovyText,'Text.out');<br>rewrite(Text);<br>reset(NovyText); | (B) assign(Text,Text.in);<br>assign(NovyText,Text.out);<br>rewrite(Text);<br>reset(NovyText);     |
| (C) assign(Text,'Text.in');<br>assign(NovyText,'Text.out');<br>reset(Text);<br>rewrite(NovyText); | (D) assign('Text.in',Text);<br>assign('Text.out',NovyText);<br>reset(Text);<br>rewrite(NovyText); |

**22** Prekrytá časť {2} obsahuje výraz

- |                    |                   |
|--------------------|-------------------|
| (A) eoln(NovyText) | (B) eof(NovyText) |
| (C) not eof(Text)  | (D) eof(Text)     |

**23** Prekrytá časť {3} obsahuje výraz

- |                |                    |
|----------------|--------------------|
| (A) Text, ' '  | (B) NovyText, znak |
| (C) Text, znak | (D) NovyText, ' '  |

**24** Procedúra *Uprav\_subor*

- (A) odstráni všetky medzery v texte.
- (B) odstráni všetky medzery v prvej skupine medzier.
- (C) odstráni prvú medzeru v skupine za sebou idúcich medzier.
- (D) skupinu za sebou idúcich medzier nahradí jednou medzerou.

## Text k úlohám 25 – 29.

Korytnačka Žofka pozná príkazy *dopredu*, *vlavo*, *vpravo*.

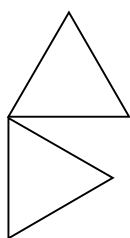
Po vykonaní príkazu *dopredu* (*d*) sa korytnačka posunie o *d* krokov dopredu v smere natočenia a nakreslí za sebou úsečku dĺžky *d*.

Pomocou príkazov *vlavo* (*u*), *vpravo* (*u*) sa korytnačka otočí o *u* stupňov vľavo alebo vpravo od aktuálneho natočenia.

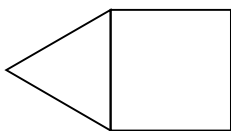
Na začiatku je vždy korytnačka natočená smerom hore.

**25** Čo nakreslí procedúra *Kresli*?

```
procedure Kresli;
begin
  dopredu(20);vpravo(30); dopredu(20);vpravo(120);
  dopredu(20);vpravo(120); dopredu(20);vlavo(150);
  dopredu(20);vpravo(120); dopredu(20);
end;
```



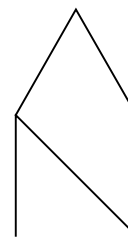
(A)



(B)



(C)

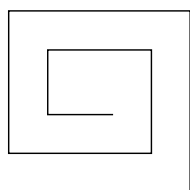


(D)

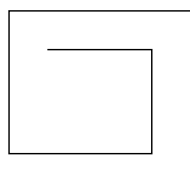
```
procedure Vzor_1(dlzka,zmena:integer);
  var i,p:integer;
begin
  p:=dlzka div zmena;
  for i:=1 to p do
    begin
      dopredu(dlzka);vlavo(90);
      dopredu(dlzka);vlavo(90);
      dlzka:=dlzka-zmena; end;
  end;
```

```
procedure Vzor_2(dlzka,zmena:integer);
begin
  while dlzka>0 do
    begin
      dopredu(dlzka);vlavo(90);
      dopredu(dlzka);vlavo(90);
      dlzka:=dlzka-zmena;end;
  end;
```

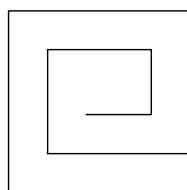
**26** Čo nakreslí procedúra *Vzor\_1* (14, 3)?



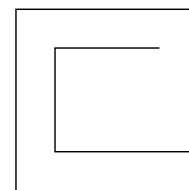
(A)



(B)



(C)



(D)

**27** Procedúra `Vzor_2` (18,15)

- (A) nakreslí o dve úsečky menej ako `Vzor_1` (18,15).
- (B) nakreslí to isté, čo `Vzor_1` (18,18).
- (C) nakreslí to isté, čo `Vzor_1` (18,15).
- (D) nakreslí o dve úsečky viac ako `Vzor_1` (18,15).

**28** Pri ktorom z uvedených volaní nastane chyba pri výpočte (run-time error)?

- (A) `Vzor_1(0,12)`
- (B) `Vzor_1(6,0)`
- (C) `Vzor_2(0,5)`
- (D) `Vzor_2(-7,0)`

**29** Koľko z nasledujúcich volaní procedúr obsahuje nekonečný cyklus?

`Vzor_1(0,2);`                      `Vzor_1(-5,2);`                      `Vzor_1(5,-2);`  
`Vzor_2(0,0);`                      `Vzor_2(-5,0);`                      `Vzor_2(5,0);`

- (A) 3
- (B) 2
- (C) 1
- (D) 0

### Text k úlohám 30 – 31.

Deklarácie premenných

```
var RC:array[1..10]of integer;
```

Predpokladajme, že typ `integer` umožňuje pracovať aj s 11-cifernými číslami.

V poli `RC` je uložených 10 číslic rodného čísla. Úlohou procedúry *Kontrola* je vytvoriť z číslic 10-ciferné číslo a vypísať „Dobré číslo“, ak je číslo deliteľné jedenástimi alebo „Zlé číslo“, ak vytvorené číslo nie je deliteľné jedenástimi.

```
procedure Kontrola;
var cislo,i:integer;
begin
  cislo:=RC[1];
  for i:=2 to 10 do {1}
  if {2}
  then writeln('Dobré číslo.')
  else writeln('Zlé číslo.');
```

```
end;
```

**30** Aký príkaz treba doplniť na miesto {1} procedúry *Kontrola*?

- (A) `cislo:=cislo+10*RC[i];`
- (B) `cislo:=cislo*10+RC[i];`
- (C) `cislo:=cislo+10*RC[11-i];`
- (D) `cislo:=cislo*10+RC[11-i];`

**31** Aký príkaz treba doplniť na miesto {2} procedúry *Kontrola*?

- (A) `cislo/11=cislo div 11`
- (B) `cislo div 11=0`
- (C) `cislo/11=cislo mod 11`
- (D) `cislo mod 11:=0`



## Text k úlohám 32 – 35.

Maťo mal vytvoriť procedúru, ktorá zväčší o 1 hodnoty všetkých prvkov poľa, ktorých hodnota je rovnaká ako hodnota prvku na zadanom mieste.

Príklad: Ak v poli sú hodnoty (-1, 2, 4, 2, 3, 2) a vybraný prvok je na mieste 4, po skončení procedúry budú v poli hodnoty (-1, 3, 4, 3, 3, 3).

Maťo napísal procedúru *Mato*:

```

procedure Mato;
var vybr,i,pom:integer;
    p:array[1..6]of integer;
begin
  {Nacitanie vybraného prvku}
  repeat
    write('Zadaj prvok: ');
    readln(vybr);
  until (vybr>0) and (vybr<7); {1}
  {Uprava pola}
  for i:=1 to 6 do {2}
    if p[i]=p[vybr] {3}
      then p[i]:=p[i]+1; {4}
end;
```

**32** Pri ktorej z nasledujúcich vstupných hodnôt nedôjde k run-time chybe, teda k chybe počas výpočtu procedúry?

- (A) 3.5                      (B) -5                      (C) 123456789012                      (D) 12x

**33** Podmienka {1} (vybr>0) and (vybr<7) za until je ekvivalentná s podmienkou

- (A) not((vybr<1) or (vybr>6))                      (B) (vybr>0) or (vybr<7)  
 (C) not(vybr>7) and not(vybr<0)                      (D) (vybr>=6) and (vybr<=0)

**34** Maťova procedúra nezodpovedá zadaniu. Aké hodnoty budú v poli *p* po vykonaní procedúry *Mato*, ak  $p = (1, 4, 1, 3, 1, 2)$ ,  $vybr = 3$ ?

- (A) (2, 4, 2, 3, 2, 2)                      (B) (1, 4, 1, 4, 1, 2)  
 (C) (2, 4, 2, 3, 1, 3)                      (D) (2, 4, 2, 3, 2, 3)

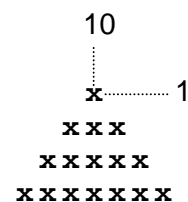
**35** Ako treba nahradiť riadky {2} – {4} procedúry *Mato*, aby procedúra pracovala podľa pôvodného zadania?

- (A) pom:= p[vybr];  
 for i:= 1 to 6 do  
 if p[i]=pom then p[i]:=p[i]+1;
- (B) for i:=1 to vybr-1 do  
 if p[i]=p[vybr] then p[i]:=p[i]+1;  
 for i:=vybr+1 to 6 do  
 if p[i]=p[vybr] then p[i]:=p[i]+1;
- (C) for i:=6 downto 1 do  
 if p[i]=p[vybr] then p[i]:=p[i]+1;
- (D) for i:=1 to 6 do  
 if i<>vybr  
 then if p[i]<>vybr then p[i]:=p[i]+1;

Text k úlohám 36 – 37.

Procedúra `Trojuh(stlpec,riadok,pocet_riadkov:integer)` „vykreslí“ na obrazovku rovnoramenný trojuholník. Vrchol oproti základni je na pozícii `[stlpec, riadok]`, parameter `pocet_riadkov` určuje počet riadkov trojuholníka.

Výstup volania `Trojuh(10,1,4)` je na obrázku.



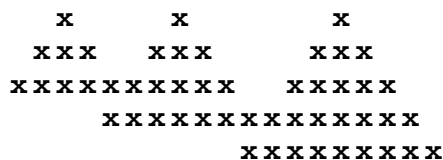
**36** Ktorý z útvarov „vykreslí“ procedúra `Stromcek`?

```

procedure Stromcek;
  var s,r,p:integer;
begin
  s:=10; r:=1; p:=3;
  while p-2<4 do
    begin Trojuh(s,r,p);
      r:=r+p-1; {1}
      p:=p+1;   {2}
    end;
  end;
end;
    
```

<pre> x xxx xxxxx x xxx xxxxx xxxxxxx x xxx xxxxx xxxxxxx xxxxxxx x xxx xxxxx xxxxxxx xxxxxxx     </pre> <p><b>(A)</b></p>	<pre> x xxx xxxxx x xxx xxxxx xxxxx xxxxx xxxxxxx xxx xxxx xxxxx xxxxx xxxxx xxxxx xxxxx xxxxx xxxxx xxxxx     </pre> <p><b>(B)</b></p>	<pre> x xxx xxxxx xxx xxxxx xxxxxxx xxxxxxx xxxxxxx xxxxxxx xxx xxxxx xxxxxxx xxxxxxx xxxxxxx xxxxx xxxxx xxxxxxx xxxxxxx xxxxxxx xxxxxxx     </pre> <p><b>(C)</b></p>	<pre> x xxx xxxxx x xxx xxxxx xxxxxxx xxx xxxxx xxxxxxx xxxxxxx xxxxxxx xxx xxxxx xxxxxxx xxxxxxx xxxxxxx xxxxxxx xxxxxxx xxxxxxx     </pre> <p><b>(D)</b></p>
----------------------------------------------------------------------------------------------------------------------------	-----------------------------------------------------------------------------------------------------------------------------------------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------

**37** Akými príkazmi musíme v procedúre `Stromcek` z predchádzajúcej úlohy nahradiť príkazy označené `{1}`, `{2}`, aby procedúra „vykreslila“ obrázok:



- |                                                                              |                                                                              |
|------------------------------------------------------------------------------|------------------------------------------------------------------------------|
| <p><b>(A)</b> <code>p:=p+1;</code> {1}<br/> <code>s:=s+2*p-1;</code> {2}</p> | <p><b>(B)</b> <code>s:=s+2*p;</code> {1}<br/> <code>p:=p+1;</code> {2}</p>   |
| <p><b>(C)</b> <code>p:=2*p+1;</code> {1}<br/> <code>s:=s+r;</code> {2}</p>   | <p><b>(D)</b> <code>s:=s+2*p-1;</code> {1}<br/> <code>p:=p+1;</code> {2}</p> |

## Text k úlohám 38 – 42.

Definujme funkcie:

`function Vyrez(text:string;zaciatok,pocet:integer):string;`  
 hodnotou funkcie *Vyrez* je podreťazec reťazca *text*, ktorý začína na pozícii *zaciatok* a má *pocet* znakov,

`function Pozicia(podtext,text:string):integer;`  
 hodnotou funkcie *Pozicia* je začiatok prvého výskytu reťazca *podtext* v reťazci *text* alebo 0, ak sa *podtext* v reťazci *text* nenachádza,

`function Dlzka(text:string):integer;`  
 hodnotou funkcie *Dlzka* je počet znakov v reťazci *text*.

Operátor „+“ spojí dva reťazce do jedného.

Premenná *ret* typu *string* má hodnotu 'MatuRiTa'.

Príklady:

príkaz `writeln(Vyrez(ret,5,4))`                      vypíše RiTa

príkaz `writeln(Dlzka(Vyrez(ret,3,2)))`              vypíše 2

príkaz `writeln(Pozicia('a',ret))`                      vypíše 2

**38** Čo vypíše príkaz `writeln(Vyrez(ret,Dlzka(ret)-6,Pozicia('iTa',ret)))`?

(A) Maturit

(B) aturIT

(C) aturit

(D) aturITa

**39** Akú hodnotu bude mať premenná *ret* po nasledujúcich priradeniach?

```
ret:='Matu_RiTa';
ret:=Vyrez(ret,Pozicia('_',ret)+1,Dlzka(ret)-Pozicia('_',ret))+ret;
ret:=Vyrez(ret,1,Pozicia('_',ret)-1);
```

(A) 'MatuRiT a'

(B) 'RiT a\_Matu'

(C) 'RiT aMatu'

(D) 'RiT aMatuRiT a'

**40** Čo vypíše procedúra *Orez*?

```
procedure Orez;
var d,h:integer;
    hadaj,ret:string;
begin
  ret:='MatuRiT a';
  hadaj:='';
  d:=1;h:=Dlzka(ret);
  while d<h do
  begin hadaj:=Vyrez(ret,h,1)+hadaj+Vyrez(ret,d,1);
        d:=d+1;h:=h-1;
  end;
  writeln(hadaj);
end;
```

(A) RiTaMatu

(B) aTiRutaM

(C) RiTautaM

(D) RiTaMatu

- 41** Na krúžku z informatiky deti hľadali podslová v slovách. Vyhral ten, kto našiel najviac podslov v danom slove. Jožko Kompik si napísal procedúru *Hladaj*.

```

procedure Hladaj;
var z,d:integer;
    slovo:string;
begin
    slovo:='maturita'; {1}
    for d:=Dlzka(slovo) downto 1 do
        begin z:=0;
            while z+d<=Dlzka(slovo) do
                begin z:=z+1;
                    writeln(Vyrez(slovo,z,d)); {2}
                end;
            end;
        end;
end;

```

Ktoré slová procedúra vypísala na prvých troch miestach?

maturita	maturita	maturita	maturita
aturita	maturit	maturit	aturita
turita	maturi	aturita	maturit
<b>(A)</b>	<b>(B)</b>	<b>(C)</b>	<b>(D)</b>

- 42** Príkaz v riadku {1} zmeníme na príkaz `slovo:='Laska'`. Koľkokrát sa v procedúre *Hladaj* vykoná príkaz {2} `writeln(Vyrez(slovo,z,d));` ?

**(A)** 10-krát                      **(B)** 15-krát                      **(C)** 20-krát                      **(D)** 25-krát

**Nasledujúci algoritmus sa vzťahuje k úlohám 43 – 44.**

```

procedure Vypocet;
var a,b,min:integer;
begin
    write('Zadaj dve prirodzené čísla: ');readln(a,b);
    if a<b then min:=a
        else min:=b;
    while (a mod min <> 0) or (b mod min <> 0) do
        min:=min-1;
    writeln('hľadané číslo = ',min);
end;

```

- 43** Najviac koľkokrát sa môže vykonať príkaz v cykle `while`?  
 Poznámka:  $\min(a, b)$  je menšie z čísel  $a, b$ ,  $\max(a, b)$  je väčšie z čísel  $a, b$ .

**(A)** raz                                              **(B)**  $\min(a, b)$ -krát  
**(C)**  $(\min(a, b) - 1)$ -krát                      **(D)**  $\max(a, b)$ -krát

- 44** Koľkokrát sa vykoná príkaz v cykle `while` pri rovnosti čísel  $a, b$ ?  
 Poznámka:  $\min(a, b)$  je menšie z čísel  $a, b$ ,  $\max(a, b)$  je väčšie z čísel  $a, b$ .

**(A)** ani raz                      **(B)** raz                      **(C)**  $a$ -krát                      **(D)**  $\min(a, b)$ -krát

## Text k úlohám 45 – 50.

Na tanečnej súťaži sa môže zúčastniť najviac 30 dievčat. V textovom súbore Tanec.txt sú mená súťažiacich, ktoré sa prihlásili. Žiadne meno nie je dlhšie ako 30 znakov.

**45** Mená tanečníčok budú uložené v poli *Tanecnicky* :

```
var Tanecnicky:tUcastnici;
```

Ako je najvhodnejšie definovať typ tUcastnici?

- (A) type tUcastnici = array[1..40] of string;
- (B) type tUcastnici = array[1..30] of array[1..30] of char;
- (C) type tUcastnici = array[1..900] of char;
- (D) type tUcastnici = array[1..30] of string[30];

**46** Procedúra *Citaj* načíta do poľa *Tanecnicky* mená všetkých súťažiacich a počet prihlásených účastníkov vráti v premennej *pocet*. Ako má byť definovaná hlavička procedúry *Citaj*?

- (A) Citaj(var pocet:integer;Tanecnicky:array[1..30] of string[30]);
- (B) Citaj(pocet:integer;Tanecnicky:tUcastnici);
- (C) Citaj(pocet:integer;var Tanecnicky:tUcastnici);
- (D) Citaj(var pocet:integer;var Tanecnicky:tUcastnici);

Poradie, v akom budú tanečníčky súťažiť, určí procedúra *Urci\_poradie* tak, že mená účastníčok v poli *Tanecnicky* náhodne poprehadzuje. Každé meno sa vymení s iným, náhodne určeným funkciou *random*.

Hodnotou funkcie *random*(*n*) je náhodné celé číslo z intervalu  $\langle 0; n-1 \rangle$ .

```
procedure Urci_poradie(pocet:integer;var Tanecnicky:tUcastnici);
var poz:integer;
    pomoc:string[30];
begin
  for i:=1 to pocet do
    begin poz:= {1}
           {2}
    end;
  end;
```

**47** Aký príkaz treba doplniť na prekrytú časť {1} ?

- (A) random(30)+1;
- (B) random(pocet)+1;
- (C) random(pocet);
- (D) random(30);

**48** Ktoré z uvedených príkazov treba doplniť na miesto {2}, ak chceme vymeniť dve mená z poľa *Tanecnicky*?

- (A) `pomoc:=poz; poz:=i; i:=pomoc;`
- (B) `Tanecnicky[pomoc]:=Tanecnicky[poz];  
Tanecnicky[poz]:=Tanecnicky[i];  
Tanecnicky[i]:=Tanecnicky[pomoc];`
- (C) `pomoc:=Tanecnicky[i];  
Tanecnicky[i]:=Tanecnicky[poz];  
Tanecnicky[poz]:=pomoc;`
- (D) `Tanecnicky[poz]:=Tanecnicky[i];  
Tanecnicky[i]:=Tanecnicky[poz];`

Súťaž hodnotí 9 rozhodcov známkami od 0 do 8 (0 je najnižšia známka, 8 je najvyššia známka). Bodový zisk tanečníčky je súčet známok od jednotlivých rozhodcov bez najlepšej a najhoršej známky. Funkcia *Bodovy\_zisk* načíta známky od jednotlivých rozhodcov a vypočíta počet získaných bodov.

```
function Bodovy_zisk:integer;
  var znamka,sucet,najhorsia,najlepsia,i:integer;
begin
  najhorsia:=9;
  najlepsia:=0;
  for i:=1 to 9 do
    begin
      readln(znamka)
      sucet:={1};
      if {2} then najlepsia:=znamka;
      if {3} then najhorsia:=znamka;
    end;
  Bodovy_zisk:=sucet-najlepsia-najhorsia;
end;
```

**49** Aký výraz je zakrytý na mieste {1} ?

- (A) `i`                      (B) `znamka`                      (C) `sucet+i`                      (D) `sucet+znamka`

**50** Aké výrazy treba doplniť na zakryté miesta {2} a {3} ?

- (A) {2} `znamka>najlepsia`                      (B) {2} `znamka>najlepsia`  
{3} `znamka<najhorsia`                      {3} `znamka>najhorsia`
- (C) {2} `znamka<najlepsia`                      (D) {2} `znamka<najlepsia`  
{3} `znamka<najhorsia`                      {3} `znamka>najhorsia`

**Koniec 1. časti testu**