

M O N I T O R 2004

pilotné testovanie maturantov



Informatika

Test I-1

Test je určený maturantom vo všetkých typoch stredných škôl, ktorí sa pripravujú na maturitnú skúšku z informatiky alebo z programovania, pričom celkový týždenný počet hodín týchto predmetov v učebnom pláne, ktorý žiaci absolvovali, bol 8 a viac (spolu za 4 roky).

- 01** Zuzana si stiahla z Internetu program typu freeware a keďže je šikovná programátorka, miernymi zmenami ho vylepšila. Prečo nesmie takto upravený program predávať?
- (A) Pretože za pôvodný softvér nezaplatila registračný poplatok.
 (B) Pretože vo vlastných programoch je povolené používať iba malé časti cudzích programov.
 (C) Pretože freeware síce možno meniť, no zmenený program možno používať iba pre vlastnú potrebu.
 (D) Pretože freeware sa nesmie upravovať a nesmie sa použiť jeho zdrojový kód.

- 02** Milan sa dnes sťažoval, že mu mailom prišiel hoax. Znamená to, že mu prišla
- (A) poplašná správa, ktorá upozorňuje na fiktívny vírus.
 (B) správa s prílohou, ktorá obsahovala vírus.
 (C) správa, ktorá bola automaticky odoslaná zo zavíreného počítača.
 (D) správa upozorňujúca na najnovšiu verziu antivírusového programu (HOT Antivir eXe update).

- 03** Pojmom *Open Source* sa označuje softvér, ktorý
- (A) má otvorený kód, t. j. ide o nedokončený program.
 (B) má zverejnené aj zdrojové kódy a ktorý môžeme ľubovoľne upravovať a ďalej šíriť.
 (C) je možné používať len v otvorených počítačových systémoch.
 (D) iným menom nazývame tiež freeware.

- 04** Tomáš vyplnil tabuľku, v ktorej vyznačil vlastnosti rôznych druhov médií. Do každej bunky vpísal „áno“ (ak médium má danú vlastnosť) alebo „nie“ (ak ju nemá). Všetky údaje uviedol správne.

		Médium			
		A	B	C	D
Vlastnosť média		Pevný disk	Disketa	RAM	CD
1	Po vypnutí počítača si zachová obsah				
2	Je to druh pamäti				
3	Môže obsahovať vírus				

Do ktorých buniek tabuľky vpísal Tomáš slovo „nie“? Bunky označte písmenom stĺpca a číslom riadku.

- 05** Antivírusový program ohlásil, že v pamäti počítača je vírus. Ak nechceme prísť o dáta uložené na disku, je najvhodnejšie
- (A) zazálohovať všetky súbory, naformátovať disk a znovu nainštalovať operačný systém.
 (B) okamžite spustiť liečenie antivírusovým programom nainštalovanom v počítači.
 (C) reštartovať počítač z nezavírenej diskety a potom spustiť liečenie antivírusovým programom.
 (D) reštartovať počítač a spustiť liečenie antivírusovým programom nainštalovaným na disku počítača.

- 06** Na digitálnom fotoaparáte máme nastavené rozlíšenie, pri ktorom sa fotografie ukladajú ako bitmapové súbory s veľkosťou 600 KB s 24 bitovým farebným kódovaním. Ak prekonvertujeme takúto fotografiu do 256 farieb, akú bude mať približne veľkosť?

- (A) 6 000 KB (B) 400 KB (C) 200 KB (D) 150 KB

07 Katka má fotografiu triedy uloženú v bitmapovom súbore a chce ju poslať MMS-kou na Petrov mobilný telefón. Fotka má rozmery 256 x 180 pixelov (obrazových bodov) a je v nej použitých 4096 farieb. MMS-ka môže mať maximálne 5 KB. Najmenej koľko MMS musí Katka poslať, ak chce poslať celú fotografiu rozloženú do viacerých MMS?

08 Andrej poslal Vladovi mailom video z výletu do jeho poštovej schránky na školskom serveri. Súbor mal veľkosť 10 MB. Vlodo si ho chcel pozrieť na počítači v školskej sieti. Školský server je pripojený k Internetu rýchlosťou 128 kbps, Andrej má pripojenie s rýchlosťou 56 kbps a prenosová rýchlosť v rámci školskej siete je 10 Mbps. Najmenej koľko sekúnd musel Vlodo čakať na prenesenie videa z jeho poštovej schránky na lokálny počítač? (Predpokladajte, že sa prenášala iba táto informácia a zanedbajte prevádzkové straty.)

09 Eva sleduje cez Internet komprimované video, ktoré je sériou bitmáp s veľkosťou 320 x 240 pixelov (obrazových bodov) v 256 farbách. Koeficient kompresie videa je 0,3. Aké rýchle pripojenie musí mať Eva, aby mohla toto video sledovať rýchlosťou dva obrázky za sekundu? Výsledok uveďte v kbps.

10 Naskenovali sme stránku textu napísaného v bežnom textovom editore, ktorú sme chceli publikovať na webe. Všimli sme si, že autor mal problémy s pravopisom, preto sme sa rozhodli, že v naskenovanom súbore pred jeho uverejnením na webe opravíme pravopisné chyby. Ktoré z nasledujúcich postupov sú na to vhodné?

- (1) Naskenovaný súbor otvoríme v bitmapovom grafickom editore a v ňom vygumujeme nesprávne písmená a nakreslíme správne.
- (2) Naskenovaný súbor otvoríme v textovom editore a použijeme funkciu *nájdí a nahradí*.
- (3) Na naskenovaný súbor použijeme program na rozpoznávanie textu a po ukončení rozpoznávania v texte opravíme chyby.
- (4) Naskenovaný súbor otvoríme vo vektorovom grafickom editore a použijeme funkciu *nahradí tvar iným tvarom*.

(A) Iba postupy (1) a (2).

(B) Iba postupy (1) a (3).

(C) Iba postupy (3) a (4).

(D) Iba postupy (2) a (4).

11 Aký je základný rozdiel medzi ICQ a IRC?

- (A) Ak chceme používať ICQ, musíme mať trvalé pripojenie na Internet, k používaniu IRC nám stačí aj pripojenie cez dial-up.
- (B) Ak chceme používať IRC, musíme mať trvalé pripojenie na Internet, k používaniu ICQ nám stačí aj pripojenie cez dial-up.
- (C) Ak chceme používať IRC, musíme sa zaregistrovať a získať trvalú identitu, pre používanie ICQ nie je registrácia nutná.
- (D) Cez IRC môžeme komunikovať iba s niekým, kto je práve prihlásený, v ICQ môžeme poslať správy aj neprihláseným osobám.

12 DNS je

- (A) komunikačný protokol, ktorý umožňuje preklad IP adresy na symbolickú adresu.
- (B) služba Internetu, ktorá prekladá symbolické adresy na IP adresy a naopak.
- (C) služba Internetu, ktorá vyberá z databázy mien protokolov komunikačný protokol.
- (D) komunikačný protokol, ktorý zabezpečuje vznik spojenia s webovým serverom.

13 Michal si vytvoril na serveri www.x.edu.sk svoju webovú stránku. Adresu stránky poslal všetkým kamarátom, no Ľuboš ho upozornil na chybu. Zobrazila sa mu iba úvodná stránka [index.html](http://www.x.edu.sk/index.html), no odkaz na stránku [novinky.htm](http://www.x.edu.sk/novinky.htm) nefungoval. Prečo najpravdepodobnejšie Ľubošovi nefungoval odkaz?

- (A) Každá [www](http://www.x.edu.sk)-stránka musí mať koncovku `.html`.
- (B) Server www.x.edu.sk mal počas celého dňa výpadok.
- (C) Michal zabudol správne nastaviť prístupové práva k tomu súboru.
- (D) Michal v odkaze uviedol namiesto `http://www.x.edu.sk/novinky.htm` len `novinky.htm`.

14 V každom z nasledujúcich riadkov sú štyri pojmy, z ktorých tri k sebe logicky patria a štvrtý sa k nim nehodí:

1. repeater, switch, modem, hub
2. klávesnica, myš, skener, ploter
3. mp3, zip, wav, midi

Keby sme z každého riadku vybrali ten pojem, ktorý medzi ostatné nepatrí, ktoré tri pojmy by sme vybrali?

- (A) modem, klávesnica, mp3
- (B) repeater, myš, zip
- (C) switch, ploter, midi
- (D) modem, ploter, zip

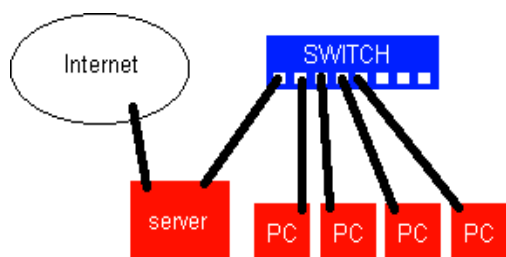
15 Ktoré z nasledujúcich štyroch tvrdení sú pravdivé?

- (1) Každý operačný systém umožňuje nastaviť prístupové práva k súborom pre jednotlivých užívateľov.
- (2) Dva počítače zapojené do jednej počítačovej siete musia pracovať s rovnakým operačným systémom.
- (3) Pojmom *multitasking* sa označuje vlastnosť operačného systému, ktorá umožňuje pracovať naraz s viacerými aplikáciami.
- (4) Operačný systém pri spustení aplikácie najskôr nahrá program do pamäte a až potom ho začne vykonávať procesor.

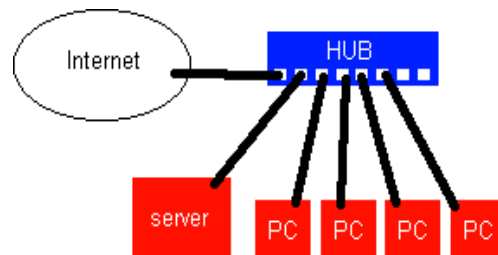
- (A) Iba (1) a (2). (B) Iba (2) a (3). (C) Iba (3) a (4). (D) Iba (1) a (4).

16 Koľko MB pamäte zaberie súbor s minútovou nahrávkou skladby s týmito charakteristikami: 44 100 Hz, 16-bitové stereo, kompresný pomer 0,2? (Výsledok zaokrúhlite na 1 desatinné miesto.)

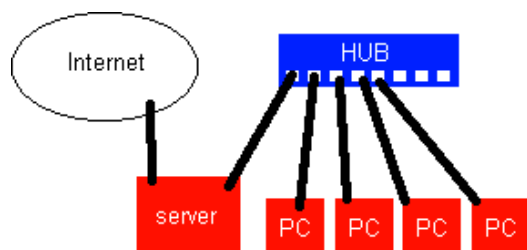
17 Ktorá z uvedených schém zapojenia malej firemnej siete je najmenej bezpečná?



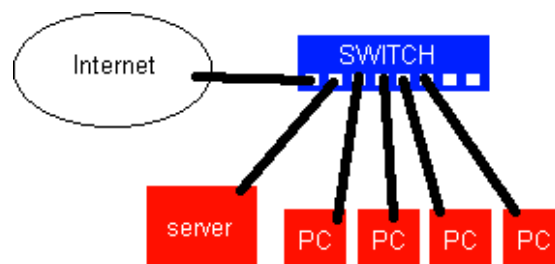
(A)



(B)

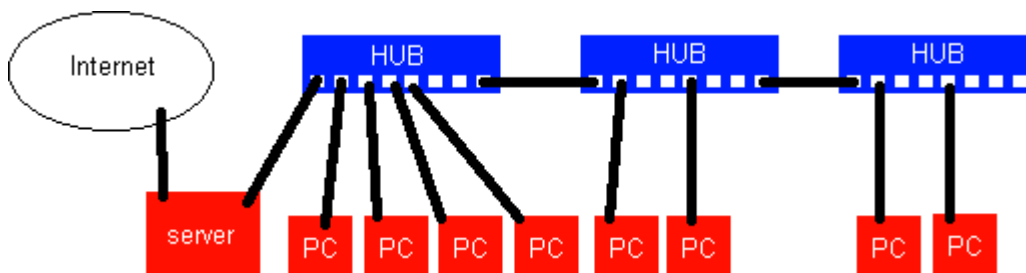


(C)



(D)

18 Akú topológiu má sieť, ktorá je zakreslená na obrázku?



- (A) Stromovú. (B) Kruhovú. (D) Hviezdovú. (C) Zbernicovú (bus).

19 Nasledujúce štyri charakteristiky sú z katalógu predajcu počítačov a počítačových komponentov. Ktorá z nich popisuje parametre monitora?

- (A) XGA 1024 x 768, 10/100 LAN, 802.11b WLAN, AC '97 Audio, 2 x PCMCIA, 1 x 1394, 1 x VGA, 3 x USB 2.0, IrDA, S-Video, 56k FaxModem (nehomologizovaný), MS, SD, MMC card-reader, touchpad, zdroj, 265 x 320 x 31mm, 2,4 kg, MS Windows XP Home edition
- (B) 2 x DDRAM, 2 x PCI, 1 x CNR, UDMA133, 2 x FDD, 1 x Serial, 1 x Paralell, 6 x USB 2.0, IRD on board, audio s wave table, 128 bit 3D shared VGA
- (C) 2048 x 1536, F2,9 / F5,6, digital zoom 4x, blesk, 2 x AA batéria nie je v cene, interná 16 MB pamäť rozširiteľná o SD / MMC card (nie je v cene) 1,6" TFT displej, obal, softvér
- (D) 1920 x 1440 / 60Hz, 1280x1024 / 104Hz, 0,28/0,24mm, 50 – 160 Hz, 30 kHz – 111 kHz, LightFrame 3, TCO99, Video Bandwith 261 MHz

Text k úlohám 20 – 23

Eva rada cestuje vlakom po Slovensku. Aby sa jej ľahšie cestovalo, vytvorila si v tabuľkovom kalkulátore cestovný poriadok. Na obrázku vidíte ukážku z jej veľkej tabuľky. V ukážke sú zobrazené iba niektoré riadky, spolu má tabuľka 1000 riadkov. Každý vlakový spoj je rozpísaný v niekoľkých riadkoch. Každý riadok obsahuje tieto informácie:

- vlak – označenie vlaku,
- stanica – názov stanice, v ktorej vlak zastavuje (prípadne východisková a konečná stanica),
- príchod – čas príchodu vlaku do stanice (ak sa jedná o východiskovú stanicu, bunka je prázdna),
- odchod – čas odchodu vlaku zo stanice (v cieľovej stanici je bunka prázdna),
- km – vzdialenosť v km od začiatkovej stanice vlaku.

Jednotlivé stanice vlaku sa nachádzajú v riadkoch pod sebou a sú zoradené chronologicky (presne tak, ako vlak stanicami prechádza).

	A	B	C	D	E	
1	vlak	stanica	km	príchod	odchod	
2	R 605 Dargov	Bratislava hl. st.	0		9:50	
3	R 605 Dargov	Žilina	203	12:34	12:40	
4	R 605 Dargov	Poprad - Tatry	344	14:31	14:34	
5	R 605 Dargov	Kysak	429	15:37	15:39	
6	R 605 Dargov	Košice	445	15:50	16:26	
7	Zr 1844 Rozsutec	Zvolen osob. st.	0		15:17	
8	Zr 1844 Rozsutec	Banská Bystrica	21	15:41	15:43	
9	Zr 1844 Rozsutec	Turčianske Teplice	66	16:27	16:28	
10	Zr 1844 Rozsutec	Martin	89	16:45	16:47	
11	Zr 1844 Rozsutec	Vrútky	96	16:54	17:06	
12	Zr 1844 Rozsutec	Žilina	117	17:25		
13	IC 501 Tatran	Bratislava hl. st.	0		5:45	
14	IC 501 Tatran	Žilina	203	7:47	7:50	
15	IC 501 Tatran	Poprad - Tatry	344	9:26	9:28	
16	IC 501 Tatran	Kysak	429	10:21	10:22	
17	IC 501 Tatran	Košice	445	10:33		
...						
996	R 603 Čingov	Bratislava hl. st.	0		7:50	
997	R 603 Čingov	Žilina	203	10:34	10:40	
998	R 603 Čingov	Poprad - Tatry	344	12:31	12:34	
999	R 603 Čingov	Kysak	429	13:37	13:39	
1000	R 603 Čingov	Košice	445	13:50		
...						

20 Funkcia $\text{Min}(\text{oblasť buniek})$ vypočíta minimálnu číselnú hodnotu z buniek v zadanej oblasti. Aký vzorec musí Eva vložiť do bunky G2, ak chce zistiť čas prvého (najsôr vychádzajúceho) vlaku z tabuľky? (Predpokladajme, že v tabuľke sú uvedené iba vlaky, ktoré vychádzajú po polnoci a do cieľovej stanice prídu v priebehu dňa.)

(A) = $\text{Min}(D2:D1000)$

(B) = $\text{Min}(E2:E1000)$

(C) = $\text{Min}(C2:D1000)$

(D) = $\text{Min}(C2:C1000)$

- 21** Eva si chce pomocou filtra zobrazit' všetky vlaky, ktoré nezačínajú v Žiline a príchod do Žiliny majú v čase po 8:00 h, avšak do 10:00 h. Aké podmienky musí pre takéto filtrovanie nastaviť?
- (A) stanica = "Žilina" and (príchod > 8:00) and (príchod < 10:00)
 (B) stanica = "Žilina" or (príchod > 8:00) and (príchod < 10:00) or (km <> 0)
 (C) stanica = "Žilina" and ((príchod > 8:00) or (príchod < 10:00))
 (D) stanica = "Žilina" and ((príchod > 8:00) and (príchod < 10:00)) or (km <> 0)
- 22** Eva si často potrebuje pozerat' odchody a príchody vlakov z jednotlivých staníc. Rozhodla sa preto, že všetky riadky tabuľky zoradí podľa názvu stanice. Ktoré bunky musí pred triedením vysvietiť do bloku?
- 23** Do bunky G2 napísala Eva vzorec = IF(C2 = 0; 0; E2 – D2) a potom ho skopirovala do celého stĺpca G. Čo bude tento vzorec počítat'?
- (A) Čas, za ktorý vlak príde do danej stanice z východiskovej stanice.
 (B) Čas, za ktorý vlak príde do danej stanice z predchádzajúcej stanice.
 (C) Čas státia vlaku v danej stanici.
 (D) Čas, za ktorý vlak príde z danej stanice do cieľovej stanice.

Text k úlohám 24 a 25

Autor tohto dokumentu nepozná základné zásady písania elektronického textu. Namiesto použitia tabuľátora sa snažil písať do stĺpcov a pomáhal si pritom vkladáním medzier. Pretože súbor má veľa riadkov, trvalo by dlho opraviť manuálne všetky riadky.

X	1	2	3	4	5	6	7	8	9	10	11	12
Janka	Moja	Košická	21	094	08	¶		
Tatiana	Kolárová	→	Pekná	58	→	→	→	210	58
Stanislav	Dobrák	Tepelná	8	→	821	54	¶	
Ján	→	Krivý	Krátka	26	622	56	¶		

(znak „ ¶ “ zastupuje enter, znak „ → “ zastupuje tabuľátor, znak „ . “ zastupuje medzeru)

- 24** Tu sú štyri možné postupy úpravy pôvodného textu:
- (1) Zmena pravého okraja na 12 cm.
 - (2) Zmena veľkosti písma.
 - (3) Zmena fontu z proporcionálneho na neproporcionálny.
 - (4) Zmena riadkovania.
- Pri ktorých dvoch z nich by sa porušilo zarovnanie do stĺpcov?
- (A) Pri postupoch (1) a (2). (B) Pri postupoch (2) a (3).
 (C) Pri postupoch (3) a (4). (D) Pri postupoch (1) a (4).

Test pokračuje na d'alšej strane.

25 Rozhodli sme sa upraviť text tak, aby v každom riadku boli jednotlivé slová oddelené práve jedným tabulátorom (medzi ulicou a číslom musí zostať jedna medzera, rovnako v smerovom čísle). Pritom vieme, že medzi jednotlivými slovami sú aspoň dve medzery alebo jeden tabulátor. Použijeme funkciu textového procesora *nájdí a nahradí*. Vyznačíme celý text a aplikujeme niektoré z nasledujúcich nahradení:

- (1) Nájdí všetky výskyty medzery a každú nahradí jedným tabulátorom.
- (2) Nájdí všetky výskyty dvoch za sebou idúcich medzier a nahradí ich jedným tabulátorom.
- (3) Nájdí všetky výskyty dvoch za sebou idúcich medzier a nahradí ich jednou medzerou.
- (4) Nájdí všetky výskyty dvoch za sebou idúcich tabulátorov a nahradí ich jedným tabulátorom.
- (5) Nájdí všetky výskyty medzery, za ktorou nasleduje tabulátor a nahradí ich jedným tabulátorom.
- (6) Nájdí všetky výskyty tabulátora, za ktorým nasleduje medzera a nahradí ich jedným tabulátorom.

Ktoré štyri z daných nahradení treba aplikovať?

(A) (2), (4), (5) a (6).

(B) (1), (2), (3) a (4).

(C) (1), (3), (4) a (6).

(D) (1), (2), (5) a (6).

26 Nasledujúci zoznam obsahuje základné etapy riešenia nejakého problému s pomocou počítača:

- (1) skúšobná prevádzka
- (2) zadanie objednávky od zákazníka
- (3) analýza problému
- (4) testovanie
- (5) návrh algoritmu a dátových štruktúr
- (6) programovanie

Vypíšte čísla etáp v správnom časovom poradí.

Text k úlohám 27 a 28

Definujme procedúru *kruznice*, ktorá vykresľuje na obrazovku zadaný počet sústredných kružníc s náhodným polomerom:

```
procedure kruznice (var pocet:Integer);           {1}
var r:Integer;
begin
  for i:= 1 to pocet do
    begin
      r:= 10 + random(50);
      Circle(320, 240, r);
    end;
  pocet:= pocet * 2;
end;
```

a majme úsek programu, ktorý využíva procedúru *kruznice*:

```
pocet:= 1;
while pocet < 9 do
  begin
    kruznice(pocet);
    pocet:= pocet + 1;
  end;
```


- 27** Predpokladajme, že žiadne dve kružnice sa nenakreslia s rovnakým polomerom. Koľko kružníc vykreslí program?
- (A) 9 (B) 11 (C) 12 (D) 81

- 28** Koľko kružníc by sa vykreslilo, ak by sme riadok {1} nahradili nasledovne:

```
procedure kruznice (pocet:Integer);
```

Text k úlohám 29 – 31

Každý zamestnanec firmy má pridelený kód, pod ktorým je registrovaný vo firemnej databáze. Raz za pol roka sa hodnotí práca zamestnancov výkonovým koeficientom z intervalu $\langle 0, 1 \rangle$. Zamestnanec si môže na intranete pozrieť svoje hodnotenie spustením programu Vykon, ktorý na základe zadaného kódu zamestnanca vyhledá a zobrazí jeho výkonový koeficient. Program Vykon používa pri vyhľadávani pole pracovníci, ktoré je utriedené zostupne (klesajúco) podľa kódu pracovníka.

```
Program vykon;
var pracovníci: array[1..600] of record
    kod: string[5];
    koef: real;
end;

z, k, s: integer;
najdikod: string[5];
begin
    {citaj hodnoty do pola pracovníci}
    write('zadaj svoj kod: '); readln(najdikod);
    z:= 1; k:= 600;
    repeat
        s:= (z + k) div 2;
        if {1} then z:= s + 1
        else
            if pracovníci[s].kod <> najdikod
            then k:= s - 1
        until (z > k) or (pracovníci[s].kod = najdikod);
        if z > k then writeln('nepoznam kod')
        else writeln(pracovníci[s].koef);
    end.
```

- 29** Akú podmienku treba doplniť na miesto {1}, aby vyhľadávanie správne fungovalo?

- (A) najdikod <= pracovníci[s].kod
 (B) najdikod > pracovníci[s].kod
 (C) najdikod >= pracovníci[s].kod
 (D) najdikod < pracovníci[s].kod

- 30** Najviac koľkokrát sa vykoná v programe vykon telo cyklu repeat until?

- (A) 8-krát (B) 9-krát (C) 10-krát (D) 11-krát

31 Predpokladajme, že zadaný kód pracovníka v premennej *najdikod* je kód pracovníka, ktorého údaje zabudli pri spracovaní výsledkov hodnotenia vložiť do poľa *pracovnici*. Ak by tieto údaje v poli *pracovnici* boli, tak by boli na mieste

(A) s alebo $s + 1$. (B) $s - 1$. (C) s . (D) $s - 1$ alebo s .

32 Ktorý z nasledujúcich algoritmov napíše najviac hviezdíčiek?

```
i:= 1;
j:= 1;
while (i<=5) and (j<=3) do
begin
  write('*');
  i:= i + 1;
  j:= j + 1;
end;
```

(A)

```
i:= 1;
j:= 1;
while not (i<=5) or (j<=3) do
begin
  write('*');
  i:= i + 1;
  j:= j + 1;
end;
```

(C)

```
i:= 1;
j:= 1;
while (i<=5) or (j<=3) do
begin
  write('*');
  i:= i + 1;
  j:= j + 1;
end;
```

(B)

```
i:= 1;
j:= 1;
while (i<=5) and not(j<=3) do
begin
  write('*');
  i:= i + 1;
  j:= j + 1;
end;
```

(D)

Algoritmus Čokoláda (Text k úlohám 33 – 35)

Nasledujúci algoritmus je návod na výpočet sumy za nákup čokolád pri uplatnení množstvej zľavy. (Symbol \leftarrow vyjadruje priradenie hodnoty do premennej.)

```

Ak poc_cok < 6 tak
    suma  $\leftarrow$  cena_za_kus * poc_cok
inak
    Ak poc_cok < 106 tak
        suma  $\leftarrow$  cena_za_kus * (5 + (poc_cok - 5) * 0,8)
    inak
        suma  $\leftarrow$  cena_za_kus * (5 + 100 * 0,8 + (poc_cok - 105) * 0,5)
    KoniecAk
KoniecAk
vypis suma

```

33 Predpokladajme, že cena_za_kus je 10 Sk. Vypočítajte výslednú sumu, ktorú zaplatíme za nákup čokolád, ak

- a) poc_cok = 4,
- b) poc_cok = 25,
- c) poc_cok = 115.

34 Ako je možné slovne vyjadriť algoritmus Čokoláda?

- (A) Za prvých 5 čokolád zaplatíme plnú sumu, ďalších 100 nakúpime s 20-percentnou zľavou a zvyšok s 50-percentnou zľavou.
- (B) Ak kúpime viac ako 5 čokolád, tak na ďalších 100 kusov dostaneme zľavu 20 %, a ak kúpime viac ako 105 čokolád, tak za každú čokoládu zaplatíme iba 80 % z bežnej ceny.
- (C) Ak kúpime viac ako 106 čokolád, tak na každú dostaneme zľavu.
- (D) Ak kúpime dvesto čokolád, tak na polovicu z nich dostaneme zľavu 80 %.

35 Ktorý z príkazov treba doplniť na miesto {1} v algoritme Cena_Čokolád, ak má byť ekvivalentný s algoritmom Čokoláda?

```

{Algoritmus Cena_Čokolád}
{1}
Ak poc_cok > 5 tak
    cena  $\leftarrow$  cena - (poc_cok - 5) * 0,2 * cena_za_kus
KoniecAk
Ak poc_cok > 105 tak
    cena  $\leftarrow$  cena - (poc_cok - 105) * 0,3 * cena_za_kus
KoniecAk
vypis cena

```

- (A) Ak poc_cok < 5 tak cena \leftarrow poc_cok * cena_za_kus
KoniecAk
- (B) Ak poc_cok < 6 tak cena \leftarrow poc_cok * cena_za_kus
KoniecAk
- (C) cena \leftarrow poc_cok * cena_za_kus
- (D) cena \leftarrow 0

Text k úlohám 36 – 38

Pred výťahom stojí rad ľudí s hmotnosťami od prvého po posledného: 80 kg, 20 kg, 75 kg, 70 kg, 120 kg, 55 kg, 85 kg, 90 kg. Súčasťou výťahu je senzor s váhou, ktorý vie zistiť, či pred výťahom niekto stojí a odvážiť prvého človeka v rade. Výťah je riadený nasledujúcim programom:

zataz ← 0

Kym (je niekto pred vytahom) a (zataz + hmotnostPrvehoVRade < 300) rob

zataz ← zataz + hmotnostPrvehoVRade

Prvy v rade nastup

KoniecKym

- 36** Koľko ľudí z radu sa vyvezie vo výťahu pri prvej jazde?
- 37** Koľkokrát bude musieť ísť výťah hore, aby vyviezol všetkých čakajúcich ľudí, ak nikto ďalší nepríde a ľudia v rade sa nebudú predbiehať?
- 38** Ak by sa tí istí ľudia v rade postavili v inom poradí, tak by možno na ich vyvezenie stačilo menej jazd výťahu. Na aký najmenší počet jazd by mohol výťah vyviezť všetkých čakajúcich ľudí?

Text k úlohám 39 – 41

Pri tréningu sa skokan do výšky riadi nasledujúcim algoritmom:

nastav pociatocnu vysku 180 cm.

opakuje

Ak skocis vysku na prvý pokus, tak zvys latku o 5 cm.

Ak skocis vysku na druhý pokus, tak zvys latku o 3 cm.

Ak skocis vysku na tretí pokus, tak zvys latku o 1 cm.

pokiaľ budeš mať 3 neuspesne pokusy na jednej vyske.

Postupnosť pokusov budeme v ďalšom zapisovať pomocou písmen p – preskočil a n – nepreskočil.

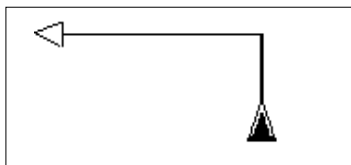
- 39** Akú najvyššiu výšku preskočil skokan, ak postupnosť jeho pokusov bola: p p n n p p n n n?
- 40** Skokan postupne skákal výšky 180 cm, 181 cm, 182 cm, 187 cm, 192 cm, 195 cm, 196 cm. Zapište pomocou písmen p, n jeho postupnosť pokusov.
- 41** Aký najmenší počet pokusov mohol mať pretekár, ak posledná výška, ktorú skákal, bola 192 cm a na ďalšiu sa už nedostal?

Text k úlohám 42 – 45

Procedúra dopredu(*d*: real); *nakreslí čiaru dĺžky d v smere, v ktorom je pero práve otočené z pozície, v ktorej sa práve nachádza. Po nakreslení čiar sa pero nachádza na jej konci a smer pera je rovnaký ako pred kreslením.*

Procedúry vľavo(*u*: real) a vpravo(*u*: real) *zmenia otočenie pera o zadaný uhol vľavo resp. vpravo. Uhol je zadaný v stupňoch. Poloha pera sa týmito príkazmi nemení.*

Príklad: Postupnosť príkazov dopredu(50); vľavo(90); dopredu(100); *nakreslí nižšie uvedený obrázok.*

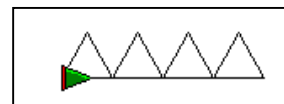


Počiatočná poloha a smer pera sú naznačené plným trojuholníkom. Koncová poloha a smer pera sú naznačené prázdny trojuholníkom. Pri zápornej hodnote parametra sa pero pri kreslení čiar pohybuje dozadu („cúva“).

Procedúra TrojL je definovaná takto:

```
procedure TrojL(d: real);
begin
  dopredu(d); vľavo(120); dopredu(d); vľavo(120); dopredu(d); vľavo(120);
end;
```

42 **Procedúra** Strechy(*pocet*: integer; *d*: real); *nakreslí postupnosť rovnostranných trojuholníkov so stranou dĺžky d. Počiatočná a koncová poloha pera a jeho otočenie sú na začiatku a na konci rovnaké (na obrázku je to naznačené šípkou). Obrázok znázorňuje príklad pre Strechy(4, 50);.*

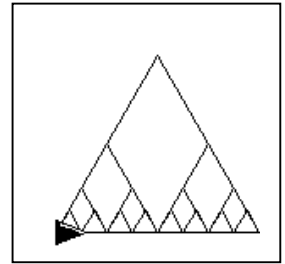


```
Procedure Strechy(pocet: integer; d: real);
var i: integer;
begin
  for i:= 1 to {1} do begin
    TrojL( {2} );
    dopredu(d);
  end;
  dopredu( {3} );
end;
```

Čo treba doplniť na miesta {1}, {2} a {3}?

- (A) {1} 4 {2} 50 {3} 200
 (B) {1} pocet {2} d {3} -(d - 1) * pocet
 (C) {1} 4 {2} 50 {3} pocet * d
 (D) {1} pocet {2} d {3} - d * pocet

43 Procedúru *Strechy* použijeme pri kreslení novej vzorky `Kopce(kolko: integer; d: real);`. Parameter *kolko* vyjadruje počet rôznych veľkostí trojuholníkov a parameter *d* dĺžku strany najväčšieho trojuholníka, ktorý sa má nakresliť. Na obrázku je výsledok príkazu `Kopce(4, 120);`.

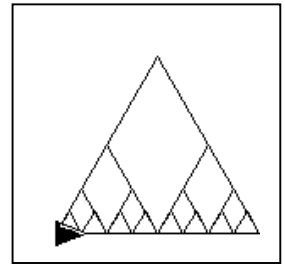


```
procedure Kopce(kolko: integer; d: real);
var p: integer;
begin
  p:= 1;
  while kolko > 0 do begin
    Strechy( {1} );
    p:= p * 2;
    d:= {2} ;
    kolko:= kolko - 1;
  end;
end;
```

Čo treba doplniť na miesta {1} a {2}?

- (A) {1} p, d {2} d * 2; (B) {1} p, d {2} d / 2;
 (C) {1} kolko, d {2} d + 1; (D) {1} kolko, d {2} d - 1;

44 Procedure `KopceR(kolko: integer; d: real);` by mala nakresliť rovnaký obrázok ako `Kopce`. Nakreslíme ho pomocou príkazu `TrojL`. Prvý parameter opäť vyjadruje počet rôznych veľkostí trojuholníkov, ktoré sa majú nakresliť a druhý parameter dĺžku strany najväčšieho trojuholníka, ktorý sa má nakresliť. Na obrázku je výsledok príkazu `KopceR(4, 120);`. Ktorá z nasledujúcich procedúr je správna?



```
Procedure KopceR(kolko:integer;d:real);
begin
  while kolko > 0 do begin
    kolko:= kolko - 1;
    TrojL(d);
    KopceR(kolko, d / 2);
    dopredu(d / 2);
    KopceR(kolko, d / 2);
  end;
end;
```

(A)

```
Procedure KopceR(kolko:integer;d:real);
begin
  if kolko > 0 then begin
    TrojL(d);
    KopceR(kolko div 2, d / 2);
    dopredu(d / 2);
    KopceR(kolko div 2, d / 2);
    dopredu(-d / 2);
  end;
end;
```

(B)

```
Procedure KopceR(kolko:integer;d:real);
begin
  if kolko > 0 then begin
    TrojL(d);
    KopceR(kolko - 1, d / 2);
    dopredu(d / 2);
    KopceR(kolko - 1, d / 2);
    dopredu(-d / 2);
  end;
end;
```

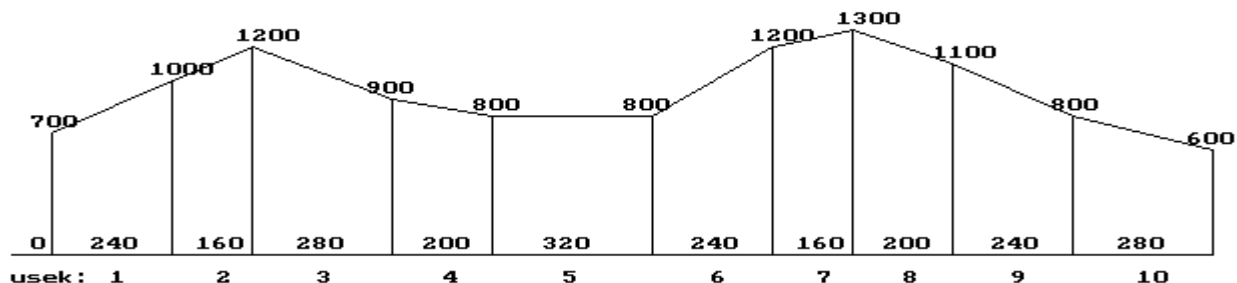
(C)

```
Procedure KopceR(kolko:integer;d:real);
begin
  while kolko > 0 do begin
    TrojL(d);
    KopceR(kolko, d / 2);
    dopredu(d / 2);
    KopceR(kolko, d / 2);
    dopredu(-d / 2);
    kolko:= kolko - 1;
  end;
end;
```

(D)

Text k úlohám 45 – 50

V poliach x , y : `array[0..100]` of integer; sú zaznamenané údaje o N úsekoch turistickej trasy ($1 < N \leq 100$). Zaznamenané sú body na trase, v ktorých sa mení sklon terénu (prudšie stúpa, začína klesať, ...). V $x[0]$, $y[0]$ sú súradnice začiatku, a platí $x[0] = 0$. V $x[i]$ je posun v smere osi x oproti predchádzajúcemu stanovisku, v $y[i]$ je nadmorská výška i -teho bodu na trase. Úsek i má na osi x krajné body $x[i - 1]$ a $x[i]$. Údaje sú v metroch.



- 45** Napíšte príkaz, ktorý je potrebné doplniť na miesto {1} tak, aby program na konci vypísal dĺžku celej trasy na osi x v metroch.

```
{var dlzka, i: integer;}

dlzka:= 0;
for i:= 1 to N do {1} ;
writeln('dlzka trasy v smere osi x: ', dlzka, ' m');
```

- 46** Nasledujúca časť programu má vypísať číslo úseku trasy, do ktorého sa dostaneme po osi x , keď prejdeme od začiatku trasy xm metrov. Dopíšte podmienku na mieste {1}.

```
{var dlzka, xm, i: integer;}

readln(xm); dlzka:= 0;
i:= 1;
while (i <= N) and ( {1} ) do begin
  dlzka:= dlzka + x[i];
  i:= i + 1;
end;
if i <= N then writeln(i)
else writeln('cela turisticka trasa je kratšia ako zadana dlzka');
```

- 47** V polovici šiesteho úseku sme zabudli uviesť zlom trasy, v ktorom je výška 1 000 metrov. Napíšte časť programu, ktorý je potrebné doplniť na miesto {1}, aby program chýbajúci úsek doplnil.

```
{var i: integer;}

for i:= {1} do begin
  x[i + 1]:= x[i]; y[i + 1]:= y[i];
end;
x[7]:= x[6] div 2; x[6]:= x[6] - x[7];
y[6]:= 1000; N:= N + 1;
```

48 Čo vypíše nasledujúci program pre $N = 10$ a pôvodne zadané polia?

```
x: 0,    240,   160,   280,   200,   320,   240,   160,   200,   240,   280
y: 700, 1000, 1200,  900,   800,   800,  1200, 1300, 1100,  800,   600
```

```
{var vyska: longint; i, pocet: integer;}

pocet:= 1; vyska:= y[1] - y[0];
for i:= 2 to N do begin
  if vyska * (y[i] - y[i - 1]) < 0 then begin
    vyska:= y[i] - y[i - 1];
    pocet:= pocet + 1;
  end;
end;
writeln(pocet);
```

49 Za predpokladu, že prvý úsek cesty nie je vodorovný (má sklon), predchádzajúci program vypíše

- (A) počet úsekov na trati, na ktorých trasa neklesá.
- (B) počet úsekov na trati, na ktorých trasa stúpa, resp. nemení nadmorskú výšku.
- (C) počet takých častí trati, na ktorých trasa neklesá alebo nestúpa.
- (D) počet krajných bodov úsekov, v ktorých má trať lokálne minimum alebo maximum.

50 Nasledujúci algoritmus má vypísať najnižšiu nadmorskú výšku na trase. Napíšte časti programu, ktoré majú byť na miestach {1} a {2}.

```
{ var i, ix: integer;}

ix:= 0;
for i:= 1 to N do begin
  if {1} then ix:= i;
end;
writeln('najnižší bod trasy je vo výške ', {2} );
```

51 Aký je rozdiel medzi protokolmi http a https?

- (A) Protokol https je bezpečnejší, pretože komunikácia prebieha šifrovane.
- (B) Protokol https je určený iba na prenos súborov cez službu www, je to náhrada služby ftp.
- (C) Protokol https umožňuje na rozdiel od http vo formulároch zadávať heslá.
- (D) Sú takmer totožné, no https je vývojovo starší, a preto je dnes už menej používaný.

Text k úlohám 52 – 55

Firma na rýchlu dopravu zásielok sídliaca v meste číslo 1 plánuje svoju činnosť pomocou špeciálneho programu. V dvojrozmernom poli *mapa*: *TmojaMapa*, kde *TmojaMapa* = *array*[1..MaxN, 1..MaxN] of *integer*, má program zapísané dĺžky najkratších ciest medzi N mestami ($N \leq \text{MaxN}$), do ktorých firma dopravuje zásielky. V prvku poľa *mapa*[*i*, *j*] je uložená dĺžka najkratšej cesty z mesta *i* do mesta *j* zadaná v kilometroch. Cesty v mape sú obojsmerné, teda by malo platiť *mapa*[*i*, *j*] = *mapa*[*j*, *i*] pre všetky mestá *i*, *j*.

- 52** Vieme, že v poli *mapa* sú zapísané dĺžky ciest medzi mestami len v časti pod hlavnou diagonálou. Napíšte časť programu, ktorý má byť v procedúre *spravsymetriu* na miestach {1} a {2} tak, aby boli v poli *mapa* správne údaje aj nad hlavnou diagonálou.

```
Procedure spravsymetriu(n: integer; var mapa: TmojaMapa);
var r, s: integer;
begin
  for r:= 2 to n do
    for s:= 1 to {1} do
      mapa[ {2} ]:= mapa[r, s];
    end;
  end;
```

- 53** Pred každou cestou sa zásielky zapisujú do textového súboru v tvare číslo_mesta číslo_zásielky. Údaje o každej zásielke sú na novom riadku.

Nasledujúci algoritmus má do poľa *destinacie* spočítať počet zásielok, ktoré treba doručiť do jednotlivých miest. Predpokladajte, že činnosti napísané v poznámke sa vykonávajú správne. Premenné a typy sú deklarované nasledovne:

```
type tmojePole= array[1..MaxN] of integer;
var destinacie: tmojePole; vstup: text; cm, cz, i: integer;

{vynuluj pole destinacie}
{otvor subor vstup na citanie}
while not eof(vstup) do begin
  readln(vstup, cm, cz);
  destinacie[ {1} ]:= {2} ;
end;
```

Napíšte časti algoritmu, ktoré chýbajú na miestach {1} a {2}.

- 54** Predpokladajme, že v každom prvku *destinacie[i]* ($0 < i \leq N$) je zapísaný počet zásielok, ktoré treba dopraviť do mesta *i*. Zásielky sa posielajú z mesta 1. Úlohou nasledujúceho programu je vypísať mestá v poradí, v akom ich doručovateľ firmy navštívi. Pri výbere nasledujúceho mesta na trase treba zvoliť najbližšie z tých, do ktorých ešte treba doručiť zásielku.

```
{var zmesta, domesta: integer; pokracuj: boolean;}
pokracuj:= true;
zmesta:= 1; writeln('z mesta 1'); destinacie[1]:= 0
while pokracuj do begin
  domesta:= minvzdialene(N, zmesta, mapa, destinacie);
  if domesta = 0 then pokracuj:= false else
  begin
    writeln(domesta, ': ', mapa[zmesta, domesta], 'km');
    destinacie[domesta]:= 0;
    {1}
  end;
end;
```

Funkcia *minvzdialene* vráti číslo najbližšieho mesta k mestu daného druhým parametrom, do ktorého treba doručiť zásielku. Počet zásielok, ktoré treba ešte doručiť do jednotlivých miest, je zapísaný v poli danom štvrtým parametrom, počet miest je daný prvým parametrom a vzdialenosti medzi mestami sú v treťom parametri.

Aký príkaz je potrebné dopísať na miesto {1}, aby uvedený algoritmus správne fungoval?

- 55** Akú časť programu treba dopísať na miesta {1} a {2} funkcie *minvzdialene*, keď vieme, že všetky cesty v poli *m* sú kratšie ako 1 000 km?

```
function minvzdialene(N, z: integer; m: TmojaMapa; d:
TmojePole):integer;
var min, mesto, i: integer;
begin
  min:= 1000; mesto:= 0;
  for i:= 1 to N do
    if (d[i] <> 0) and {1} then begin
      min:= m[ {2} ]; mesto:= i;
    end;
  minvzdialene:= mesto;
end;
```

Text k úlohám 56 – 60

Dátová štruktúra rad (queue) má stratégiu pridávania a výberu prvkov FIFO (čo znamená prvý dnu – prvý von), a zásobník (stack) je známy stratégiou LIFO (čo znamená posledný dnu – prvý von). Pre oba typy máme definované štyri operácie. V jednom programe možno použiť súčasne viacej štruktúr, preto má každá svoje číslo. Do oboch štruktúr možno ukladať celé čísla.

Operácie:

<i>vyprázdni:</i>	<i>procedure vyprazdni(cislostruc: integer);</i>
<i>vloz hodnotu h:</i>	<i>procedure vloz(cislostruc, h: integer);</i>
<i>vyber:</i>	<i>function vyber(cislostruc: integer): integer;</i>
<i>vráť pravdu, práve vtedy, keď je prázdna</i>	<i>function prazdny(cislostruc:integer): boolean;</i>

Algoritmus Pracuj k úlohám 56 a 57

```
{var i: integer;}

vyprazdni(1);
for i:= 1 to 5 do vloz(1, i);
write(vyber(1), vyber(1), vyber(1));
vloz(1, 7); vloz(1, 8);
write(vyber(1), vyber(1));
```

- 56** Čo vypíše algoritmus *Pracuj*, ak použitá štruktúra je zásobník?

- 57** Čo vypíše algoritmus *Pracuj*, keď použitá štruktúra bude rad?

- 58** Čo vypíše nasledujúci program, keď štruktúra číslo 1 je zásobník a štruktúra číslo 2 je rad?

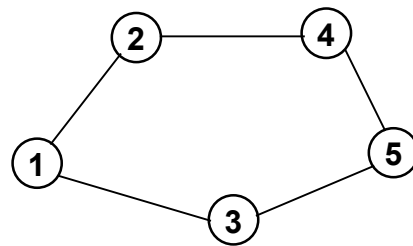
```
{var i: integer;}
vyprazdni(1); vyprazdni(2);
for i:= 1 to 3 do vloz(1, i);
for i:= 3 to 6 do vloz(2, i);
i:= 1;
While not prazdny(1) and not prazdny(2) do begin
  vloz(i, vyber(1) + vyber(2));
  if i = 1 then i:= 2 else i:= 1;
end;
if not prazdny(1) then write(vyber(1));
if not prazdny(2) then write(vyber(2));
```

Algoritmus Cesty k úlohám 59 a 60

```
{mapa[1..N, 1..N] of integer; pokračuj: boolean;
start, ciel, zobce, doobce: integer;}
```

```
vyprazdni(1);
vloz(1, start);
pokracuj:= true;
while not prazdny(1) and pokračuj do begin
  zobce:= vyber(1);
  doobce:= 1;
  while (doobce <= N) and pokračuj do begin
    if (mapa[zobce, doobce] > 0) then begin
      write(zobce, doobce);
      if doobce = ciel then pokračuj:= false
      else begin
        vloz(1, doobce);
        mapa[zobce, doobce]:= -1; mapa[doobce, zobce]:= -1;
      end;
    end;
    doobce:= doobce + 1;
  end;
end;
end;
```

mapa	1	2	3	4	5
1	0	1	1	0	0
2	1	0	0	1	0
3	1	0	0	0	1
4	0	1	0	0	1
5	0	0	1	1	0



59 Predpokladajme, že dátová štruktúra 1 je zásobník. Čo vypíše algoritmus Cesty pre mapu z obrázku, kde $N = 5$, $start = 1$, $ciel = 4$?

60 Predpokladajme, že dátová štruktúra 1 je rad. Čo vypíše algoritmus Cesty pre mapu z obrázku, kde $N = 5$, $start = 1$, $ciel = 4$?

Koniec testu.