

Korešpondenčný seminár z programovania XXI. ročník, 2003/04

Katedra vyučovania informatiky FMFI UK,
Mlynská Dolina, 842 48 Bratislava

*KSP finančne podporujú: aSc – Applied Software Consultants spol. s r.o.
MICROSTEP-MIS spol. s r.o.*

Vzorové riešenia 1. kola letnej časti, kat. Z

Milí riešitelia!

Jarné sústreďenie máme úspešne za sebou a vám sa dostávajú do rúk vzorové riešenia prvého kola. Pozorne si ich prečítajte, aby ste vedeli ako sa tie úlohy mali riešiť. Ešte nie je nič stratené. Do jesenného sústreďenia je ešte veľa času a jedno kolo KSP. Poučený týmito riešeniami sa ešte každý z vás môže na sústredko dostať.

Ráno býva múdrejšie večerať.

KSPáci

1. Zozolvár najväčší v okolí

opravovala Monika
(max. 15 bodov)

Na začiatok si treba uvedomiť, že v každom rade s N bohatiermi s rôznymi výškami musí existovať aspoň jeden, ktorý sa hrozivo týči (teda je vyšší od oboch susedov, resp. ak je na kraji, tak je vyšší od svojho suseda). Očíslujme si bohatierov od 1 po N , tak ako stoja v rade. Začnime z kraja: ak krajný (prvý) bohatier sa hrozivo netýči, znamená to, že jeho sused, teda bohatier číslo 2 je vyšší. Bohatier číslo 3 má dve možnosti buď je nižší ako bohatier 2, vtedy sa bohatier číslo 2 hrozivo týči, alebo je vyšší a tak pokračujeme v rovnakej úvahe ďalej. Ak nenájdeme žiadneho hrozivo sa týčiaceho a sme na konci, znamená to, že máme vzostupne usporiadaný rad, teda, že bohatier číslo N je vyšší ako bohatier číslo $N - 1$ a teda bohatier N sa hrozivo týči.

Pri vyhľadávaní hrozivo sa týčiaceho bohatiera netreba kontrolovať všetkých. Stačí určiť interval, v ktorom sa bude nachádzať aspoň jeden hrozivo sa týčiaci. Tento interval budeme zmenšovať takto: Na začiatku máme začiatok daný 1. bohatierom a koniec daný N -tým bohatierom. Vezmeme si stredného bohatiera (nech je S -tý v rade) označme ho A a toho, čo je hneď vedľa neho (to je $S + 1$ -vý bohatier) a označme ho B . Ak je bohatier A vyšší ako bohatier B , znamená to, že v intervale od 1. bohatiera po bohatiera A (vrátane) sa musí nachádzať aspoň jeden, ktorý sa hrozivo týči¹. Ak je bohatier A nižší ako bohatier B , znamená to, že v intervale od bohatiera B po bohatiera číslo N sa musí nachádzať aspoň jeden, ktorý sa hrozivo týči. Tým sme zmenšili interval, v ktorom sa nachádza aspoň jeden bohatier, ktorý sa hrozivo týči. Rovnakým postupom pokračujeme v tomto menšom intervale až kým nezostane interval s jedným bohatier (začiatok a koniec je v tom istom mieste). Keďže sme stále zmenšovali interval, v ktorom sa nachádza hrozivo sa týčiaci bohatier a zmenšili sme ho na jedného človeka, tak tento človek bude hrozivo sa týčiaci.

Časová zložitosť je logaritmická od počtu bohatierov ($O(\log N)$), pretože každý interval delíme na dve rovnaké časti a pracujeme len s jednou polovicou ako s novým intervalom. Pamäťová zložitosť je konštantná ($O(1)$), pretože využívame len konštantný počet premených nezávislých od počtu bohatierov.

Hodnotenie:

- riešenie v časovej zložitosti $O(\log N)$ 15 – 12 bodov

¹Rozmyslite si to

- riešenie v časovej zložitosti $O(N)$ 10 – 5 bodov
- za popis som strhávala maximálne 3 body
- za použitie poľa som strhávala maximálne 2 bod (pokiaľ sa používalo na zapamätanie si výšok bohatierov a pokiaľ sa aj využili tieto informácie)
- za chýbajúci odhad zložitosti som strhávala 1 bod

Listing programu:

```
program Zozolvar;
var N:integer;
    zaciatok, koniec: integer;
    stred,vyska_stred,vyska_vedla :integer;
begin
    writeln('Zadaj pocet N:');
    read(N);
    zaciatok:=1;
    koniec:=N;
    while (zaciatok < koniec) do
        begin
            stred:=(zaciatok+koniec) div 2;
            writeln('Zadaj vysku ',stred,'. bohatiera: ');
            read(vyska_stred);
            writeln('Zadaj vysku ',stred+1,'. bohatiera: ');
            read(vyska_vedla);
            if (vyska_stred > vyska_vedla) then
                koniec:= stred
            else
                zaciatok:= stred+1;
            end;
            writeln('Bohatier ',zaciatok,' sa hrozivo tyci.');
```

```
end.
```

opravovala Julka a YoYo δ
(max. 13 bodov)

2. Z krajiny Lemmingov

Riešenia tohto príkladu sa delili zväčša na 2 typy. Asi polovica porátala matiku za tým a tá druhá polovica buď binárne alebo jednoducho lineárne zistili, v akej výške sa za diaľnicu zaplatilo najmenej. A aké bolo bodovanie? Za vzorové riešenie v časovej zložitosti $O(n)$ bolo 13 bodov, za horšiu pamäťovú zložitost (dalo sa to aj v konštantnej) sme body nestrhávali. Za časovú zložitost $O(N \log H)$ bolo 10 bodov a za $O(NH)$ bolo 8 bodov. Za nefunkčné riešenia sme dávali maximálne 3 body. Z tejto hodnoty sa potom strhávali body za chýbajúci alebo nedostatočný popis a chýbajúci alebo chybný odhad zložitosti.

Vzorové riešenie: Vyjadrime najprv dĺžku mosta a tunelu

Z podobnosti trojuholníkov (napr. $a_1 a_2 a_3$ a $X a_2 Y$ (kde $|XY| = m_2$)) sa dá spočítať cena mostov, čiže

$$m_1 = \frac{x \cdot (a_1 - a_0)}{H}, m_2 = \frac{x \cdot (a_3 - a_1)}{H}, m_3 = \frac{x \cdot (a_5 - a_3)}{H}.$$

Cena mostu je $m_i \cdot x$, čiže $\frac{x^2 \cdot (f_i)}{H}$, kde f_i je príslušné $a_j - a_k$. Čiže keď chceme spočítať celkovú cenu mostov, tak po vybratí pred zátvorku bude tá cena rovná

$$\frac{x^2}{H} \cdot (a_1 - a_0 + a_3 - a_1 + a_5 - a_3 + a_6 - a_5),$$

2.5cm

zázračne povyškráva a výjde $\frac{x^2}{H} \cdot (a_6 - a_0)$.

Pre tunely to už nevyjde tak pekne, pretože cena je závislá od druhej mocniny výšky, ale aj tak sa dá ľahko spočítať. Z podobnosti kopca a kopca nad tunelom bude $t_1 = (a_2 - a_0) \cdot \frac{H - x}{H}$.

Čiže celková cena tunelov je $\frac{(H-x)^2}{H^2} \cdot [(a_2 - a_0)^2 + (a_4 - a_2)^2 + (a_6 - a_4)^2] = \frac{(H-x)^2}{H^2} \cdot D$. Celková cena mostov a tunelov je kvadratická funkcia závislá od x . Nech ešte $r = a_6 - a_0$.

$$f(x) = \frac{r}{H} \cdot x^2 + \frac{(H-x)^2}{H^2} \cdot D$$

Po úprave dostaneme

$$f(x) = \frac{D+rH}{H^2} \cdot x^2 - \frac{2D}{H} \cdot x + D$$

Kedže je to kvadratická funkcia, jej minimum je vo vrchole paraboly, v našom prípade tomu zodpovedá

$$x = \frac{DH}{D+Hr}$$

No a podobne sa dá postupovať pre ľubovoľne veľké n . Program je už iba vypočítanie tohto vzťahu. A aby sme to mali v konštantnej pamäťovej zložitosti, tak si nepamätáme všetky a_i , ale rovno počítame D a r .

Listing programu:

```
program lemming;
var n,H,r,D,i,a,b,c,a0:integer;
begin
  read(n,H);
  D:=0;
  read(c);
  a0:=c;
  for i:=2 to n+1 do begin
    read(b,a);
    D:=D+sqr(a-c);
    c:=a;
  end;
  r:=a-a0;
  if (D+H*r<>0) then writeln(D*H/(D+H*r)) else
    writeln('Mate divne kopce');
end.
```

opravoval Rastó, popis Goober
(max. 15 bodov)

3. Zeofína v záhrade

Tento príklad nebol vôbec ťažký. Bolo iba treba trochu porozmýšľať. Obsah všeobecného mnohouholníka sa počíta ťažko. Takže sa budeme snažiť vyrátať ho pomocou obsahov nejakých jednoduchších útvarov. No a čo je jednoduchšie ako trojuholník?

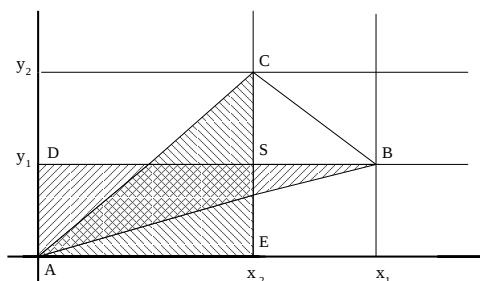
Najprv sa teraz pozrime, ako sa dá počítať obsah trojuholníka. A nie len takého hocijkého, ale takého, u ktorého vieme len súradnice jeho troch vrcholov. Určite sa nič nestane, keď si ten trojuholník posunieme tak, že bude mať jeden vrchol v počiatku súradnicovej sústavy (t.j. jeho súradnice budú $A = [0, 0]$). Súradnice zvyšných dvoch vrcholov potom budú $B = [x_1, y_1]$ a $C = [x_2, y_2]$. Keď si to nakreslíte a trochu si započítate, zistíte, že obsah takéhoto trojuholníka je niekedy $\frac{1}{2}(x_1y_2 - x_2y_1)$ a inokedy zase $\frac{1}{2}(x_2y_1 - x_1y_2)$. Pozrime sa na obrázok. Skúsme spočítať obsah trojuholníka ABC .

$$S_{ABC} = S_{BSC} + S_{ADB} + S_{AEC} - S_{AESD}$$

Teraz si to prepíšme do súradníc:

$$S_{ABC} = \frac{1}{2}(x_1 - x_2)(y_2 - y_1) + \frac{1}{2}x_1y_1 + \frac{1}{2}x_2y_2 - y_1x_2 = \frac{1}{2}(x_1y_2 - x_2y_1)$$

Ak by sme vymenili polohu bodov B a C , dostali by sme výraz $\frac{1}{2}(x_2y_1 - x_1y_2)$.²



3.5cm

Teda skutočný obsah trojuholníka je absolútnou hodnotou jeho orientovaného obsahu.

Teraz pojem orientovaného obsahu rozšírime na všetky mnohoúhelníky. Vezmime si nejaký mnohoúhelník a zvolíme si úplne ľubovoľný bod X . Na začiatku si povieme, že orientovaný obsah je rovný nule. Postupne budeme prechádzať strany mnohoúhelníka (v jednom, pevne zvolenom, smere). Vždy, keď prejdeme stranu YZ (čiže ideme z vrcholu Y do vrcholu Z a nie naopak!), tak si k nemu pripočítame orientovaný obsah trojuholníka XYZ . Keď prejdeme všetky strany, vyhlásime to, čo nám vyšlo, za orientovaný obsah celého mnohoúhelníka. Okrem toho bude platiť, že jeho absolútna hodnota je rovná obsahu mnohoúhelníka.

Táto definícia vyzerá určite veľmi pochybne. Najprv sa zamyslime nad tým, čo by sa stalo, keby sme si zvolili opačný smer. Nuž, vtedy by nám všetky orientované obsahy vychádzali s opačnými znamienkami, a teda výsledok by sa líšil iba v znamienku (čo by nám asi až tak nevadilo, keďže tak to bolo aj pri trojuholníkoch). Horšie by bolo, keby pri inej voľbe bodu X (ale pri tom istom smere prechádzania) vyšlo niečo iné. Našťastie, dokážeme, že takýto prípad nenastane.

Najprv treba ukázať, že takáto definícia funguje aj pre trojuholníky (keďže pre tie sme zavádzali orientovaný obsah inak a nebolo by dobre, keby nám mali vychádzať rôzne výsledky). Keďže je to veľmi ľahké, nechávame vám to na premyslenie (len malá rada: zide sa rozobrať tri prípady – bod X vnútri, na hranici a vonku z trojuholníka. Vo všetkých sa ukáže, že vyjde práve orientovaný obsah trojuholníka tak, ako sme ho definovali na skôr).

No a teraz si vezmime nejaký viac-ako-troj-uholník $V = A_1A_2 \dots A_n$. V takomto útvaru existuje aspoň jedna uhlopriečka (nech je to uhlopriečka A_1A_k), ktorá leží celá vnútri mnohoúhelníka, a teda ten mnohoúhelník môžeme pozdĺž nej rozstrihnúť na dva menšie – konkrétne $V_1 = A_1A_2 \dots A_k$ a $V_2 = A_kA_{k+1} \dots A_nA_1$. Teraz začneme počítať (podľa hore uvedenej definície) orientovaný obsah mnohoúhelníka V . Keď prechádzame strany poradí $A_1A_2, A_2A_3, \dots$, robíme to isté, ako by sme robili pri počítaní orientovaného obsahu V_1 . Takže keď pridáme po stranu $A_{k-1}A_k$, dostaneme takmer celý jeho orientovaný obsah. Ešte nám však chýba jeho strana A_kA_1 , ktorá je v mnohoúhelníku V uhlopriečkou. Teda doterajší zrátaný orientovaný obsah je rovný orientovanému obsahu mnohoúhelníka V_1 , až na to, že chýba orientovaný obsah XA_kA_1 . Nevadí! Pokračujme ďalej po stranách $A_kA_{k+1}, \dots$ až kým neprídeme po vrchol A_1 . Takto sme vlastne obišli takmer celý mnohoúhelník V_2 , akurát nám chýba strana A_1A_k . Teda do celkového orientovaného obsahu sme zarátali celý orientovaný obsah V_2 , až na trojuholník XA_1A_k . Lenže my vieme, že ak sme raz zabudli zarátať XA_kA_1 a raz XA_1A_k , tak sme vlastne na nič nezabudli, lebo súčet orientovaných obsahov týchto dvoch trojuholníkov je nulový (keďže majú rovnakú veľkosť, len opačné znamienko).

²Rozmyslite si ako to bude vyzeráť, ak by boli súradnice nejakého bodu záporné

Takže orientovaný obsah mnohouholníku V je rovný súčtu orientovaných obsahov jeho dvoch častí V_1 a V_2 . Keďže tieto časti majú menej vrcholov ako pôvodný mnohouholník, môžeme na ne použiť indukčný predpoklad, ktorý hovorí, že ich orientovaný obsah je nezávislý od voľby bodu X . Ale potom ani orientovaný obsah celého mnohouholníka V nezávisí od voľby X ! No a taktiež nie je ťažké vidieť, že absolútna hodnota tohto orientovaného obsahu je rovná súčtu absolútnych hodnôt orientovaných obsahov V_1 a V_2 . Tieto absolútne hodnoty sú však rovné skutočným obsahom mnohouholníkov V_1 a V_2 , a ich súčet je preto rovný obsahu mnohouholníku V . Presne to sme chceli dokázať.

Toto je aj priamo návodom na vzorové riešenie. Zvolíme si nejaký bod (napríklad aj prvý vrchol mnohouholníka) a prechádzame postupne po stranách mnohouholníka a zratúvame jeho orientovaný obsah. Na konci, keďže nás zaujíma skutočný obsah Zeofíninej záhradky, spravíme absolútnu hodnotu (ak by sme si náhodou zvolili taký smer, v ktorom by vyšiel orientovaný obsah záporný).

Časová zložitosť je $O(n)$, lebo každý vrchol patrí do práve dvoch trojuholníkov, a tie spracúvame každý práve raz. Pamäťová je $O(1)$, lebo nám stačí pamätať si aktuálny a posledný vrchol.

Body ste mohli stratiť za nedostatočný popis, zlý alebo žiadny odhad zložitosti. Za nefunkčné riešenia sa dalo získať maximálne 5 bodov.

Listing programu:

```
#include <stdio.h>
#include <math.h>

int main()
{
    float oldx,oldy,x=0,y=0,a,obsah=0;
    int uhol = 0;
    char cmd[80];

    while(fscanf(stdin,"%s %f",cmd,&a) == 2) {
        if(strncmp(cmd,"DOPREDU",7) == 0) {
            oldx=x; oldy=y;
            y+= sin((uhol*M_PI)/180.)*a;
            x+= cos((uhol*M_PI)/180.)*a;
            obsah += (x*oldy-y*oldx)/2;
        } else if(strncmp(cmd,"DOPRAVA",7) == 0) {
            uhol -= a;
            if(uhol < 0) uhol+= 360;
        } else if(strncmp(cmd,"DOLAVA",6) == 0) {
            uhol += a;
            if(uhol >= 360) uhol -= 360;
        }
    }
    printf("%.2f\n",fabs(obsah));
}
```

4. Zúfalo hladný Janko

Martin
(max. 15 bodov)

Väčšina z Vás riešila tento príklad backtrackom – postupne prešla všetky cesty, ktorými mohol Janko preskákať jahodové pole. Za takéto riešenie s časovou zložitou $O(4^{M+N})$ ste mohli získať najviac 7 bodov. Niektorí z Vás však vymysleli aj lepšie riešenie. Za riešenie s časovou

zložitostou $O(MN)$ ste mohli získať najviac 15 bodov a za riešenie s horšou polynomiálnou časovou zložitostou maximálne 13 bodov.

Vzorové riešenie bude používať metódu takzvaného dynamického programovania. Pre každé políčko jahodového poľa spočítame, koľko najviac jahôd by mohol Janko zožrať, ak by svoju cestu skončil v tomto políčku. Takýto počet jahôd pre políčko v r -tom riadku a s -tom stĺpci označme $p_{r,s}$. Počet jahôd na tomto políčku označíme $j_{r,s}$. Ľahko si všimneme, že ak sa do políčk v $(r-2)$ -hom riadku v $(s-1)$ -vom a v $(s+1)$ -vom stĺpci a v $(s-2)$ -hom stĺpci v $(r-1)$ -vom a $(r+1)$ -vom riadku vieme z ľavého horného políčka skakaním dostať, tak

$$p_{r,s} = j_{r,s} + \max(p_{r-2,s-1}, p_{r-2,s+1}, p_{r-1,s-2}, p_{r+1,s-2}).$$

S krajnými políčkami to je podobné, ibaže neuvažujeme políčka, na ktoré sa Janko nevie dostať.

Ako však nájsť hodnotu $p_{M,N}$? Môžeme ju priamočiaro rekurzívne odbacktrackovať, čo je však veľmi pomalé, alebo môžeme jednotlivé hodnoty $p_{r,s}$ počítať postupne po uhlopriečkach – všimnite si, že na výpočet hodnôt p v i -tej uhlopriečke stačí poznať hodnoty p v $(i-1)$ -vej a $(i-3)$ -tej uhlopriečke. Takto nám bude nájdenie hodnoty $p_{M,N}$ trvať $O(MN)$. Pamäťová zložitost' bude tiež $O(MN)$.

Pri riešení úlohy si treba ešte uvedomiť, že ak $M, N \geq 4$, tak pre Janka existuje aspoň jedna cesta, po ktorej sa dá z ľavého horného políčka dostať na pravé dolné políčko. Dôkaz tohto tvrdenia je pomerne jednoduchý, preto Vám ho nechávam ako domácu úlohu ;-).

Listing programu:

```
#include <stdio.h>

#define MAXM 100
#define MAXN 100

int janko[4][2] = {{-1, +2}, {+1, +2}, {+2, +1}, {+2, -1}};
/* Su 4 moznosti ako sa Janko moze pohnut */
int N, M, pf[MAXM][MAXN];
/* Pozn: skratka pf zamena 'play field';) */

int main(void)
{
    int i, u, r, s, max_s, r2, s2, max;

    scanf("%d %d ", &N, &M);
    for (r=0; r<M; r++)
        for (s=0; s<N; s++)
            scanf("%d ", &pf[r][s]);

    /* pole ma M+N-1 uhlopriecok, pricom na prvej (pre u=0) Janko zacina */
    for (u=1; u<M+N-1; u++) {
        if (u < M) {
            r = u; s = u;
        } else {
            r = M-1; s = u-M+1;
        }
        if (u < N) max_s = u;
        else max_s = N-1;

        while (s <= max_s) {
            max = -1;
            for (i=0; i<4; i++) { /* sem mohol skocit zo 4 stran */
                r2 = r - janko[i][0];
```

```

    s2 = s - janko[i][1];
    if (r2 >= 0 && r2 < M && s2 >= 0 && s2 < N)
    if (max < pf[r2][s2]) max = pf[r2][s2];
}
/* -1 znamená, že na policko sa neda doskackat */
if (max < 0) pf[r][s] = -1;
else pf[r][s] = pf[r][s] + max;
r--; s++;
}
}
printf("Janko zje najviac %d jahod.\n", pf[M-1][N-1]);
return 0;
}

```

5. Zo sveta steganografie

opravoval MišoF.
(max. 15 bodov)

Ako hovoril samotný text zadania, taká steganografická správa sa často môže tváriť ako niečo iné – napríklad bloček od nákupu, obrázok alebo trebárs zadania KSP. A naozaj, v týchto zadaniach KSP bolo skrytých správ neúrekom. Presnejšie štyri.

Začnime prvou a asi najvýraznejšou. V texte zadani boli niektoré písmená zvýraznené, keď ste ich všetky našli, mohli ste sa dozvedieť, že **heslo je hruška**.

Prvé písmená viet zadania „Zo sveta steganografie“ boli tiež zaujímavé – ukrývali oznam **akcia začína o polnoci**.

Tretou, ťažšou časťou bol náhodne vyzerajúci zhhluk písmen na konci zadania. Po chvíli pozorovania si môžete všimnúť, že sa v ňom približne opakujú vzorky dĺžky 16 znakov. V podstate stačilo vyznačiť znaky, kde nastala zmena... Ale. Skúsenejších už napadlo, že niečo podobné už videli niekde inde. A to pri 3D obrázkoch, najčastejšie volaných *stereogramy*.

V našom mozgu vzniká dojem priestoru tak, že sa obraz z ľavého a pravého oka líši. A to práve využívajú stereogramy na dosiahnutie dojmu, že obrázok je trojrozmerný. Princíp týchto obrázkov je taký, že keď sa zahľadíte nie priamo na obrázok, ale akoby **za obrázok**, pravé oko bude vnímať opakujúcu sa vzorku o jedno opakovanie posunutú. A drobnými zmenami vzorky vieme dosiahnuť, že nám niektoré jej časti „vystúpia dopredu“ a utvoria nejaký obrázok.

Ak ste sa ešte so stereogrammi nestretli (a máte prístup na internet), skúste si zadať do google slovo stereogram a preklikať niekoľko prvých liniek, na obrázku sa ich ľahšie „naučíte vidieť“ ako na texte z nášho zadania.

Stereogram zo zadania ukrýval číslo 47. Štyridsaťsedem je najčastejšie sa vyskytujúce náhodné číslo. Spája sa s ním celý kult, viac sa o ňom môžete dozvedieť napr. na stránke <http://www.47.net/47society/47story/>.

Poslednú správu ukrýval obrázok, na ktorý ukazovala linka na spodku strany so zadani (tam, kde je väčšinou adresa www.ksp.sk). Obrázok je stále tam. Na prvý pohľad sú na ňom stromy. Zadanie však hovorilo aj o tom, že aj v obrázku sa dá skryť nejaká iná správa. Napríklad stačí zobrať z každej farby každého políčka len posledné dva bity – obrázok sa nám prekvapivo zmení. Nebudem prezrádzať ako, pohrajte sa. (Najjednoduchšie je obrázok najskôr skonvertovať na 24bitovú bitmapu.)

Bodovanie bolo jednoduché – čím viac toho kto našiel a napísal, tým viac bodov dostal.

Výsledková listina po 1. kole kategórie KSP-Z

	Meno a priezvisko	Škola	Ročník	11	12	13	14	15	Σ
1	Takáč Slavomír	Gymnázium	2	14	13	15	15	4	61
2	Herman Peter	Gym. Jura Hronca BA	1	8	13	13	6	13	53
3	Mikuláš Ján	Gym. Haličská Lučenec	2	12	13	9	7	11	52
4	Ďuďák Juraj	Gym. Golianova Nitra	2	9		15	7	8	39
5	Rauová Ivana	Gym. Jura Hronca BA	3		13	14		8	35
6	Mikuláš Ondrej	Gym. Haličská Lučenec	1	6	12		7	9	34
7	Dobiaš Michal	Gym. Jura Hronca BA	2	10	10	4		8	32
8	Bundala Daniel	Gym. Jura Hronca BA	2	15	3	4	9		31
9	Dzurenko Miroslav	Gym. Ľ. Štúra Zvolen	3	7	8		7	8	30
10	Kajan Peter	Gym. Jura Hronca BA	1	3	11	1	5	8	28
11	Ambrošová Lucia	Gym. Grösslingová BA	2	10	10		6		26
11	Baumann Martin	Gym. Jura Hronca BA	1	3	12			11	26
13	Fedák Matúš	Gym. Stará Ľubovňa	2	10	6		6	3	25
14	Jerguš Ján	Gym. Alejová Košice	1	7			7	10	24
15	Paulis Peter	Gym. Cyrila a Metoda Nitra	3	3	8	4		8	23
16	Holub Michal	Gym. Jura Hronca BA	3	8	8	4	2		22
17	Sábo Jozef	Gym. Školská Spiš. Nová Ves	1	13		1	6	1	21
17	Strizenec Michal	Gym. Jura Hronca BA	3			13		8	21
19	Pančík Andrej	Gym. Tajovského B. Bystrica	1	12	7			1	20
20	Mindek Peter	Škola pre mim. nadané deti BA	2				7	12	19
20	Šimko Jakub	Gym. Jura Hronca BA	3	2		11		6	19
22	Mazák Tomáš	Gym. Jura Hronca BA	1		11			6	17
22	Zachar Lukáš	Gym. A.Sládkoviča B. Bystrica	3	5	6		6		17
24	Kováčovský Tomáš	Neznama škola		3	12				15
25	Danko Juraj	Gym. P. de Coubertina Piešťany	1	5	1	5	0	3	14
26	Bodnár Jozef	Gym. Fiľakovo	3		13				13
27	Šuranyi Martin	Gymnázium Levice	3	4				8	12
28	Sivák Michal	Gym. Ľudovíta Štúra Trenčín	2		8	3			11
29	Macák Michal	Gym. Giraltovce	4		10				10
29	Vanderka Peter	Gym. Trstená	3				10		10
31	Bálint Farkaš	Gym. maďarské Šahy	3	4	3	2			9
31	Valentíny Martin	Gym. Jura Hronca BA	3		9				9
33	Korcsok Peter	Gym. maďarské Šahy	0				7	1	8
33	Sivák Vladimír	Gym. Ľudovíta Štúra Trenčín	2		8				8
35	Žubrietovský Tomáš	Gym. Ľ. Štúra Zvolen	3	5	1			1	7
36	Kolesár Štefan	Gym. Alejová Košice	2	5					5
37	Balga Jozef	Gym. maďarské Šahy	4	4					4
37	Kompuš Patrik	Gym. Šrobárova Košice	1	4		0			4
37	Petrežsél Tibor	Gym. maďarské Šahy	4	4					4
40	Paľko Peter	SPŠE Michalovce	2	3					3
40	Pandoščák Michal	SPŠE Michalovce	2	3					3
42	Bredová Lucia	Gym. Šrobárova Košice	1			2.5			2
42	Hopková Dominika	Gym. Šrobárova Košice	1			2.5			2
42	Mondoková Denisa	Gym. Šrobárova Košice	1			2.5			2
42	Tokolyova Zuzana	Gym. Šrobárova Košice	1			2.5			2
46	Lizák Ivan	Gym. Jura Hronca BA	2	2					2
46	Panáč Pavol	Gym. Jura Hronca BA	2	2					2
48	Jacko Vladimír	Gym. Alejová Košice		1					1
48	Klimo Vladimír	Gym. Jura Hronca BA	2	1					1
48	Oremus Vladimír	Gym. Jura Hronca BA	2	1					1

	Meno a priezvisko	Škola	Ročník	11	12	13	14	15	Σ
48	Verčimák Viliam	SPŠE Michalovce	2	1	0		0		1
52	Nemjo Martin	SPŠE Michalovce	2			0			0