

## Општинско такмичење из информатике (анализа проблема са решењима)

Општинско такмичење из информатике (за школску 2012/2013. године) одржано је 23. фебруара 2013. Немамо податак колико такмичара је учествовало, али знамо да се на Округно такмичење пласирало 409 такмичара што је својеврсан рекордан број учесника за Србију.

Отприлике однос такмичара 2. и 1. категорије који су се пласирали на Округно такмичење је 2.8:1.

Дакле, ове године смо сведоци значајно већег интересовања за овај вид такмичења у односу на претходне године. Структура такмичења је остала иста као и прошлих година: такмичари су током 120 минута решавали три проблема.

Како нисмо били у прилици да анализирамо кодове такмичара за целу Србију, већ само са одређених места, то ће наша анализа такмичарских грешака бити делимична. Охрабрујемо колеге професоре да нам пошаљу фрагменте својих анализа на емаил [osnovciprogramiranje@gmail.com](mailto:osnovciprogramiranje@gmail.com)

Ова верзија билтена је радна и подложна изменама, а све напомене и предлози колега професора и такмичара су добродошле, јер сетимо се и овом приликом стихова нашег песника Јована Јовановића Змаја:

*"Где ја стадох - ти ћеш поћи!*

*Што не могох - ти ћеш моћи!*

*Куд ја нисам - ти ћеш доћи!*

*Што ја почех - ти продужи!*

*Још смо дужни - ти одужи!"*

И ове године, избор програмских језика је остао исти као и претходне године: BASIC, Pascal, C/C++, C#. Програмска окружења: QuickBasic, SmallBasic, VB.NET, Free Pascal, Dev Pascal, Delphi, Dev-C++, Code::Blocks, MS Visual Studio.

Подсећамо да ове године бирамо и националну екипу која ће учествовати на међународним такмичењима ЈБОИ 2013, Грчка, јул 2013. (Јуниорска Балканска Олимпијада из Информатике)

ЈИОИ 2013, Бугарска 8.-15. септембра 2013. (Јуниорска Интернационална Олимпијада из Информатике) Напомињемо да по пропозицијама на оба такмичења се користе само Pascal и C/C++.

Захваљући претходно добрим наступима, екипа Србије је позвана и на међународни турнир Шумен 2013 (програмски језик C/C++) и Master of Informatics 2014 (програмски језик C/C++).

Дакле, амбициознији такмичари морају бити свесни текућег тренда у избору програмског језика на међународним такмичењима у јуниорској категорији, те ћемо и ми без смањења општости решења приказати у форми C/C++ кода.

### Проблем 1, категорија 1:

Институт за шумарство је констатовао да је стање шума и шумског земљишта подручја Србије неповољно са више аспеката (производних, квалитативних и структурних) и да су функције шума редуковане нерационалним газдовањем у прошлости. У жељи да обнови шумски фонд, шумско газдинство *Оморика* направило је стручни план стратегије пошумљавања. У оквиру заштите и унапређивања шумских екосистема спроводе се мере обнављања шумских ресурса. Стратегија предвиђа да се свака правоугаона шумска површина ширине  $L$  и дужине  $H$  огради тако што се полазећи од једног темена постави стабло на сваких  $S$  метара по ивицама шуме или паралелно њима унутар шуме. Написати програм POSUMLJAVANJE којим се унесе цели бројеви  $L$ ,  $H$ ,  $S$  и израчунава колико стабала шумско газдинство *Оморика* треба да постави тако да на описан начин огради максималну површину шуме.

### ПРИМЕР

**УЛАЗ**

10 5 3

**ИЗЛАЗ**

8

Решење:

```
#include <stdio.h>
int main()
{
    int l,h,s,stabla;
```

```
scanf("%d%d%d", &l, &h, &s);
stabl= 2 * (l / s + h / s);
printf("%d", stabla);
return 0;
}
```

Грешке такмичара: У формулацији задатка је јасно речено да су улазни подаци цели бројеви. Но, неки такмичари су користили реалне бројеве приликом уноса и као резултат операције дељења и множења добили број стабала у формату реалног броја.

### Проблем 2, категорија 1:

Програмер Васа жели да помогне својој малој сестри да провежба сабирање и множење једноцифрених бројева. Сваки пут када Васа напише један троцифрен број, његова сестра мора да између сваке две узастопне цифре постави знак за сабирање или знак за множење и израчуна добијени израз. На крају мора да саопшти Ваци највећи резултат који је добила. Ваш задатак је да напишете програм MAXIZRAZ и помогнете Ваци да брзо и тачно прегледа сестрине задатке. Ваш програм треба да учита један троцифрен број и испише највећу вредност која се добије ако се између сваке две узастопне цифре броја постави или знак за сабирање или знак за множење.

#### ПРИМЕР

**УЛАЗ**

1 2 8

**ИЗЛАЗ**

1 7

Анализа проблема: Улазни троцифрен број се може унети као број (ниска) са три цифре без употребе бланко карактера и његове цифре се могу чувати као вредност три променљиве  $x$ ,  $y$ ,  $z$  или као нумерички податак када треба пажљиво рашчланити вредности његових цифара. Потом је потребно формирати могуће изразе уметањем оператора  $+$   $*$  између цифара и одредити максимални израз од могућа 4 изрази:  $x+y+z$ ,  $x+y*z$ ,  $x*y+z$ ,  $x*y*z$ .

Решење:

```
#include<iostream>
using namespace std;
```

```
int main()
{
    char a,b,c;
    cin >> a >> b >> c;
    int x=a-'0', y=b-'0', z=c-'0';
    int v = x+y+z;
    if(x+y*z>v) v=x+y*z;
    if(x*y+z>v) v=x*y+z;
    if(x*y*z>v) v=x*y*z;
    cout << v << endl;
    return 0;
}
```

Грешке такмичара: Неки такмичари нису провежбали задатке у којима се одређује максимум три задата цела броја, те су овом приликом само исписивали резултат тест примера 17 очекујући бар утешне поене (што се није десило). Иначе, у овом задатку уопште није било потребно користити низове.

### Проблем 3, категорија 1:

Српски тим и ове године учествује на Светском шампионату у модерном плесу. Оцена такмичара се формира на основу збира оцена свих чланова жирија из ког су изузете најмања и највиша оцена. Жири се састоји од шест чланова. Написати програм PLES којим се уносе целобројне ненегативне оцене чланова жирија за један национални тим и који израчунава и исписује укупну оцену према горе наведеном правилу. Ако сви чланови жирија дају исту оцену, коначна оцена се формира као сума свих оцена. Највећа оцена коју може дати један члан жирија није већа од 20.

#### ПРИМЕР

Анализа проблема:

Решимо задатак методом лажне претпоставке тако што ћемо иницијализовати променљиву *min* на 21, јер максимална оцена жирија има вредност 20. Иницијализујмо променљиву *max* са 0, јер је то најнижа могућа оцена. Сваку од учитаних оцена жирија ћемо памтити у некој од шест променљивих  $a_1..a_6$ . Редом за сваку оцену проверимо да ли је већа од *max* оцене или мања од *min* оцене.

Сумирајмо оцене у променљивој *s*.

Ако су оцене једнаке, испишемо суму свих оцена. Иначе, испишемо суму свих оцена *s* умањену за оцене *min* и *max*.

Решење:

```
#include <iostream>
using namespace std;
int main()
{ int min=21, max=0;
  int a1,a2,a3,a4,a5,a6;
  cin>>a1>>a2>>a3>>a4>>a5>>a6;
  if(a1<min) min = a1;
  if(a1>max) max = a1;
  if(a2<min) min = a2;
  if(a2>max) max = a2;
  if(a3<min) min = a3;
  if(a3>max) max = a3;
  if(a4<min) min = a4;
  if(a4>max) max = a4;
  if(a5<min) min = a5;
  if(a5>max) max = a5;
  if(a6<min) min = a6;
  if(a6>max) max = a6;
  int s=a1+a2+a3+a4+a5+a6;
  if(a1==a2&&a2==a3&&a3==a4&&a4==a5&&a5==a6) cout<<s<<endl;
  else cout<< s - min - max <<endl;
}
```

Грешке такмичара: Неки такмичари су заборавили да у имплементацији изврше проверу да ли су све оцене једнаке.

### Проблем 1, категорија 2:

Институт за шумарство је констатовао да је стање шума и шумског земљишта подручја Србије неповољно са више аспеката (производних, квалитативних и структурних) и да су функције шума редуковане нерационалним газдовањем у прошлости. У жељи да обнови шумски фонд, шумско газдинство *Оморика* направило је стручни план стратегије пошумљавања. У оквиру заштите и унапређивања шумских екосистема спроводе се мере обнављања шумских ресурса. Стратегија предвиђа да се свака правоугаона шумска површина ширине *L* и дужине *H* ограда тако што се полазећи од једног темена постави стабло на сваких *S* метара по ивицама шуме или паралелно њима унутар шуме тако да се на описан начин ограда максимална површина шуме. Написати програм POSUMLJAVANJE којим се уносе цели бројеви *L*, *H*, *S* и израчунава колико процената површине шуме није ограђено. Проценте штампати са тачношћу на две децимале.

### ПРИМЕР

УЛАЗ

10 5 3

ИЗЛАЗ

46.00

Решење:

```
#include <stdio.h>
int main()
{
```

```

int l,h,s,neogradjeno,povrsina;
float procenat;
scanf("%d%d%d", &l, &h, &s);
povrsina=l*h;
neogradjeno= povrsina - (l / s) * s * (h / s) * s; //povrsina-ogradjen deo
procenat= (float)neogradjeno*100/povrsina;
printf("%.2f", procenat);
return 0;
}

```

Грешке такмичара: Многи такмичари нису обезбедили испис излазног резултата на две децимале. Такође, неки такмичари нису уопште ни имплементирали форматирани испис реалног излазног резултата.

## Проблем 2, категорија 2:

Погледати Проблем 3, категорија 1

## Проблем 3, категорија 2:

Сваке године ученици имају прилику да у школским свечаним салама присуствују прослави школске славе. Разредне старешине су одлучиле да распореде ученике према прозивнику на следећи начин: најпре се у сали попуњава први ред у смеру лево надесно, потом на исти начин се попуњава други ред и тако редом док се не попуни цела сала која има  $n$  редова са по  $m$  места за седење у сваком реду. Међутим, кад је дошао директор школе, образложио је разредним старешинама зашто такав распоред седења није коректан и предложио је прераспodelу седења на следећи начин: у сваком реду (од првог до последњег) ученици треба најпре да попуне сва прва места, потом сва друга места и тако даље. Напишите програм PROSLAVA, којим се уносе цели бројеви  $n$ ,  $m$  ( $1 < m < 3 \cdot 10^4$ ,  $1 < n < 3 \cdot 10^4$ ) и који израчунава и исписује колико ученика у прераспodelи ће остати на својим првобитним местима. Сматрајте да школа има довољно ученика да попуни целу салу.

### ПРИМЕР 1

| УЛАЗ | ИЗЛАЗ |
|------|-------|
| 3 3  | 3     |

### ПРИМЕР 2

| УЛАЗ | ИЗЛАЗ |
|------|-------|
| 2 4  | 2     |

Објашњење за први пример: ако  $n = 3$  и  $m = 3$ , првобитни распоред ученика је

```

1 2 3
4 5 6
7 8 9

```

Након прераспodelе, распоред ученика је:

```

1 4 7
2 5 8
3 6 9

```

Ученици с редним бројевима 1, 5 и 9 ће остати на својим местима.

Анализа проблема:

Седиште се не мења ако је редни број колоне расподеле једнак редном броју колоне прераспodelе и ако аналогно важи за врсте.

```

#include <iostream>
using namespace std;
int main()
{
    long int n, m; // vrsta, kolona
    cin>>n>>m;
    long int br = 0;
    for (int sediste=0;sediste<n*m;sediste++)
        if (sediste / n == sediste % m && sediste / m == sediste % n)
            ++br;
}

```

```
//sediste se ne menja ako je redni broj kolone raspodele jednak rednom broju kolone preraspodele
//i ako analogno vazi za vrste
```

```
cout<<br<<endl;
return 0;
```

```
}
```

Решење:

```
#include <iostream>
```

```
using namespace std;
```

```
int nzd(int a,int b)
```

```
{
```

```
while (b)
```

```
{ int r = a % b;
```

```
  a = b;
```

```
  b = r;
```

```
}
```

```
return a;
```

```
}
```

```
int main()
```

```
{ int n, m;
```

```
cin>>n>>m;
```

```
cout<<1+nzd(n - 1, m - 1) << endl;
```

```
return 0;
```

```
}
```

Грешке такмичара: У овом задатку није било потребно користити низове (ни једнодимензионе ни дводимензионе низове). Неки такмичари су алоцирали превише меморије декларишући матрицу димензије 30000\*30000 и услед тога им програм није радио. Такође, нека решења грубом силом нису могла да дају коректан резултат у времену од 10 секунди. Било је и такмичара који су добили утешне поена на тест примерима за квадратну матрицу, тј. за случај  $n = m$