

**Државно такмичење из програмирања, Београд – 2. април 2016.**  
**II категорија (7. и 8. разред)**

У сваком задатку временско ограничење је 1 секунда, а меморијско ограничење је 64 MB.

1. Дата су два цела броја  $a, b$  ( $0 < a < b < 1\ 000\ 000\ 000\ 000\ 000$ ) и цео број  $c$  ( $1 < c < 1\ 000\ 000\ 000\ 000\ 000$ ). Написати програм (конзолну апликацију) **DELIIVI** која ће исписати колико целих бројева између  $a$  и  $b$  (укључујући и  $a$  и  $b$ ) је дељиво датим бројем  $c$ .

**УЛАЗ:** У првом реду стандардног улаза дат је природан број  $a$ , у другом реду стандардног улаза дат је природан број  $b$ , у трећем реду стандардног улаза дат је природан број  $c$ .

**ИЗЛАЗ:** У једином реду стандардног излаза исписати тражени број садржалаца.

**ПРИМЕР**

УЛАЗ	ИЗЛАЗ	УЛАЗ	ИЗЛАЗ	УЛАЗ	ИЗЛАЗ
8	7	8	5	8	154320986
40		40		1234567890	
5		8		8	

Решење

Када не би било постављено дозвољено време за рад програма, онда једно од решења би било испробати проверу дељивости бројем  $C$  за све бројеве између  $A$  и  $B$ . Али, број тих провера може бити веома велик за све бројеве чија вредност не прелази десетине милиона.

Да би се решио овај проблем са мањим бројем провера и тиме задовољило временско ограничење, примећујемо да бројеви између  $A$  и  $B$ , који су дељиви бројем  $C$ , се јављају у редовним интервалима, од којих сваки садржи скуп узастопних  $C$  бројева.

Колико има таквих интервала? Број тих интервала одговара броју садржалаца броја  $C$  који су између бројева  $A$  и  $B$ .

Остаје још да се пажљиво нађе најмањи и највећи садржалац броја  $C$  који су између бројева  $A$  и  $B$  односно бројеви  $a1$  и  $b1$ , те је укупан резултат једнак  $1+(b1-a1)/c$

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

```
long long int a,b,c;
```

```
int main()
{
    cin >> a >> b >> c;
    long long int a1=a;
    if(a%c!=0) a1=a+c-a%c; // najmanji sadrzalac od c, ne manji od broja a
    long long int b1=b-b%c; // najveci sadrzalac broja c, ne veci od broja b
    long long int r=0;
    if(a1<=b1) r = 1+(b1-a1)/c; //formula za prebrojavanje
    cout << r << endl;
}
```

2. Напишите програм (конзолну апликацију) **KVADJED** који за дати број налази производ који се добија када се дати број помножи самим собом. Дати број се представља са  $n$  ( $1 \leq n \leq 900000$ ) јединица.

**Улаз:** У једином реду стандардног улаза дат је природан број  $n$ .

**Излаз:** У једином реду стандардног излаза исписати тражени производ.

**ПРИМЕР**

УЛАЗ	ИЗЛАЗ	УЛАЗ	ИЗЛАЗ	УЛАЗ	ИЗЛАЗ
1	1	2	121	9	12345678987654321

Објашњење 2. примера:  $11 * 11 = 121$

Објашњење 3. примера:  $111\ 111\ 111 * 111\ 111\ 111 = 12345678987654321$

Решење:

Уочимо за  $n < 10$

$1 * 1 = 1$

$11 * 11 = 121$

$111 * 111 = 12321$

$1111 * 1111 = 1234321$

$11111 * 11111 = 123454321$

$111111 * 111111 = 12345654321$

```
#include <cstdio>
```

```
using namespace std;
```

```
int main (){
```

```
    int r=0, n;
```

```
    scanf("%d",&n);
```

```
    for (int i=1;i<=2*n;i++)
```

```
    {
```

```
        r*=10;
```

```
        if (i<n)r+=9;
```

```
        else if (i==n)r+=8;
```

```
            else if (i==2*n)r+=1;
```

```
        if (i>1)printf("%c",48+r/81);
```

```
        r%=81;
```

```
    }
```

```
    printf("\n");
```

```
    return 0;
```

```
}
```

3. Дат је низ природних бројева  $X$  који се састоји од  $N$  бројева. Сваки број у низу је цео број између бројева 1 и  $N$  (укључујући и 1 и  $N$ ), односно свака вредност се појављује тачно једном у низу.

Кажемо да подниз низа  $X$  је сваки низ који добијамо уклањањем нула или више бројева с почетка низа  $X$ , и потом уклањањем нула или више бројева с краја низа  $X$ . Написати програм (конзолну апликацију) **PODSRED** који ће исписати колико има различитих поднизова низа  $X$  непарне дужине којима је средњи елемент (медијана) једнак датом броју  $M$ . Медијана или средњи елемент неког низа је елемент који се налази на половини низа након његовог сортирања.

**Улаз:** У првом реду стандардног улаза се налазе два природна броја  $N$  ( $1 \leq N \leq 100\ 000$ ) и  $M$  ( $1 \leq M \leq N$ ).

У другом реду стандардног улаза се налази се  $N$  природних бројева, елемената низа  $X$ , раздвојених с по једним бланко карактером.

**Излаз:** У једином реду стандардног излаза исписати један број, тражени број различитих поднизова непарне дужине којима је средњи елемент једнак броју  $M$ .

**Примери**

Улаз	Улаз	Улаз
6 4	7 4	7 4
1 2 3 4 5 6	7 5 3 1 2 4 6	5 7 2 4 3 1 6
Излаз	Излаз	Излаз
3	3	4

Објашњење 1. примера: три подниза којима је средњи елемент једнак броју 4 су 4, 3 4 5, 2 3 4 5 6.

Објашњење 2. примера: три подниза којима је средњи елемент једнак броју 4 су 4, 2 4 6, 7 5 3 1 2 4 6.

Објашњење 3. примера: четири подниза којима је средњи елемент једнак броју 4 су 4, 7 2 4, 5 7 2 4 3, 5 7 2 4 3 1 6

Решење:

Ако је медијана неког подниза једнака  $M$ , онда у том поднизу најпре као члан се јавља број  $M$ .

Нас ће за сваки подниз интересовати разлика броја чланова који су већи од  $M$  и броја чланова који су мањи од  $M$ . Означимо ту функцију са  $A(X)$ . Важи да подниз има медијану једнаку  $M$  ако и само ако он садржи

број  $M$  и ако је разлика броја чланова који су већи од  $M$  и броја чланова који су мањи од  $M$  једнака  $0$ .

Дакле, ако подниз садржи број  $M$ , и ако га раставимо на два подниза – подниз лево од броја  $M$  и подниз десно од броја  $M$ , тада подниз има средњи елемент једнак  $M$  ако је  $A(X_{\text{лево}}) + A(X_{\text{десно}}) = 0$ .

Дакле, идеја решења је да за све поднизове који се завршавају бројем  $M$  се израчуна функцију  $A$ , и чувамо колико има поднизова за које је вредност те функције једнака броју  $x$ .

Симетрично, за све поднизове који почињу бројем  $M$  рачунамо функцију  $A$ , и чувамо колико има поднизова за које је вредност функције  $A$  једнака тачно  $x$ .

Укупном збиру додајемо број поднизова за сваку вредност растојања  $x$  ( $-N \leq x \leq N$ ) односно  $A(X_{\text{лево}}) * A(X_{\text{десно}})$ .

#### Корисне напомене у вези прегледања задатака

- Оставите себи времена како бисте проверили да ли сте у кореном директоријуму креирали датотеке чији назив је исти као и назив задатка у формулацији коју сте добили (нпр. NIZN,...)
- У том фолдеру се памте искључиво .pas, .c .cpp,... изворни (source) кодови чија имена морају бити као у формулацији задатка.
- За решавање задатака, такмичари могу да користе програмске језике C, C++, Pascal, Basic, C#.
- Такмичари обавезно креирају конзолне апликације због аутоматског прегледа задатака (C, C++, Pascal, C# тј. FreePascal, gcc, g++ компајлер) и полуаутоматског прегледа задатака (Basic).
- Подаци се читају/исписују преко стандардног улаза и излаза - немојте користити додатне датотеке!
- Излазни подаци морају бити тачно у облику датим у опису задатка. Немојте исписивати додатне поруке “Решење је...”.
- На крају програма обавезно уклонити наредбе које неки од Вас користе како би задржали излазни екран са исписаним резултатом:

Pascal	C/C++	C#
readln;	for(;;); system("pause") getchar(); getch();	Console.ReadLine(); Console.Read(); Console.ReadKey(); for (;;);

- Уколико је потребно користити 64-битне бројеве, користите int64 у Pascal-у, односно long long у C/C++-у; обратите пажњу да long у C/C++-у не мора увек бити 64-битни тип. Уколико за учитавање/испис 64-битних бројева у C/C++-у користите функције scanf/printf, потребно је употребити спецификатор %lld.
- У C/C++ кодовима, користити <iostream> а не <iostream.h>. Такође, морате експлицитно include-овати све библиотеке чије функције користите (нпр. <cstring>, <cstdlib>, <algorithm>).

У неким окружењима (DevC++) ваш код ће радити и без тога али не и на званичном систему! Слично је и са укључивањем неких библиотека из алата Microsoft Visual C++ (нпр. <stdafx.h>,...) које нису подржане у званичном компајлеру.

- У C/C++ кодовима функција main мора бити декларисана као “int main()” а не као “void main()” или “main()”. Такође, ова функција мора враћати вредност, тј. морате имати “return 0;”

ПОНЕДЕЉАК 04.04.2016.

До 15ч – привремена ранг листа на сајту Рачунарске гимназије и Друштва математичара Србије

ПЕТАК 08.04.2016.

До 15ч – подношење приговора на резултате привремене ранг листа е-поштом на адресу [takmicenjeinf@gmail.com](mailto:takmicenjeinf@gmail.com)

Претходно се региструјте на платформи BubbleVee.org и само тестирајте број тест примера који Вам успешно пролазе. Слично, можете превући тест примере и са сајта Друштва математичара Србије и ручно проверити рад Ваших програма.

ПОНЕДЕЉАК 11.04.2016.

Објава коначних резултата на сајту Друштва математичара Србије