



| Udruženje nastavnika i profesora informatike | [www.eduit.ba](http://www.eduit.ba) |

**KANTONALNO TAKMIČENJE  
IZ INFORMATIKE ZA UČENIKE SREDNJIH ŠKOLA 2016**

*Sarajevo, maj 2016. godin*

## KONAČNI REZULTATI KANTONALNOG TAKMIČENJA IZ INFORMATIKE ZA UČENIKE SREDNJIH ŠKOLA 2016

Mjesto	Učenik	Škola	mrav	lubenice	enkripcija	zgrade	UKUPNO BODOVI
1.	Muamer Parić	Druga gimnazija	100.0	90.0	60.0	100.0	350
2.	Irfan Smajević *	Međunarodna srednja škola	100.0	100.0	0.0	100.0	300
3.	Fatih Zukorlić *	Tursko-bosanski sarajevo koledž	100.0	100.0	0.0	100.0	300
4.	Ali Dlakić	Druga gimnazija	100.0	90.0	0.0	80.0	270
5.	Faruk Alibašić	Međunarodna srednja škola	40.0	80.0	50.0	90.0	260
6.	Paško Zdilar *	Srednja elektrotehnička škola	100.0	90.0	0.0	60.0	250
7.	Haris Šoljić *	Druga gimnazija	90.0	80.0	30.0	50.0	250
8.	Adnan Brđanin	Druga gimnazija	40.0	90.0	0.0	90.0	220
9.	Adnan Šabanović	Druga gimnazija	100.0	40.0	0.0	70.0	210
10.	Haris Hanjalić	Međunarodna srednja škola	100.0	0.0	0.0	100.0	200
11.	Tarik Džaka	Međunarodna srednja škola	100.0	0.0	0.0	40.0	140
12.	Miron Banjac	Međunarodna srednja škola	20.0	20.0	0.0	100.0	140
13.	Adi Šoše	Prva gimnazija	0.0	30.0	0.0	100.0	130
14.	Armin Bilal	Četvrta gimnazija	40.0	0.0	0.0	60.0	100
15.	Damir Živojević	Međunarodna srednja škola	0.0	0.0	0.0	70.0	70
16.	David-Dragan Hadžić	Katolički školski centar	0.0	0.0	0.0	60.0	60
17.	Nikola Raič	Srednja elektrotehnička škola	40.0	0.0	0.0	10.0	50
18.	Edis Krsmanović	Srednja elektrotehnička škola	30.0	0.0	0.0	0.0	30
19.	Adrian Milišić	Gimnazija Obala	0.0	0.0	0.0	20.0	20
20.	Armin Teskeredžić	Gimnazija Dobrinja	0.0	0.0	0.0	10.0	10

Ostali takmičari su ostvarili 0 bodova

\* Učenici su ostvarili jednak broj bodova, ali na osnovu propozicija prednost se daje učeniku koji je ranije završio takmičenje.

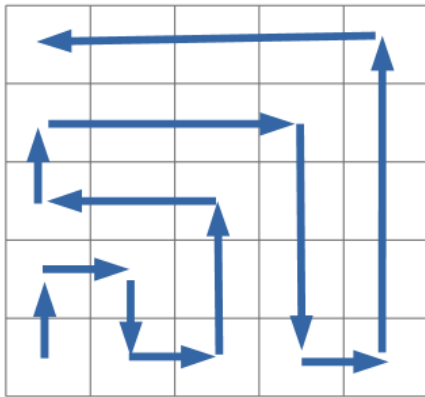
## Rezultati po školama

Mjesto	Učenik	Škola	BODOVI	ŠKOLA UKUPNO
1.	Muamer Parić	Druga gimnazija	350	620
	Ali Dlakić		270	
2.	Irfan Smajević	Međunarodna srednja škola	300	560
	Faruk Alibašić		260	
3.	Paško Zdilar	Srednja elektrotehnička škola	250	300
	Nikola Raič		50	
4.	Fatih Zukorlić	Tursko-bosanski sarajevo koledž	300	300
5.	Adi Šoše	Prva gimnazija	130	130
6.	Armin Bilal	Četvrta gimnazija	100	100
7.	Damir Živojević	Međunarodna srednja škola	70	70
8.	David-Dragan Hadžić	Katolički školski centar	60	60
9.	Adrian Milišić	Gimnazija Obala	20	20
10.	Armin Teskeredžić	Gimnazija Dobrinja	10	10

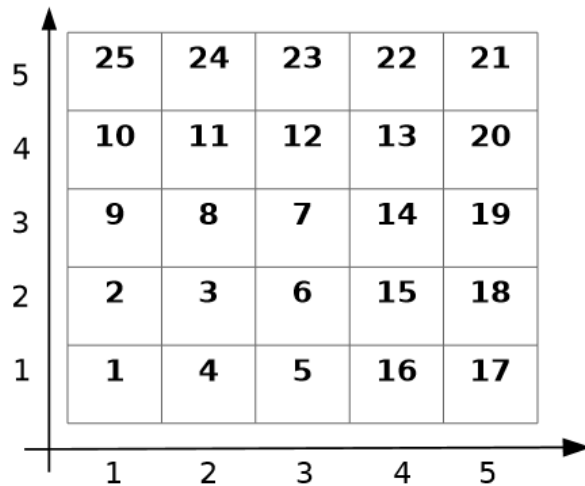
**ZADACI**

**MRAV**

Mrav Mirko je dobio zadatak da na jednoj ogromnoj livadi pronade hranu. Mravinjak se nalazi u donjem lijevom uglu livade. Mrav je mali i ne može vidjeti cijelu livadu odjednom. Kako bi pronašao hranu najbližu mravinjaku, Mirko je odlučio da livadu podijeli na polja veličine 10x10 centimetara i da zatim obilazi ta polja po redoslijedu kao na slici 1:



Slika 1.



Slika 2.

Kraljica bi željela da zna u svakom trenutku gdje se Mirko nalazi. Vaš zadatak je da napravite program koji rješava ovaj problem, odnosno koji daje lokaciju (koordinate) na kojima se nalazi mrav nakon N koraka. Postavićemo koordinatni sistem sa ishodištem u donjem lijevom uglu tako da prvo polje koje obilazi mrav ima koordinate (1,1), drugo polje ima koordinate (1,2), zatim (2,2), pa (2,1), pa (3,1) itd. Na slici 2 smo dali ovaj koordinatni sistem i u svako polje smo upisali broj koji označava korak nakon kojeg će se Mirko nalaziti u tom polju.

Vaš program za dato N treba ispisati koordinate na kojima se Mirko nalazio nakon N koraka.

**Format ulaza**

Ulaz će se sastojati od nekoliko (ne više od 10) linija i u svakoj liniji se nalazi broj N ( $N \leq 10^9$ ). Nula (0) označava kraj unosa.

**Format izlaza**

Za svakih od tih brojeva N ispisati dva broja – Mirkove koordinate.

## Primjeri

### Primjer br. 1

<i>Ulaz:</i>	<i>Izlaz:</i>
8	2 3
20	5 4
25	1 5
0	

### Ograničenja na resurse i opis subtaskova

**1 AT (10 bodova):**  $N \leq 50$

**3 AT (30 bodova):**  $N \leq 1000$

**Podzadatak 3 (60 bodova):**  $N \leq 1000000000$

**ZAGRADE**

Dat je string dužine najviše 1000 znakova. U stringu se mogu nalaziti velika i mala slova, cifre i svi uobičajeni znakovi interpunkcije, te se između ostalog mogu nalaziti i zagrade: obične ili male zagrade (), uglaste ili srednje zagrade [], i vitičaste ili velike zagrade {}.

Program treba vratiti broj 1 ako je svaka otvorena zagrada bilo kojeg tipa ispravno zatvorena zagradom istog tipa, a 0 ako nije. Ne treba provjeravati matematičko pravilo ga se male zagrade nalaze unutar srednjih a srednje unutar velikih.

Na primjer, ako string glasi

```
[()]{[()]}()
```

Program treba ispisati 1. Vidimo da je otvorena srednja zagrada na poziciji 1 zatvorena na poziciji 4, mala zagrada na poziciji 2 je zatvorena na poziciji 3 itd. Ali ako string glasi

```
[()]
```

treba ispisati 0 jer se male i srednje zagrade ukrštaju. Ako se u stringu ne nalazi niti jedna zagrada treba ispisati 1. Znakove koji nisu zagrade zanemariti.

**Format ulaza**

U prvom redu se nalazi jedan prirodan broj N ne veći od 100, a u narednih N redova se nalaze stringovi ne duži od 1000 znakova.

Format izlaza

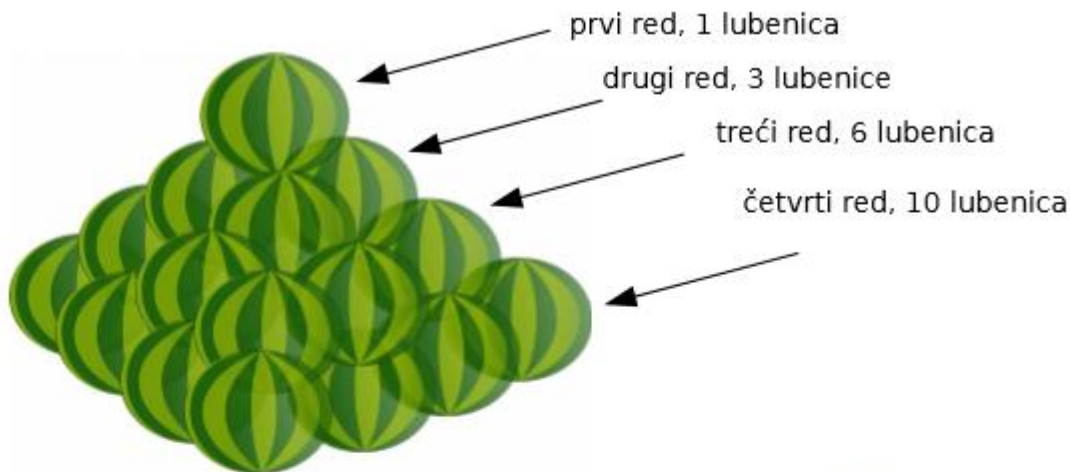
Za svaki od N stringova treba ispisati nulu ili jedinicu, u zasebnom redu.

**Primjer**

Ulaz:	Izlaz:
2	1
[()]{[()]}()	0
[()]	

## LUBENICE

Neki seljak je poredao lubenice pored raskrsnice kako bi ih prodao. Lubenice su obrazovale trostranu piramidu: na vrhu se nalazila jedna lubenica, u drugom redu 3 lubenice, u trećem redu 6 lubenica itd.



Pogled odozgo na četvrti red sa 10 lubenica.

Vidimo da lubenice obrazuju jednakostranični trougao.



Kada mu je stigao sin, prevezao je određeni broj lubenica do sljedeće raskrsnice, tako da su prodavali na dva mjesta. Poslije podjele, sinova gomila je također obrazovala pravilnu trostranu piramidu, a i one lubenice koje su ostale kod oca su ponovo obrazovale pravilnu trostranu piramidu! Koliko je bilo lubenica?

Vaš zadatak je da napišete program u koji se unose dva broja  $A$  i  $B$ , a na izlazu se ispisuju svi brojevi  $N$  između  $A$  i  $B$  za koje je moguć opisani scenario.

### Format ulaza

Na ulazu će se nalaziti dva prirodna broja  $A$  i  $B$  razdvojena znakom razmaka. Potrebno je naći sve brojeve  $N \in [A, B]$  za koje je moguć opisani scenario sa lubenicama. Brojevi su manji od  $2^{32}$ .



**Format izlaza**

Niz brojeva koji zadovoljavaju opisani scenario, svaki u zasebnom redu izlaza.

**Primjeri****Primjer br. 1**

<i>Ulaz:</i>	<i>Izlaz:</i>
1 1000	20 680

**Ograničenja na resurse i opis subtaskova**

**1 AT (10 bodova):**  $A, B \leq 10^3$

**3 AT (30 bodova):**  $A, B \leq 10^6$

**Podzadatak 3 (60 bodova):**  $A, B \leq 2^{32}$

## ENKRIPCIJA

Nedavno je izmišljena nova metoda enkripcije koja svoju sigurnost, kao i mnoge druge,uguje prostim brojevima. On što je sada novo je što brojeve u bazi 10 možemo pretvoriti u neku drugu brojnu bazu manju ili jednaku 10 (koja ima smisla, tj. veću ili jednaku bazi 2) i pročitati ih kao da su zapisani standardno u bazi 10. Za tako neku bazu  $2 \leq B \leq 10$  i prirodan broj  $X$ , ćemo broj dobijen prethodno opisanim postupkom za  $X$  i bazu  $B$ , označavati kao  $X\{B\}$ .

Npr. broj  $1010_{(10)} \rightarrow 1010_{(2)}$ , a to možemo pročitati kao da je hiljadu i deset tj.  $1010_{10}$ , pa je  $10\{2\}=1010$ ,  $10\{3\}=101$ ,  $10\{4\}=22$ , ...,  $10\{9\}=11$  i  $10\{10\}=10$ . Onaj ko želi da razbije novu enkripciju treba pokušati da pronađe što više super-složenih brojeva. Super-složeni broj je broj  $X$  za koji vrijedi da su svi brojevi  $X\{B\}$  za  $2 \leq B \leq 10$  složeni (naravno kad su intepretirani u bazi 10). Pored toga, kako bi dokazali svoj tvrdnje da je broj super-složen, morate naći barem jedan djelilac za svaku od tih baza (koji nije 1 ili sam taj broj), tj. ukupno 9 djelilaca. U gornjem primjeru odmah primijetimo da 10 nije super-složen, jer iako je za  $B=2, 4, 5, 6, 8, 10$  složen, to vidimo da za  $B=3, 7$  i  $9$  dobijamo redom 101, 13 i 11 što kada ih interpretiramo u bazi 10 su prosti brojevi. Vaš zadatak je da generišete  $K$  različitih (bez ponavljanja) supersloženih brojeva koji su striktno veći od nekog broja  $N$ . Poredak brojeva je nebitan, a i svako tačno rješenje će biti prihvaćeno. Format ulaza U prvom redu se nalaze dva prirodna broja  $N(10 \leq N \leq 1000000)$  i  $1 \leq K \leq 500$ , čije značenje je opisano u tekstu.

Format izlaza Izlaz treba da sadrži tačno  $N$  linija sa po 10 brojeva, razdvojenih razmacima, i kao prvi broj u liniji stoji super-složeni broj  $X$  koji ste našli, a sljedećih 9 brojeva predstavlja po jedan djelilac od  $X\{B\}$  za redom  $B=2, 3, \dots, 10$ .

### Primjer

Ulaz:	Izlaz:
11 5	51 11 2 3 3 3 2 3 2 3 69 3 2 3 2 3 2 3 2 3 99 11 2 3 2 3 3 11 2 3 125 3 2 11 2 5 2 5 2 5 129 11 2 3 2 3 3 3 3 3

**Objašnjenje:** Za broj 99 imamo da je super-složen jer je  $99\{B\}$  za B od 2 do 10: 1100011 10200 1203 344 243 201 143 120 99, a po jedan njihov djelilac redom je: 11 2 3 2 3 3 11 2 3.

Pazite da brojevi u manjim bazama postaju dosta veći (brojem cifara), te je garantovano da rješenja postoje tako da najveći broj generisan u tačnom algoritmu stane u 64-bitni cjelobrojni tip.

## **PRIJEDLOG MOGUĆIH RJEŠENJA**

## MRAV – C++

```

1  #include <iostream>
2  #include <cmath>
3  #include <algorithm>
4  #include <stdlib.h>
5
6  using namespace std;
7
8  int main()
9  {
10     int n;
11     int a;
12     int b;
13     cin>>n;
14     while (n>0)
15     {
16         a=1;
17         while (a*a<n)
18         {
19             a++;
20         }
21         b=a*a-n;
22         if (b==0)
23         {
24             if (a%2==0) {cout<<a<<" "<<"1"<<endl;} else {cout<<"1"<<" "<<a<<endl;}
25         }
26         else if (b<a)
27         {
28             if (a%2==0) {cout<<a<<" "<<1+b<<endl;} else {cout<<1+b<<" "<<a<<endl;}
29         }
30         else if (b>=a)
31         {
32             if (a%2==0) {cout<<a+a-b-1<<" "<<a<<endl;} else {cout<<a<<" "<<a+a-b-1<<endl;}
33         }
34
35
36         cin>>n;
37     }
38     return 0;
39 }

```

## ZAGRADE – C++

```

1  #include <string>
2  #include <iostream>
3  #include <stack>
4
5  using namespace std;
6
7  template<typename Tip>
8  Tip poptop(stack<Tip> s) {
9      Tip t = s.top();
10     s.pop();
11     return t;
12 }
13
14 bool provjeri_zagrade(std::string tekst)
15 {
16     stack<char> stek;
17     if(tekst.size() == 0) return false;
18     for(int i = 0; i < int(tekst.size()); i++)
19     {
20         if(tekst[i] == '(' || tekst[i] == '[' || tekst[i] == '{')
21             stek.push(tekst[i]);
22         if (tekst[i] == ')' || tekst[i] == ']' || tekst[i] == '}') {
23             if (stek.size() == 0) return false;
24
25             if (tekst[i] == ')' && stek.top() != '(') return false;
26             if (tekst[i] == ']' && stek.top() != '[') return false;
27             if (tekst[i] == '}' && stek.top() != '{') return false;
28             stek.pop();
29         }
30     }
31     if(stek.size() != 0) return false;
32     return true;
33 }
34
35 int main() {
36     int n; string s;
37     cin >> n;
38     getline(cin, s);
39     for (int i(0); i<n; i++) {
40         getline(cin, s);
41         if (provjeri_zagrade(s))
42             cout << "1\n";
43         else
44             cout << "0\n";
45     }
46     return 0;
47 }

```

LUBENICE - Pascal

```
var
    p:array[1..3000] of Int64;
    m, n, tn, pn, j, i: Int64;
    a, b: Int64;
begin
    m := 0; n := 0; tn := 0; pn := 0; a := 0;
    ReadLn(a,b);
    repeat
        j := n; i := 1;
        n := n+1; tn := tn+n;
        if pn > high(LongInt)-tn then break;
        pn := pn+tn; p[n] := pn;
        while (i <= j) and (p[i]+p[j] <> pn) do
            begin
                while (i<n) and (p[i] < pn-p[j]) do i := i+1;
                while (j>0) and (p[i] > pn-p[j]) do j := j-1;
            end;
            if i<=j then
                begin
                    m := m+1;

                    if (pn >= a) and (pn <= b) then writeln(pn);
                end;
            until pn>b;
        end.
```

## ENKRIPCIIJA – C++

```

1
2 #include <cstdio>
3 #include <vector>
4 using namespace std;
5 vector<unsigned int> prost(10000);
6 vector<bool> isPrime(1<<17);
7 unsigned int koliko=0;
8 void sito() {
9     for(int i=0; i<=1<<16; i++) isPrime[i]=i%2;
10    isPrime[2]=true;
11    prost[koliko]=2;
12    for(unsigned int x=3; x<1<<16; x+=2) {
13        if(isPrime[x]) {
14            koliko++;
15            prost[koliko]=x;
16            for(unsigned int l=x*x; l<1<<16; l=l+2*x) {
17                isPrime[l]=false;
18            }
19        }
20    }
21 }
22 unsigned long long dajBazu(int X, int B) {
23     unsigned long long rez=0;
24     unsigned long long t=1;
25     while(X!=0) {
26         rez+=(X%B)*t;
27         X=X/B;
28         t=t*10;
29     }
30     return rez;
31 }
32 unsigned long long jelProst(unsigned long long u) {
33     for(unsigned int i=0; i<koliko; i++) {
34         if(prost[i]*prost[i]>u) return 0;
35         if(u%prost[i]==0) return prost[i];
36     }
37     return 0;
38 }
39 int main()
40 {
41
42     unsigned int n;
43     unsigned int k;
44     sito();
45     scanf("%d %d", &n, &k);
46     unsigned long long djelioc[9];
47     while(k>0) {
48         n++;
49         bool nProst=false;
50         djelioc[8]=jelProst(n);
51         if(djelioc[8]!=0) nProst=true;
52         for(unsigned short b=2; b<=9&&nProst; b++) {
53             djelioc[b-2]=jelProst(dajBazu(n,b));
54             nProst=(djelioc[b-2]!=0);
55         }
56         if(nProst) {
57             printf("%d ", n);
58             for(unsigned short j=0; j<9; j++) printf("%d ", djelioc[j]);
59             printf("\n");
60             k--;
61         }
62     }
63
64     return 0;
65 }
66

```