

**KANTONALNO TAKMIČENJE
IZ INFORMATIKE ZA UČENIKE OSNOVNIH ŠKOLA 2015**

Sarajevo, april 2015. godine

KANTONALNO TAKMIČENJE IZ INFORMATIKE 2015
za učenike osnovnih škola sa područja Kantona Sarajevo

SPISAK PRIJAVLJENIH ŠKOLA

Redni broj	Škola	R/B	Takmičari	Razred	Programski jezik	Nastavnik mentor
1.	OŠ „Mehmed-beg Kapetanović Ljubušak“	1.	1. Zatega Admir	V	QBASIC	Edin Smajić
		2.	2. Šabić Ammar	V	QBASIC	
		3.	3. Šehić Amina	VI	QBASIC	
2.	OŠ „Saburina“	4.	1. Fočak Aiša	VI	QBASIC	Suvada Vlajčić
		5.	2. Hajrić Amina	VI	QBASIC	
		6.	3. Ibrahimović Emira	V	QBASIC	
3.	OŠ „Osman Nakaš“	7.	1. Velašević Boris	IX	QBASIC	Mahir Zolota
		8.	2. Čengić Kemal	VIII	QBASIC	
4.	OŠ „Hrasno“	9.	1. Golubović Ervin	IX	QBASIC	Haris Zametica
		10.	2. Hadžić Armin	IX	QBASIC	
		11.	3. Mujezinović Matej	IX	QBASIC	
5.	OŠ „Četvrta osnovna“	12.	1. Skelić Lejla	IX	QBASIC	Samira Gojaković
		13.	2. Hošić Nejra	IX	QBASIC	
		14.	3. Šehović Belma	IX	QBASIC	
6.	OŠ "Hilmi ef.Šarić" Tarčin	15.	1. Smajević Arslan	VII	C++	Mehmedalija Salihović
		16.	2. Čičko Ermin	IX	C++	
		17.	3. Šerbo Edin	VII	QBASIC	
7.	OŠ „Srednje“	18.	1. Dervišević Ishak	VIII	C++	Šerif Vrapčević
		19.	2. Mališević Haris	VI	Basic	
		20.	3. Malagić Eldin	VIII	Basic	
8.	OŠ „Zahid Baručija“	21.	1. Đaka Tarik	IX	QBASIC	Edisa Bajramović
		22.	2. Botić Emir	IX	QBASIC	
		23.	3. Smajlović Enis	IX	QBASIC	

PROGRAM TAKMIČENJA

08:15-08:30	Okupljanje ekipa
08:30-09:00	Otvaranje takmičenja, prezentacija domaćina i izvlačenje takmičarskih šifri
09:00-12:00	Pojedinačno takmičenje
12:00-12:45	Pregled zadataka i priprema preliminarne rezultata
12:15-12:45	Ručak
12:45-13:30	Uvid u zadatke i eventualne reklamacije
13:45-14:15	Proglašenje pobjednika

Konačni rezultati Kantonalnog takmičenja iz Informatike za učenike osnovnih**Plasman učenika pojedinačno i plasman škola**

Mjesto	Prezime i ime	Škola	Z1	Z2	Z3	Z4	UKUPNO
1.	Velašević Boris	OŠ “Osman Nakaš”	0	10	100	0	110
2.	Zatega Admir	OŠ “Mehmed-beg Kapetanović Ljubušak”	40	0	30	0	70
3.	Dervišević Ishak	OŠ “Srednje”	0	0	30	0	30
4.	Đaka Tarik	OŠ “Zahid Baručija”	0	0	10	0	10
4.	Ibrahimović Emira	OŠ “Saburina”	0	0	10	0	10
5-19.	Ostali takmičari		0	0	0	0	0

Z A D A C I

Chuck Norris (chuck_norris)

Binarni brojni sistem sa 1 i 0 je dobar, ali “binarni” sistem samo sa nulama je još bolji! Originalno, ovaj koncept je dizajnirao Chuck Norris kako bi slao takozvane “unarne” poruke.



Princip kodiranja je sljedeći:

Ulazna poruka se sastoji od ASCII karaktera gdje je svaki karakter predstavljen sa 7 bita.

Izlazna kodirana poruka se sastoji od blokova nula.

Blokovi se razdvajaju pomoću praznog mjesta.

Dva susjedna bloka se koriste za predstavljanje serije bita koji imaju istu vrijednost (serija jedinica ili serija nula):

Prvi blok: je uvijek 0 ili 00. Ako je 0, onda serija sadrži samo jedinice, a ako je 00, onda serija sadrži samo nule.

Drugi blok: sadrži onoliko nula koliko se u seriji koju kodiramo nalazi jedinica ili nula.

Na primjer, ako Chuck Norris želi poslati prvo slovo svog imena “C”, čija je binarna reprezentacija 1000011, onda će se upotrebom ove tehnike dobiti:

- 0 0 (prva serija sadrži jednu jedinicu),
- 00 0000 (druga serija sadrži četiri nule),
- 0 00 (treća serija sadrži dvije jedinice).

To znači da će Chuck Norris poslati 0 0 00 0000 0 00.

Chuck Norris je od pretjerane upotrebe revolvera povrijedio palac i kažiprst na svojoj desnoj ruci, pa ne može više koristiti “unarnu” tastaturu koja ima dvije tipke: 0 i space. Vaš zadatak je da pomognete Chuck Norrisu i napišete program koji će sa standardnog ulaza učitati poruku koji želi poslati, a na standardni izlaz ispisati kodiranu “unarnu” poruku.

Ulaz:

Na standardnom ulazu nalazi se samo jedan podatak: poruka koju želi poslati Chuck Norris. Poruka je niz validnih ASCII karaktera (osim znaka "razmak") koji neće biti duži od 10000 karaktera.

Za takmičare koji programiraju u QBasicu/QB64 ulaz se čita iz datoteke “**input.txt**” koja se nalazi u istom folderu kao i .BAS program (bez zadavanja puta, dakle C:\input.txt nije ispravno).

Izlaz:

Kodirana “unarna” poruka.

Primjer 1:

Ulaz:

C

Izlaz:

0 0 00 0000 0 00

Primjer 2:

Ulaz:

CC

Izlaz:

0 0 00 0000 0 000 00 0000 0 00

Sortiranje prostih i složenih brojeva

Na ulazu se nalazi niz pozitivnih cijelih brojeva. Potrebno je:
razdvojiti proste od složenih brojeva, te ispisati najprije proste a zatim složene;
proste sortirati u opadajućem redoslijedu, a složene u rastućem.

Ulaz:

Na standardnom ulazu nalazi se najprije pozitivan cijeli broj N (ne veći od 1000). Zatim slijedi N pozitivnih cijelih brojeva.

Za takmičare koji programiraju u QBasicu/QB64 ulaz se čita iz datoteke “**input.txt**” koja se nalazi u istom folderu kao i .BAS program (bez zadavanja puta, dakle C:\input.txt nije ispravno).

Izlaz:

Najprije je potrebno ispisati sve proste brojeve sa ulaza, i to u opadajućem redoslijedu, a zatim složene brojeve u rastućem redoslijedu.

Primjer 1:

Ulaz:

5
3
18
11
7
22

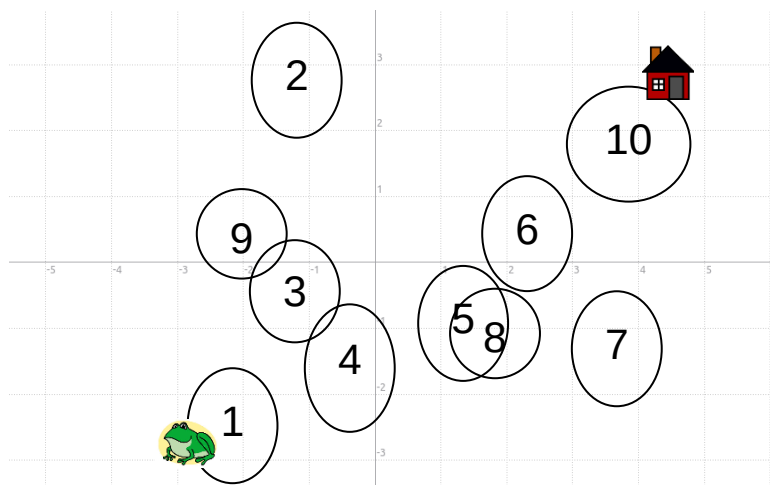
Izlaz:

11
7
3
18
22

Žabac

Žabac Frogi se čitav dan igrao na jezeru i hvatao mušice. Kada se počelo mračiti, shvatio je da je odlutao daleko od kuće. Pomozi Frogiju da se vrati kući prije večere!

Da bi Frogi prešao jezero, mora skakati sa jednog na drugi list lokvanja. Radi jednostavnijeg rješenja, listovi lokvanja su predstavljeni kao savršene kružnice na površini jezera. Frogi je još mali i može preskočiti sa jednog na drugi list samo ako su udaljeni manje od 10 cm (0.1 metar). Listovi se mogu dodirivati, preklapati, čak je moguće da jedan list potpuno prekriva drugi. Listovi su označeni rednim brojevima 1, 2, 3...



Vaš zadatak je da napravite program koji ispisuje kojim listovima Frogi treba ići da bi stigao do svoje kuće, ili ako nije moguće stići nježno saopštiti Frogiju da danas nema večere.

Ulaz:

Na standardnom ulazu nalaze se sljedeći podaci:

Linija 1: dva realna broja (pozitivna ili negativna) koji označavaju koordinate na kojima se trenutno nalazi žabac Frogi.

Linija 2: dva realna broja (pozitivna ili negativna) koji označavaju koordinate na kojima se nalazi Frogijeva kuća.

Linija 3: pozitivan cijeli broj N koji predstavlja broj lokvanja ($0 < N < 100$).

U narednih N linija nalaze se podaci o lokvanjima: tri realna broja koji označavaju koordinate centra kružnice i poluprečnik kružnice.

Za takmičare koji programiraju u QBasicu/QB64 ulaz se čita iz datoteke "input.txt" koja se nalazi u istom folderu kao i .BAS program (bez zadavanja puta, dakle C:\input.txt nije ispravno).

Izlaz:

Ukoliko se žabac Frogi trenutno ne nalazi na lokvanju ili ukoliko nije moguće doći do kuće, program treba na standardnom izlazu ispisati poruku "Frogi ce ostati gladan" (bez navodnika).

U suprotnom treba ispisati lokvanje preko kojih će Frogi skakati. Svaki lokvanj je označen svojim rednim brojem, pa za prvi lokvanj koji je naveden na ulazu treba ispisati 1, za drugi 2 itd. Treba ispisati ove brojeve na standardnom izlazu u jednom redu razdvojene razmakom. U slučaju da postoji više puteva treba ispisati najkraći (sa najmanje skokova), a slučaj kada postoji više puteva sa istim brojem koraka neće biti testiran (svejedno je šta vaš program ispisuje).

Primjer 1:

Ulaz:

```
0 0
2 2
2
1 1 0.5
1.5 1.5 0.5
```

Izlaz:

Frogi ce ostati gladan

Primjer 2:

Ulaz:

```
0 0
2 2
2
1 1 0.5
0.5 0.5 0.8
```

Izlaz:

Frogi ce ostati gladan

Primjer 3 (prikazan na slici):

Ulaz:

```
-3 -3
4 2.5
10
-2 -2.5 1.2
-1 2.5 1
-1 -0.5 0.7
0.2 -1.8 1.2
1.3 -1 1
2.3 0.4 1
3.8 -1.3 0.8
1.8 -1 0.5
-2 0.5 0.7
3.8 1.8 1
```

Izlaz:

1 4 5 6 10

PRIJEDLOG RJEŠENJA (C++)

ASCII

```
#include <iostream>
#include <string>
#include <vector>
#include <algorithm>

using namespace std;

int main()
{
    int L;
    cin >> L; cin.ignore();
    int H;
    cin >> H; cin.ignore();
    string T;
    getline(cin, T);
    for (int i = 0; i < H; i++) {
        string ROW;
        getline(cin, ROW);
        for (int j = 0; j < T.size(); j++) {
            for (int k = 0; k < L; k++) {
                char c = T[j];
                if (c >= 'a' && c <= 'z')
                    c = c + 'A' - 'a';
                else if (c < 'A' || c > 'Z')
                    c = 'A' + 26;
                cout << ROW[(int)(c - 65) * L + k];
            }
            cout << endl;
        }
        cout << endl;
    }
}
```

CHUCK NORRIS

```
#include <iostream>
#include <string>
#include <vector>
#include <algorithm>

using namespace std;

int main()
{
    string MESSAGE;
    getline(cin, MESSAGE);

    char s = 0;
    int ones = 0;
    int zeros = 0;
    for (int i = 0; i < MESSAGE.size(); i++) {
        char c = MESSAGE[i];
        for (int j = 0; j < 7; j++) {
            switch(s) {
                case 0:
                    if ((c & 64) == 64) {
                        ones++;
                        s = 1;
                    } else {
                        zeros++;
                        s = 2;
                    }
                    break;
                case 1:
                    if ((c & 64) == 64) {
                        ones++;
                    } else {
                        zeros++;
                        cout << "0 ";
                        while(ones-- > 0)
                            cout << "0";
                        ones = 0;
                        cout << " ";
                        s = 2;
                    }
                    break;
                case 2:
                    if ((c & 64) == 64) {
                        ones++;
                        cout << "00 ";
                        while(zeros-- > 0)
                            cout << "0";
                        zeros = 0;
                        cout << " ";
                        s = 1;
                    } else {
                        zeros++;
                    }
                    break;
            }
            c = c << 1;
        }
    }
}
```

```
switch(s) {  
    case 1:  
        cout << "0 ";  
        while(ones-- > 0)  
            cout << "0";  
        break;  
    case 2:  
        cout << "00 ";  
        while(zeros-- > 0)  
            cout << "0";  
        break;  
}  
cout << endl;  
}
```

SORTIRANJE PROSTIH I SLOŽENIH BROJEVA

```
#include <iostream>
#include <algorithm>
#include <fstream>
#include <cstdlib>

using namespace std;

void generate() {
    int brulaza(50);
    ofstream test("ulaz.txt");
    test << brulaza << endl;
    for (int i(0); i<brulaza; i++)
        test << rand()%1538 << endl;
    test.close();
}

int main() {
    //generate();
    //return 0;

    int prosti[1000], slozeni[1000];
    int n;
    cin >> n;
    int brp(0), brs(0);
    for (int i(0); i<n; i++) {
        int broj;
        cin >> broj;
        bool prost(true);
        for (int k(2); k<broj; k++)
            if (broj%k==0) prost=false;
        if (prost)
            prosti[brp++] = broj;
        else
            slozeni[brs++] = broj;
    }

    sort(prosti, prosti+brp);
    sort(slozeni, slozeni+brs);
    for (int i(brp-1); i>=0; i--)
        cout << prosti[i]<<endl;
    for (int i(0); i<brs; i++)
        cout << slozeni[i]<<endl;
    return 0;
}
```

ŽABAC

```

#include <iostream>
#include <cmath>
#include <vector>

using namespace std;

struct Tacka { double x, y; };
struct Kruznica { Tacka c; double r; };

Kruznica lokvanji[100];
int br_lokvanja;
int debug=0;

double daljina(Tacka a, Tacka b) {
    return sqrt((a.x-b.x)*(a.x-b.x) + (a.y-b.y)*(a.y-b.y));
}

vector<int> dfs(vector<int> put, int poc, int kraj) {
    put.push_back(poc);
    if (poc==kraj) return put;
    vector<int> najbolji;
    for (int i(0); i<br_lokvanja; i++) {
        if (i==poc) continue;
        double maxd = lokvanji[poc].r + lokvanji[i].r + 0.1;
        double d = daljina(lokvanji[poc].c, lokvanji[i].c);
        if (debug>0) cout << "d("<<poc<<", "<<i<<")="<<d<<
maxd="<<maxd<<endl;
        if (d > maxd) continue;

        // Da li smo stigli?
        if (i == kraj) {
            put.push_back(kraj);
            return put;
        }

        // Da li je veÄ# bio tu
        bool bio(false);
        for (int j(0); j<put.size()-1; j++)
            if (put[j] == i) bio=true;
        if (bio) continue;

        // Rekurzija
        vector<int> ovuda = dfs(put, i, kraj);
        if (najbolji.size() == 0 || ovuda.size() > 0 && ovuda.size() <
najbolji.size())
            najbolji = ovuda;
    }
    return najbolji;
}

int main(int argc, char** argv) {
    Tacka frogi, kuca;
    debug = argc-1;

    cin >> frogi.x >> frogi.y;
    cin >> kuca.x >> kuca.y;
    cin >> br_lokvanja;
    for (int i(0); i<br_lokvanja; i++)

```



```
cin >> lokvanji[i].c.x >> lokvanji[i].c.y >> lokvanji[i].r;

// Da li je frogi na lokvanju?
int na_lokvanju = -1;
for (int i(0); i<br_lokvanja; i++) {
    if (debug==1) cout << "zabac_lokvanj i="<<i<<" d="<<daljina(frog,
lokvanji[i].c)<<" maxd="<<(lokvanji[i].r+0.1)<<endl;
    if (daljina(frog, lokvanji[i].c) < lokvanji[i].r) {
        na_lokvanju = i;
        break;
    }
}
if (na_lokvanju == -1) {
    cout << "Frog ce ostati gladan" << endl;
    return 0;
}

// Da li je kuca manje od 0.1 od lokvanja
int kuca_lokvanj = -1;
for (int i(0); i<br_lokvanja; i++) {
    if (debug==1) cout << "kuca_lokvanj i="<<i<<" d="<<daljina(kuca,
lokvanji[i].c)<<" maxd="<<(lokvanji[i].r+0.1)<<endl;
    if (daljina(kuca, lokvanji[i].c) < lokvanji[i].r+0.1) {
        kuca_lokvanj = i;
        break;
    }
}
if (kuca_lokvanj == -1) {
    cout << "Frog ce ostati gladan" << endl;
    return 0;
}

vector<int> put;
put = dfs(put, na_lokvanju, kuca_lokvanj);
if (put.size() == 0)
    cout << "Frog ce ostati gladan";
else
    for (int i(0); i<put.size(); i++)
        cout << (put[i]+1) << " ";
cout << endl;

return 0;
}
```

Napomena o načinu bodovanja zadataka i programskom jeziku BASIC

Nastavni planovi i programi informatike predviđaju upotrebu programskog jezika BASIC, a u većini škola se koristi okruženje QBasic za DOS ili noviji QB64. No u novije vrijeme na državnim takmičenjima, a od 2014. godine i na Kantonalnom takmičenju, zadaci se pregledaju automatski. Ovo nije moguće realizirati koristeći QBasic ili QB64, pa se za te potrebe koristi FreeBasic u "QBasic modu".

Iako autori softvera FreeBasic garantuju da je uz korištenje "QBasic moda" ovaj program 99% kompatibilan sa QBasic-om, postoje neke sitne razlike na koje želimo da skrenemo pažnju:

- **U FreeBasic-u nije moguće deklarirati niz i varijablu koji se isto zovu**

U QBasic-u ako imate sljedeći kod:

```
DIM a$(50)
INPUT a$
```

on će ispravno raditi budući da su niz a\$ i string a\$ različite varijable, a stringovi se ne moraju deklarirati. Nažalost, neki učenici očito misle da se linijom DIM a\$(50) deklarira string dužine 50 znakova. Linija DIM a\$(50) ustvari deklarira niz od 50 stringova! FreeBasic neće dozvoliti postojanje niza i varijable sa istim imenom, pa će kod dati iznad na serveru proizvesti Compiler error.

- **S druge strane, nizovi se moraju dimenzionirati.**

Primjećujemo da neki dijalekti QBasica dozvoljavaju korištenje nizova bez dimenzioniranja u nekim slučajevima. U FreeBasicu se svi korišteni nizovi moraju propisno deklarirati naredbom DIM.

- **Ograničenja broja karaktera na ulazu**

Kao što je poznato u QBasic-u se ne može u string unijeti više od oko 200 karaktera. FreeBasic dozvoljava unos do 4096 karaktera. Da biste izbjegli ove probleme, ako se u zadatku traži unos velikog broja karaktera, prepravite vaš program tako da umjesto sa standardnog ulaza čita podatke iz datoteke **input.txt**.

- Posebno molimo takmičare da ne koriste naredbu LINE INPUT za čitanje sa standardnog ulaza zbog poznatog buga u FreeBasic okruženju. Molimo vas da koristite INPUT za standardni ulaz ili datoteku **input.txt**. Prilikom sastavljanja zadataka će se paziti da veličina ulaza ne prekoračuje 4096 karaktera te da stringovi ne sadrže razmake.