



Općinsko takmičenje iz informatike

Osnovne škole

Kanton Sarajevo, 2025

RJEŠENJA

Zadatak 1 - 42

C++

```
#include<iostream>
using namespace std;
int main() {
    int N; cin >> N;
    if(N > 42) cout << "Veci" << endl;
    else if(N < 42) cout << "Manji" << endl;
    else cout << "Jednak" << endl;
    return 0;
}
```

Python

```
n = int(input())
if n > 42:
    print("Veci")
elif n < 42:
    print("Manji")
else:
    print("Jednak")
```

Zadatak 2 - ABC

C++

```
#include<iostream>
using namespace std;
int main() {
    int a, b, c; cin >> a >> b >> c;
    string poredak; cin >> poredak;
    // Poredamo po velicini
    int temp;
    if(b > c) { temp = b; b = c; c = temp; }
    if(a > b) { temp = a; a = b; b = temp; }
    // Sada je zagantovano da je a najmanji, potrebno još provjeriti da li je b veće od c
    if(b > c) { temp = b; b = c; c = temp; }
    for(char ch : poredak) {
        if(ch == 'A') cout << a << " ";
        else if(ch == 'B') cout << b << " ";
        else if(ch == 'C') cout << c << " ";
    }
    return 0;
}
```

Python

```
a_str, b_str, c_str = input().split()
a = int(a_str)
b = int(b_str)
c = int(c_str)

poredak = input()

if b > c:
    b, c = c, b

if a > b:
    a, b = b, a
# U ovom trenutku je zagantovano da je a najmanji element, potrebno provjeriti da li je b
veće od c
if b > c:
    b, c = c, b

for ch in poredak:
    if ch == 'A': print(a, end=" ")
    if ch == 'B': print(b, end=" ")
    if ch == 'C': print(c, end=" ")
```

Zadatak 3 - XOX

C++

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

int value(char c) {
    if(c == 'x') return 1;
    return 2;
}

int main() {
    string matrica[3];
    for(int i = 0; i < 3; i++) cin >> matrica[i];

    // Provjera dijagonala
    if(matrica[0][0] == matrica[1][1] && matrica[0][0] == matrica[2][2])
        cout << value(matrica[1][1]);
    else if(matrica[0][2] == matrica[1][1] && matrica[1][1] == matrica[2][0])
        cout << value(matrica[1][1]);
    // Provjera kolona
    else if(matrica[0][0] == matrica[1][0] && matrica[0][0] == matrica[2][0])
        cout << value(matrica[0][0]);
    else if(matrica[0][1] == matrica[1][1] && matrica[0][1] == matrica[2][1])
        cout << value(matrica[0][1]);
    else if(matrica[0][2] == matrica[1][2] && matrica[0][2] == matrica[2][2])
        cout << value(matrica[0][2]);
    // Provjera redova
    else if(matrica[0][0] == matrica[0][1] && matrica[0][0] == matrica[0][2])
        cout << value(matrica[0][0]);
    else if(matrica[1][0] == matrica[1][1] && matrica[1][0] == matrica[2][1])
        cout << value(matrica[1][0]);
    else if(matrica[2][0] == matrica[2][1] && matrica[2][0] == matrica[2][2])
        cout << value(matrica[2][0]);
    return 0;
}
```

Python

```
def value(c):
    if c == 'x': return 1
    return 2

matrica = []

for i in range(3):
    matrica.append(input())

# Provjera dijagonala
if matrica[0][0] == matrica[1][1] and matrica[0][0] == matrica[2][2]:
    print(value(matrica[0][0]))
elif matrica[0][2] == matrica[1][1] and matrica[1][1] == matrica[2][0]:
    print(value(matrica[1][1]))
# Provjera kolona
elif matrica[0][0] == matrica[1][0] and matrica[0][0] == matrica[2][0]:
    print(value(matrica[0][0]))
elif matrica[0][1] == matrica[1][1] and matrica[0][1] == matrica[2][1]:
    print(value(matrica[0][1]))
elif matrica[0][2] == matrica[1][2] and matrica[0][2] == matrica[2][2]:
    print(value(matrica[0][2]))
# Provjera redova
elif matrica[0][0] == matrica[0][1] and matrica[0][0] == matrica[0][2]:
    print(value(matrica[0][0]))
elif matrica[1][0] == matrica[1][1] and matrica[1][0] == matrica[1][2]:
    print(value(matrica[1][0]))
elif matrica[2][0] == matrica[2][1] and matrica[2][0] == matrica[2][2]:
    print(value(matrica[2][0]))
```

Zadatak 4 - Skočko

C++

```
#include<iostream>
using namespace std;
int niz_skrivena[6], niz_pokusaj[6];
int main() {
    string skrivena, pokusaj;
    cin >> skrivena;
    cin >> pokusaj;
    int pogodjeni = 0, pogresno_mjesto = 0;
    for(int i = 0; i < 4; i++) {
        // Ukoliko se znakovi poklapaju, tada imamo pogodak
        if(skrivena[i] == pokusaj[i]) pogodjeni++;
        else {
            // Brojimo pojavljivanja određenog znaka u pokušaju i skrivenoj
            kombinaciji ukoliko nismo pogodili
            niz_skrivena[skrivena[i] - 'A']++;
            niz_pokusaj[pokusaj[i] - 'A']++;
        }
    }
    for(int i = 0; i < 6; i++)
        pogresno_mjesto += min(niz_skrivena[i], niz_pokusaj[i]);

    cout << pogodjeni << " " << pogresno_mjesto << endl;
    return 0;
}
```

Python

```
niz_skrivena = [0] * 6
niz_pokusaj = [0] * 6

skrivena = input()
pokusaj = input()

pogodjeni = 0
pogresno_mjesto = 0

for i in range(4):
    if skrivena[i] == pokusaj[i]:
        pogodjeni += 1
    else:
        niz_skrivena[ord(skrivena[i]) - ord('A')] += 1
        niz_pokusaj[ord(pokusaj[i]) - ord('A')] += 1

for i in range(6):
    pogresno_mjesto += min(niz_skrivena[i], niz_pokusaj[i])

print(pogodjeni, pogresno_mjesto)
```

Zadatak 5 - Farma malina

Analizom posljednjeg primjera $N = 959$, zaključujemo da je optimalna raspodjela grmova $N = 512 + 256 + 128 + 32 + 16 + 8 + 4 + 2 + 1$. Kao što vidimo, najmanje grmova postićemo ukoliko broj N raspišemo kao sumu stepena broja 2. Dolazimo do zaključka da je potrebno prebrojati koliko će članova imati gornja suma, a to je upravo broj jedinica u binarnom zapisu.

C++

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

int main() {
    int N; cin >> N;
    int brojac_jedinica = 0;
    while(N) {
        if(N % 2 == 1) brojac_jedinica++;
        N /= 2;
    }
    cout << brojac_jedinica << endl;
    return 0;
}
```

Python

```
N = int(input())
brojac_jedinica = 0

while N != 0:
    if N % 2 == 1:
        brojac_jedinica += 1
    N = N // 2

print(brojac_jedinica)
```


U nastavku prikaz još jednog načina za brojanje broja jedinica u binarnom zapisu koji koristi trik za eliminaciju posljednje jedinice iz binarnog zapisa.

C++

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

int main() {
    int N; cin >> N;
    int brojac_jedinica = 0;
    while(N) { N &= (N - 1); brojac_jedinica++; }
    cout << brojac_jedinica << endl;
    return 0;
}
```

Python

```
N = int(input())
brojac_jedinica = 0

while N != 0:
    N = (N & (N - 1))
    brojac_jedinica += 1

print(brojac_jedinica)
```