
Tema 6

TEME

- Algoritmi
- Rekurzije

PRIMER 1: Broj cifara i zbir cifara broja od 0 do 1000 000 000

23645 -> 5 20

0 -> 1 0

RESENJE 1 - algoritamski:

```
#include <iostream>
#include <math.h>
using namespace std;

int main() {
    int n;
    cin >> n;
    int brojCifara = 0, zbirCifara = 0;

    do {
        brojCifara++;
        int cifra = n % 10;
        zbirCifara += cifra;
        n /= 10;
    } while (n>0);
    cout << brojCifara << " " << zbirCifara << endl;
    return 0;
}
```

RESENJE 2 - rekurzije:

```
#include <iostream>
#include <math.h>
using namespace std;

int main() {
    int n;
    cin >> n;
    cout << brojCifara(n) << " " << zbirCifara(n) << endl;
}
```

```

    return 0;
}

int brojCifara(int n) {
    return (n<10)? 1 : (brojCifara(n/10) + 1);
}

int zbirCifara(int n) {
    return (n<10)? n : (zbirCifara(n/10) + n%10);
}

```

OBJASNJENJE:

- Doci do rekurzije iz prvog resenja, ne direktno napisati, nego postupno korak po korak
- Odmotati rekurziju za neki manji broj i pokazati kako se one pozivaju, tj stavljaju i skidaju sa steka.
- Koncept rekurzija postoji svuda, ne samo u cpp

PRIMER 2: Pronalazak maksimalnog broja

```

// a, b, c integeri   cin
// 15, 1, 82
// -> a je najveći   cout
// leaderboard

#include <iostream>
using namespace std;

int main() {
    int a, b, c, d, maks;
    cin >> a >> b >> c >> d;
    maks = a;    // a=maks=13
    if (b > maks) {
        maks = b;    // b = 23 -> maks = 23
    }
    // maks = a
    if (c > maks) {
        maks = c;
    }
    // maks = a
    if (d > maks) {
        maks = d;
    }
    cout << "Najveci je " << maks << endl;

/*
    if (a > b && a > c) {

```

```

        cout << "Najveci je " << a << endl;
    } else if (b > a && b > c) {
        cout << "Najveci je " << b << endl;
    } else {
        cout << "Najveci je " << c << endl;
    }
    ...
*/
return 0;
}

```

OBJAŠNENJE:

- Prvo uraditi razvrstavanje sa ifovima bez logicki operatora (3 broja a b c)
- Drugu verziju uraditi sa ifovima i logickim operatorima (3 broja a b c)
- Trecu verziju uraditi da se doda cetvrti broj d, kojim se ovi ifovi zakomplikuju
- Onda se pokaze finalno resenje, da se maksimum dodeli da je jedna vrednost i onda samo promeni ako je neki veci.

BONUS PRIMERI - NE MORAJU DA SE RADE AKO NEMA VREMENA, ZADATI ZA VEZBU

PRIMER 3: Od cifara cetvorocifrenog prirodnog broja treba formirati najmanji trocifren broj.

5302 -> 203

- Izdvojimo cifre c, d, s, h iz cetvorocifrenog broja
- Urediti cifre od najmanjeg no najveceg $b \leq b \leq b \leq b$
 - Napravimo swap funkciju
 - Napravimo za uredjenje 4 broja
- Formiramo broj tako da cifra stotina bude razlicita od nule
- Pri formiranu broja, za cifru stotica uzimamo najmanji broj \neq od nule a za desetice i jedinicu najmanje od preostalih cifara

RESENJE:

```

#include <iostream>
using namespace std;
void swap(int& a, int&b){
    int temp = a;
    a = b;
    b = temp;
}
// Funkcija za uredjivanje 4 cela broja a, b, c, d u neopadajucem //
poretku a <= b <= c <= d
void sortiraj(int& a, int& b, int& c, int& d) {

```

```

    if (a > b) swap(a, b);
    if (a > c) swap(a, c);
    if (a > d) swap(a, d);
    if (b > c) swap(b, c);
    if (b > d) swap(b, d);
    if (c > d) swap(c, d);
}
// Funkcija za kreiranje trocifrenog broja abc
int trocifreniBroj(int a, int b, int c) {
    return 100 * a + 10 * b + c; }

int main() {
    int broj;
    // citanje broja
    cin >> broj;
    // odredjivanje cifra broja
    int c0 = broj % 10;
    int c1 = broj / 10 % 10;
    int c2 = broj / 100 % 10;
    int c3 = broj / 1000;
    // uredjivanje cifara tako da je c0 <= c1 <= c2 <= c3
    sortiraj(c0, c1, c2, c3);
    // formiranje i ispis najmanjeg trocifrenog broja
    int x;
    if (c0 != 0)
        x = trocifreniBroj(c0, c1, c2);
    else if (c1 != 0)
        x = trocifreniBroj(c1, c0, c2);
    else if (c2 != 0)
        x = trocifreniBroj(c2, c0, c1);
    else
        x = trocifreniBroj(c3, c0, c1);
    cout << x << endl;
    return 0;
}

```

OBJAŠNENJE:

- Prenos argumenata je po referenci, da ne bi pravili nove promenljive nego samo upisujemo vrednosti

PRIMER 4: Armstrongov broj da li je uneti broj. (cifre na stepen n za n-tocifren broj i onda zbir svega da bude taj broj)

1002 -> Ne

370 -> Da **// $3^3 + 7^3 + 0^3 = 27 + 343 = 370$**

RESENJE:

```

#include <iostream>
#include <math.h>
using namespace std;

int brojCifara(int n) {
    return (n<10)? 1 : (brojCifara(n/10) + 1);
}

int stepen(int x, int n) {
    return (n!=1)? stepen(x, n-1) : 1;
}

int zvirStepenaCifara(int n, int k) {
    int zbir = 0;
    do {
        zbir += stepen (n%10, k);
        n /= 10;
    } while (n > 0);
    return zbir;
}

bool armstrongov(int n) {

    return zvirStepenaCifara(n, brojCifara(n)) == n;
}

int main() {
    int n;
    cin >> n;
    if (armstrongov) {
        cout << "DA" << endl;
    }
    else {
        cout << "NE" << endl;
    }
    return 0;
}

```