

Operatori u C-u - malo matematike 😊

Ne brinite, kad sam rekao malo matematike, mislio sam minimum. Uglavnom ce C da odradi sve za nas. U ovoj lekciji cete da naucite kako da prepoznate osnovne matematicke operatore i kako C gleda na njih, po pitanju tezine. (tbd.)

Osnovne aritmeticke operacije

Operator	Primjer	Opis
+	a + b	sabiranje
-	a - b	oduzimanje
*	a * b	mnozenje
/	a / b	dijeljenje
%	a % b	Moduo (ostatak pri dijeljenju)

```
ukupnaCijena = cijenaProizvoda + troskoviDostave - porez;  
novaCijena = staraCijena - -snizenje;
```

Izvorni kod: kod-150_primjer_sabiranja.c

```
#include<stdio.h>  
int godine_karaktera = 14;  
  
int main(){  
    printf("Za tri godine, Goku ce imati %d godina.\n", godine_karaktera +  
3);  
    return 0;  
}
```

Rezultat

Izvorni kod: kod-156_osnovne-matematicke-operacije.c

```
#include<stdio.h>

int main(){

    float x = 14.0;
    float y = 4.0;
    float rezultatRealnih;

    int a = 14;
    int b = 4;
    int rezultatCjelobrojnih;

    rezultatRealnih = x / y;
    printf("%.1f podijeljeno sa %.1f jednako je %.1f\n",
x,y,rezultatRealnih);
    rezultatCjelobrojnih = a / b;
    printf("%d podijeljeno sa %d jednako je %d \n",a,b,rezultatCjelobrojnih);
    rezultatRealnih = a / b;
    printf("%d podijeljeno sa %d jednako je %.1f \n", a,b,rezultatRealnih);
    rezultatCjelobrojnih = a % b;
    printf("%d moduo %d je jednako %d \n", a,b,rezultatCjelobrojnih);

    return 0;
}
```

Rezultat

```
14.0 podijeljeno sa 4.0 jednako je 3.5
14 podijeljeno sa 4 jednako je 3
14 podijeljeno sa 4 jednako je 3.0
14 moduo 4 je jednako 2
```

moduo se koristi samo kod rada sa cjelobrojnim
ostatakPriDijaljenju = a % b; /* 4 staje u 14 tri puta, do 14 ima jos 2 */

Izvorni kod: kod-157_izracunavanje-cijene-guma.c

```
#include<stdio.h>
#define POPUST .10
```

```

int main(){
    int brojGuma;
    float cijenaGume, cijenaSaPopustom, ukupnaCijena;

    printf("Koliko guma planirate kupiti? ");
    scanf(" %d", &brojGuma);
    printf("Kolika je cijena jedne gume (molim unesite cijenu u formatu xx.xx)? ");
    scanf(" %f", &cijenaGume);
    ukupnaCijena = brojGuma * cijenaGume;
    cijenaSaPopustom = ukupnaCijena - (ukupnaCijena * POPUST);

    printf("Ukupna cijena guma bez popusta %.2f \n", ukupnaCijena);
    printf("Cijena guma sa popustom %.2f ", cijenaSaPopustom);

    return 0;
}

```

Rezultat

```

Koliko guma planirate kupiti? 4
Kolika je cijena jedne gume (molim unesite cijenu u formatu xx.xx)? 33.33
Ukupna cijena guma bez popusta 133.32
Cijena guma sa popustom 119.99

```

Prioritet operatora (tbd. prioritet,vaznost)

```

int ukupno = 3 + 3 * 3; /* dodijeli vrijednost 12 promjenljivoj ukupno */

```

Tabela prioriteta operatora

Prioritet	Operator	Asocijativnost
1	()	od lijeva ka desno
2	*, /, %	od lijeva ka desno
3	+, -	od lijeva ka desno
4	<, <=, >, >=	od lijeva ka desno
5	==, !=	od lijeva ka desno

6	&&	od lijeva ka desno
7		od lijeva ka desno
8	? :	od desna ka lijevo
9	=, *=, /=, %= +=, -=	od desna ka lijevo
10	,	od lijeva ka desno

```

int ukupno = 3 + 7 * 2 / 2 % 3 + 11 - 3;
//          \ /
//          3 + 14 / 2 % 3 + 11 - 3
//          \ /
//          3 + 7 % 3 + 11 - 3
//          \ /
//          3 + 1 + 11 - 3
//          \ /
//          4 + 11 - 3
//          \ /
//          15 - 3
//          12

```

Uvodjenjem zagrada mozemo kontrolisati prioritet operatora

```

float srednja_vrijednost = 1 + 2 + 3 + 4 / 4; // necemo dobiti srednju
vrijednost
float srednja_vrijednost = (1 + 2 + 3 + 4) / 4; // dobicemo srednju
vrijednost

```

Dodjeljivanje jednakosti u jednom redu je moguće upravo zbog asocijativnosti operatora jednakosti, sa desna ka lijevo

```

a = 1; b = 1; c = 1; d = 1; e = 1; f = 1;
a = b = c = d = e = f = 1; // dodijeli vrijednost 1 promjenjivoj f; zatim
dodijeli vrijednost promjenjive f promjenjivoj e; itd

```

Vazno je primijetiti da svaki izraz u C-u (primjer $e = 1$;) daje neku konacnu vrijednost. Dakle $e = 1$; dodijeljuje vrijednost promjenljivoj e , ali isto tako e daje vrijednost 1, koja se kasnije moze koristiti, pohraniti u nekom drugom izrazu ili promjenljivoj.

```
int a;  
int b;  
  
int a = 3 * ( b = 2 ); //dodijeli vrijednost 2 promjenljivoj b; dodijeli  
vrijednost 6 promjenljivoj a
```

Izvorni kod: kod-158_rad-sa-operatorima-dodjele-i-zagradama.c

```
#include<stdio.h>  
  
int main(){  
  
    int bodovi1, bodovi2, bodovi3, bodovi4, rekordanBrojBodova;  
    float prosjekBodova, razlikaOdNajboljegProsjeka, procentualnaRazlika;  
  
    rekordanBrojBodova = 95;  
    bodovi1 = bodovi3 = 80;  
    bodovi2 = 75;  
  
    printf("Koliko bodova ste dobili na cetvrtom testu? [0 - 100] ");  
    scanf(" %d", &bodovi4);  
  
    prosjekBodova = (bodovi1 + bodovi2 + bodovi3 + bodovi4) / 4;  
  
    printf ("Prosjek bodova %.1f. \n", prosjekBodova);  
  
    razlikaOdNajboljegProsjeka = rekordanBrojBodova - prosjekBodova;  
    procentualnaRazlika = 100 * (razlikaOdNajboljegProsjeka /  
    rekordanBrojBodova);  
  
    printf("Vas broj bodova je za %.1f bodova ", razlikaOdNajboljegProsjeka);  
    printf("manji od rekordnog broja bodova (%d).\n", rekordanBrojBodova);  
    printf("Razlika izrazena u procentima iznosi %.1f %% ",  
    procentualnaRazlika);  
    printf("od rekordnog broja bodova. \n");  
  
    return 0;  
  
}
```

Rezultat

Koliko bodova ste dobili na cetvrtom testu? [0 - 100] 76
Prosjek bodova 77.0.
Vas broj bodova je za 18.0 bodova manji od rekordnog broja bodova (95).
Razlika izrazena u prncentima iznosi 18.9 % od rekordnog broja bodova.

Za razliku od drugih programskih jezika, C ima mnogo vise operatora ali znatno manje komandi.

Izvorni kod: kod-159_brojac.c

```
#include<stdio.h>

int main(){
    int brojac = 0;

    brojac = brojac + 1; // uvecaj brojac za 1
    printf("Brojac ima vrijednost %d \n", brojac);
    brojac = brojac + 1; // uvecaj brojac za 1
    printf("Brojac ima vrijednost %d \n", brojac);
    brojac = brojac + 1; // uvecaj brojac za 1
    printf("Brojac ima vrijednost %d \n", brojac);
    brojac = brojac + 1; // uvecaj brojac za 1
    printf("Brojac ima vrijednost %d \n", brojac);
    brojac = brojac + 1; // uvecaj brojac za 1
    printf("Brojac ima vrijednost %d \n", brojac);
    brojac = brojac + 1; // uvecaj brojac za 1
    printf("Brojac ima vrijednost %d \n", brojac);

    return 0;
}
```

Rezultat

```
Brojac ima vrijednost 1
Brojac ima vrijednost 2
Brojac ima vrijednost 3
Brojac ima vrijednost 4
Brojac ima vrijednost 5
Brojac ima vrijednost 6
Brojac ima vrijednost 7
```

Operatori dodjele

Pored osnovnog operatora dodjele `=`, postoje kombinovani operatori dodjele koji izvravaju vise operacija od same dodjele.

Kao sto smo imali u primjeru prvog brojaca

```
brojac = brojac + 1;
```

to mozemo napisati na nacin

```
brojac += 1;
```

Operator	Primjer	Skraceno od
<code>*</code>	<code>a *= 2</code>	<code>a = a * 2</code>
<code>/</code>	<code>a /= 2</code>	<code>a = a / 2</code>
<code>%</code>	<code>a %= 2</code>	<code>a = a % 2</code>
<code>+</code>	<code>a += 2</code>	<code>a = a + 2</code>
<code>-</code>	<code>a -= 2</code>	<code>a = a - 2</code>

Izvorni kod: kod-160_brojac2.c

```
#include<stdio.h>

int main(){
    int brojac = 0;

    brojac += 1; // uvecaj brojac za 1
    printf("Brojac ima vrijednost %d \n", brojac);
    brojac += 1; // uvecaj brojac za 1
    printf("Brojac ima vrijednost %d \n", brojac);
    brojac += 1; // uvecaj brojac za 1
    printf("Brojac ima vrijednost %d \n", brojac);
    brojac += 1; // uvecaj brojac za 1
    printf("Brojac ima vrijednost %d \n", brojac);
    brojac += 1; // uvecaj brojac za 1
    printf("Brojac ima vrijednost %d \n", brojac);
```



```
    brojac += 1; // uvecaj brojac za 1
    printf("Brojac ima vrijednost %d \n", brojac);
    brojac += 1; // uvecaj brojac za 1
    printf("Brojac ima vrijednost %d \n", brojac);

    return 0;
}
```

Rezultat

```
Brojac ima vrijednost 1
Brojac ima vrijednost 2
Brojac ima vrijednost 3
Brojac ima vrijednost 4
Brojac ima vrijednost 5
Brojac ima vrijednost 6
Brojac ima vrijednost 7
```

Unarni operatori ++, --

Operator	Primjer	Opis
++	a++	inkrement
		uvecava vrijednost za 1
--	a--	dekrement
		smanjuje vrijednost za 1

Izvorni kod: kod-161_brojac3.c

```
#include <stdio.h>

int main(){
    int brojac = 0;

    brojac++; // uvecaj brojac za 1
    printf("Brojac ima vrijednost %d \n", brojac);
    brojac++; // uvecaj brojac za 1
    printf("Brojac ima vrijednost %d \n", brojac);
    brojac++; // uvecaj brojac za 1
    printf("Brojac ima vrijednost %d \n", brojac);
}
```

```
    brojac++; // uvecaj brojac za 1
    printf("Brojac ima vrijednost %d \n", brojac);
    brojac++; // uvecaj brojac za 1
    printf("Brojac ima vrijednost %d \n", brojac);
    brojac++; // uvecaj brojac za 1
    printf("Brojac ima vrijednost %d \n", brojac);

    return 0;
}
```

Rezultat

```
Brojac ima vrijednost 1
Brojac ima vrijednost 2
Brojac ima vrijednost 3
Brojac ima vrijednost 4
Brojac ima vrijednost 5
Brojac ima vrijednost 6
```

Kastovanje (typecasting)

Typecasting - kastovanje je trenutna promjena varijable iz jednog tipa u drugi.

Format kastovanja

```
(tipPodatka)vrijednost
```

Izvorni kod: kod-165_kastovanje-varijable.c

```
#include <stdio.h>

int main(){

    int godine = 20;
    float tezina = 54.4;

    printf("Godine prije kastovanja %d \n", godine);
    printf("Tezina prije kastovanja %.1f \n", tezina);
    printf("Godine nakon kastovanja %.1f, tezina nakon kastovanja %d \n",
        (float)godine, (int)tezina);
    printf("Godine + tezina = %d ", godine + (int)tezina);
}
```

```
    return 0;  
}
```

Rezultat

```
Godine prije kastovanja 20  
Tezina prije kastovanja 54.4  
Godine nakon kastovanja 20.0, tezina nakon kastovanja 54  
Godine + tezina = 74
```

Uslov if .. else

Format uslova if

```
if(uslov)  
{ blok C komandi; }
```

U slucaju kad zelimo nasem programu dati mogucnost odlucivanja, ukljuciti logiku i donosenje odluka na osnovu odredjenih uslova dolazimo do potrebe za koristenjem **if** uslova.

Rame uz rame sa uslovom **if** idu i relacioni operatori, ciju tabelu mozete vidjeti u nastavku.

Relacioni operatori

Operator	Primjer	Opis
==	a == b	a je jednako b
>	a > b	a vece od b
<	a < b	a manje od b
>=	a >= b	a vece ili jednako od b
<=	a <= b	a manje ili jednako od b
!=	a != b	a razlicito od b

`==` nije isto sto i `=`. `==` ispituje da li je lijeva strana jednako desnoj, dok `=` predstavlja operator dodjele i ono sto je na lijevoj strani dodjeljuje promjenljivoj na desnoj strani.

Prilikom koristenja relacionih operatora, mozemo dobiti dvije vrijednosti, tacno (ili vrijednost 1) i netacno (ili vrijednost 0). Ovo se jos zove logicka 1 i logicka 0.

Izvorni kod: kod-166_uslov-if.c

```
#include <stdio.h>
#define TRENUTNA_GODINA 2020

int main(){
    int godinaRodjenja, godineKaraktera;

    printf("Koje godine ste rodjeni? \n");
    scanf(" %d", &godinaRodjenja);

    godineKaraktera = TRENUTNA_GODINA - godinaRodjenja;

    printf("Trenutno imate %d godina \n", godineKaraktera);

    if ((godinaRodjenja % 4) == 0){
        printf("Rodjeni ste u prestupnoj godini. \n");
    }
    return 0;
}
```

Rezultat

```
Koje godine ste rodjeni?
1987
Trenutno imate 33 godina
```

Rezultat

```
Koje godine ste rodjeni?
2000
Trenutno imate 20 godina
Rodjeni ste u prestupnoj godini.
```

Format uslova if .. else

```
if(uslov)
{ blok C komandi; }
else
{ blok C komandi; }
```

Izvorni kod: kod-167_uslov-if-else.c

```
#include <stdio.h>
#define TRENUTNA_GODINA 2020

int main(){
    int godinaRodjenja, godineKaraktera;

    printf("Koje godine ste rodjeni? \n");
    scanf(" %d", &godinaRodjenja);

    godineKaraktera = TRENUTNA_GODINA - godinaRodjenja;

    printf("Trenutno imate %d godina \n", godineKaraktera);

    if ((godinaRodjenja % 4) == 0){
        printf("Rodjeni ste u prestupnoj godini. \n");
    }
    else {
        printf("Niste rodjeni u prestupnoj godini. \n");
    }

    return 0;
}
```

Rezultat

```
Koje godine ste rodjeni?
1987
Trenutno imate 33 godina
Niste rodjeni u prestupnoj godini.
```

Rezultat

```
Koje godine ste rodjeni?
2000
Trenutno imate 20 godina
Rodjeni ste u prestupnoj godini.
```

Uslovni operatori

Izvorni kod: kod-167_pogadjanje-broja.c

```
#include <stdio.h>
#define TAJNIBROJ 4

int brojPokusaj;

int main(){
    printf("Pokusajte pogoditi tajni broj? [0 - 10] \n");
    scanf(" %d", &brojPokusaj);

    if (brojPokusaj == TAJNIBROJ){
        printf("Cestitamo pogodili ste. Tajni broj %d \n", TAJNIBROJ);
    }
    else if (brojPokusaj < TAJNIBROJ){ // mozemo reci else if ili samo if
        printf("Tajni broj je veci od unesenog. Molimo pokusajte ponovo.
\n");
    }
    else{
        printf("Tajni broj je manji od unesenog. Molimo pokusajte ponovo.
\n");
    }

    return 0;
}
```

Rezultat

```
Pokusajte pogoditi tajni broj? [0 - 10]
1
Tajni broj je veci od unesenog. Molimo pokusajte ponovo.
```

Rezultat

```
Pokusajte pogoditi tajni broj? [0 - 10]
6
Tajni broj je manji od unesenog. Molimo pokusajte ponovo.
```

Rezultat

```
Pokusajte pogoditi tajni broj? [0 - 10]
4
Cestitamo pogodili ste. Tajni broj 4
```

Logicki operatori

Ponekad relacioni operatori jednostavno nisu dovoljni i stvaraju visak koda, tu uskacu logicki opearatori. Kombinacijom logickih i relacionih operatora nas kod postaje citljiviji i jednostavniji.

Operator	Primjer	Opis
&&	a && b	i a i b daju tacno
	a b	ili a ili b daju tacno
!	!a	nije a, daje netacno

Logicki operatori se obicno koriste izmedju relacionih testova. Nekoliko primjera

```
if ((brojPokusaj < 4) && (brojPokusaj > 4))
```

```
if ((radniSati > 40) || (visinaPlate > 65000.00))
```

```
if (!(brojPokusaj < 4))
```

bi bilo isto kao da smo napisali

```
if ((brojPokusaj >= 4))
```

Izvorni kod: kod-168_logicki-operatori.c

```
#include <stdio.h>
#include <stdbool.h>
```

```
int main() {
    int brojUtakmica;
    int brojFudbalera;
    int maxBrojUtakmica = 4;
    int brojLopti;
    bool prijateljskaUtakmica = true;

    printf("Unesite broj odigranih utakmica? \n");
    scanf(" %d", &brojUtakmica);
    printf("Unesite broj fudbalera? \n");
    scanf(" %d", &brojFudbalera);
    printf("Unesite broj lopti? \n");
    scanf(" %d", &brojLopti);
    if ((brojUtakmica < maxBrojUtakmica) && (brojFudbalera >= 11)) {
        if (maxBrojUtakmica - brojUtakmica == 1){
            printf("Imate pravo odigrati jos %d utakmicu\n", maxBrojUtakmica
- brojUtakmica);
        }
        else{
            printf("Imate pravo odigrati jos %d utakmice. \n",
maxBrojUtakmica - brojUtakmica);
        }
    }
    else{
        printf("Odigrali ste maksimalan broj utakmica \n");
        printf("ili nemate dovoljno igraca. Minimalan broj je 11. \n");
        return 0;
    }

    brojLopti--;
    printf("Broj lopti %d\n", brojLopti);
    printf("Prijateljska utakmica %d \n", prijateljskaUtakmica);
    if ((!prijateljskaUtakmica) || (brojLopti <= 2)){
        printf("Moracete igrati na boljem terenu. \n");
    }
    else{
        printf("Utakmica moze da pocne. \n");
    }

    return 0;
}
```

Rezultat

```
Unesite broj odigranih utakmica?
2
Unesite broj fudbalera?
22
```



```
Unesite broj lopti?  
4  
Imate pravo odigrati jos 2 utakmice.  
Broj lopti 3  
Prijateljska utakmica 1  
Utakmica moze da pocne.
```

Rezultat

```
Unesite broj odigranih utakmica?  
2  
Unesite broj fudbalera?  
22  
Unesite broj lopti?  
3  
Imate pravo odigrati jos 2 utakmice.  
Broj lopti 2  
Prijateljska utakmica 1  
Moracete igrati na boljem terenu.
```

Izvorni kod: kod-169_odgovor-da-ili-ne.c

```
#include <stdio.h>  
char pitanje;  
int main(){  
    printf("Da li zelite postaviti pitanje? [Yy/Nn] \n");  
    scanf(" %c", &pitanje);  
  
    if (pitanje == 'Y' || pitanje == 'y'){  
        printf("Postavite pitanje.\n");  
    }  
    else if (pitanje == 'N' || pitanje == 'n'){  
        printf("Hvala sto ste ucestvovali\n");  
    }  
    else{  
        printf("Niste izabrali pravi karakter.\n");  
    }  
  
    return 0;  
}
```

Rezultat

```
Da li zelite postaviti pitanje? [Yy/Nn]
Y
Postavite pitanje.
```

Rezultat

```
Da li zelite postaviti pitanje? [Yy/Nn]
y
Postavite pitanje.
```

Rezultat

```
Da li zelite postaviti pitanje? [Yy/Nn]
N
Hvala sto ste ucestvovali
```

Rezultat

```
Da li zelite postaviti pitanje? [Yy/Nn]
n
Hvala sto ste ucestvovali
```

Rezultat

```
Da li zelite postaviti pitanje? [Yy/Nn]
A
Niste izabrali pravi karakter.
```

Prioritet logickih operatatora

Ako imamo

```
if (godine < 20 || plata < 1200 && radniSati > 15) {
```

C ce to da vidi na sledeci nacin

```
if ((godine < 20) || ((plata < 1200) && (radniSati > 15))) {
```

Uslovni operator (zamjena za if .. else)

Za razliku od vecine operatora koji zahtijevaju po dva argumenta, **Uslovni operator** zahtijeva tri argumenta.

Format uslovnog operatora

```
uslov ? tacanIzraz : netacanIzraz;
```

Izvorni kod: kod-170_uslovni-operator.c

```
#include <stdio.h>

int main(){
    int a, b;

    printf("Unesite vrijednost a? \n");
    scanf(" %d", &a);
    printf("Unesite vrijednost b? \n");
    scanf(" %d", &b);
    (a < b) ? printf("a je manje od b\n") : printf("a je vece od b\n");

    return 0;
}
```

Rezultat

```
Unesite vrijednost a?
3
Unesite vrijednost b?
4
a je manje od b
```

Rezultat

```
Unesite vrijednost a?
4
```

Unesite vrijednost b?
3
a je vece od b

Izvorni kod: kod-171_uslovni-operator.c

```
#include <stdio.h>

int main(){
    int godine;
    char slovo;
    printf("Unesite broj godina: \n");
    scanf(" %d", &godine);

    slovo = (godine > 2) ? 'e' : 'u';

    printf("Imate %d godin%c\n", godine, slovo);

    return 0;
}
```

Rezultat

Unesite broj godina:
1
Imate 1 godinu

Rezultat

Unesite broj godina:
3
Imate 3 godine

Izvorni kod: kod-172_dijeljivost-broja.c

```
#include <stdio.h>

int main(){
    int broj;
```

```
printf("Unesite broj od 1 do 100\n");
printf("a ja cu provjeriti da li je djeljiv sa \n");
printf("brojevima od 2 do 9 \n");
scanf(" %d", &broj);

printf("%d %s djeljiv sa 2.\n", broj, (broj % 2 == 0) ? ("je") :
("nije"));
printf("%d %s djeljiv sa 3.\n", broj, (broj % 3 == 0) ? ("je") :
("nije"));
printf("%d %s djeljiv sa 4.\n", broj, (broj % 4 == 0) ? ("je") :
("nije"));
printf("%d %s djeljiv sa 5.\n", broj, (broj % 5 == 0) ? ("je") :
("nije"));
printf("%d %s djeljiv sa 6.\n", broj, (broj % 6 == 0) ? ("je") :
("nije"));
printf("%d %s djeljiv sa 7.\n", broj, (broj % 7 == 0) ? ("je") :
("nije"));
printf("%d %s djeljiv sa 8.\n", broj, (broj % 8 == 0) ? ("je") :
("nije"));
printf("%d %s djeljiv sa 9.\n", broj, (broj % 9 == 0) ? ("je") :
("nije"));

return 0;
}
```

Rezultat

```
Unesite broj od 1 do 100
a ja cu provjeriti da li je djeljiv sa
brojevima od 2 do 9
64
64 je djeljiv sa 2.
64 nije djeljiv sa 3.
64 je djeljiv sa 4.
64 nije djeljiv sa 5.
64 nije djeljiv sa 6.
64 nije djeljiv sa 7.
64 je djeljiv sa 8.
64 nije djeljiv sa 9.
```

sizeof() operator

Za pronalazenje tacne vrijednosti koju neki tip podatka zauzima u memoriji, koristimo **sizeof()** operator. Iako danasnji kompajleri standardno koriste velicinu od 4 byte-a za pohranjivanje integera i dalje ima onih koji imaju drugacije vrijednosti.

Izvorni kod: kod-173_sizeof-operator.c

```
#include <stdio.h>

char nazivKaraktera[] = "Goku";
char nazivPlanete[8] = "Zemlja";
int godineKaraktera = 14;
float tezinaKaraktera = 44.34;

int main(){
    printf("Velicina promjenjive nazivKaraktera u memoriji %ld \n",
        sizeof(nazivKaraktera));
    printf("Velicina promjenjive nazivPlanete u memoriji %ld \n",
        sizeof(nazivPlanete));
    printf("Velicina promjenjive godineKaraktera u memoriji %ld \n",
        sizeof(godineKaraktera));
    printf("Velicina promjenjive tezinaKaraktera u memoriji %ld \n",
        sizeof(tezinaKaraktera));

    return 0;
}
```

Rezultat

```
Velicina promjenjive nazivKaraktera u memoriji 5
Velicina promjenjive nazivPlanete u memoriji 8
Velicina promjenjive godineKaraktera u memoriji 4
Velicina promjenjive tezinaKaraktera u memoriji 4
```

Vazno je napomeniti da duzina stringa i velicina (sizeof()) nisu isto, dakle ovo su dvije potpuno druge vrijednosti. Duzina je broj byte-ova koja ne ukljucuje null zero i izracunava se upotrebom funkcije strlen(), prilikom cega moramo koristiti **string.h** header. Velicina (sizeof()) stringa predstavlja broj karaktera potreban da se string pohrani, ukljucujuci null zero.

Izvorni kod: kod-172_dijeljivost-broja.c**Rezultat**

Unesite broj godina:

3

Imate 3 godine

```
**`Izvorni kod: kod-172_dijeljivost-broja.c`**  
```c
```

### Rezultat

Unesite broj godina:

3

Imate 3 godine

```
`Izvorni kod: kod-172_dijeljivost-broja.c`
```c
```

Rezultat

Unesite broj godina:

3

Imate 3 godine

```
**`Izvorni kod: kod-172_dijeljivost-broja.c`**  
```c
```

### Rezultat

Unesite broj godina:

3

Imate 3 godine

```
`Izvorni kod: kod-172_dijeljivost-broja.c`
```c
```

Rezultat

Unesite broj godina:

3

Imate 3 godine

