

INFORMATIČKI KLUB
FUTURA

```
#include<stdio.h> C  
int main()  
{ RADIONICE PROGRAMIRANJA  
  printf("Hello World!");  
  ZA SREDNJE ŠKOLE  
  return 0;  
}
```

RADIONICE PROGRAMIRANJA ZA SREDNJE ŠKOLE - 1. RADIONICA

Krunoslav Žubrinić, Informatički klub FUTURA
Dubrovnik, 25. listopada 2014.



Creative Commons



slobodno smijete:

- **dijeliti** — umnožavati, distribuirati i javnosti priopćavati djelo
- **remiksirati** — prerađivati djelo



pod slijedećim uvjetima:

- **imenovanje**. Morate priznati i označiti autorstvo djela na način kako je specificirao autor ili davatelj licence (ali ne način koji bi sugerirao da Vi ili Vaše korištenje njegova djela imate njegovu izravnu podršku).
- **nekomercijalno**. Ovo djelo ne smijete koristiti u komercijalne svrhe.
- **dijeli pod istim uvjetima**. Ako ovo djelo izmijenite, preoblikujete ili stvarate koristeći ga, preradu možete distribuirati samo pod licencom koja je ista ili slična ovoj.



U slučaju daljnjeg korištenja ili distribuiranja morate drugima jasno dati do znanja licencne uvjete ovog djela. Najbolji način da to učinite je linkom na ovu internetsku stranicu.

Od svakog od gornjih uvjeta moguće je odstupiti, ako dobijete dopuštenje nositelja autorskog prava.

Ništa u ovoj licenci ne narušava ili ograničava autorova moralna prava.

Tekst licence preuzet je s <http://creativecommons.org/>.

Sadržaj

- Računalni program i algoritam
- Uvod u programski jezik C
- Deklaracija varijabli, unos i ispis
- Tipovi podataka
- Korištenje matematičkih funkcija u C-u
- Grananje

Računalni program



- Program je skup naredaba čijim se izvršenjem obavlja željeni posao
 - Naredba je temeljni element programa
 - Algoritam je uputa kako riješiti neki problem
 - Obično se sastoji od niza koraka
 1. Zagrij vodu
 2. Stavi vrećicu čaja
 3. Ostavi stajati 5 minuta
 4. Posluži
- Je li ovaj algoritam dovoljno detaljan za računalo?

Opis algoritma



□ Algoritam se može opisati na različite načine:

- Blok dijagramom
- Pseudokodom



Početak

Unesi duljinu stranice kvadrata

Duljina ≤ 0 :

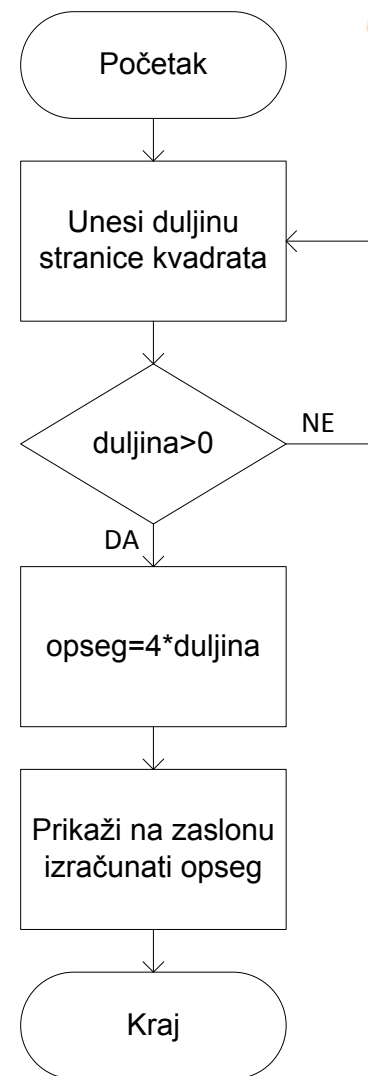
Vrati se na unos duljine stranice

Duljina > 0 :

Izračunaj $opseg = 4 * duljina$

Prikaži izračunati opseg

Kraj



Programski jezik C



Q : Zbog čega je programski jezik dobio ime C?

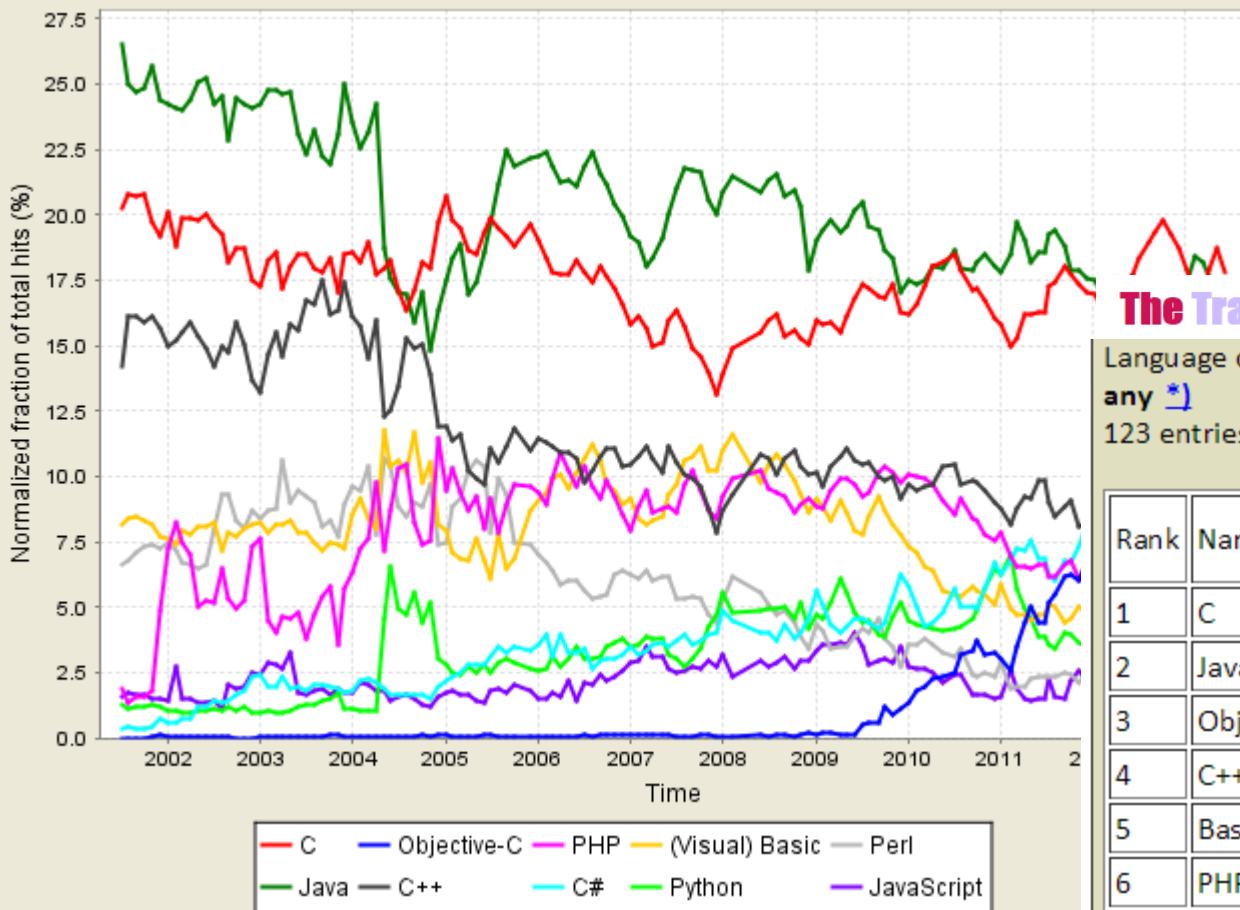
A : Jer je prije njega bio programski jezik B!

- Jedan od najvažnijih programskih jezika u povijesti komercijalne računalne industrije.
 - „Djed” većine suvremenih programskih jezika.
 - Mnogi od njih su naslijedili njegovu sintaksu:
 - C++, Java, JavaScript, C#, PHP, Objective-C,...
 - Jezik opće namjene i vrlo niske razine.
 - Koristi se kada je ključna brzina izvođenja i/ili prenosivost.
- Unatoč starosti, još uvijek jedan od najpopularnijih programskih jezika.

Popularnost programskih jezika



TIOBE Programming Community Index



The Transparent Language Popularity Index

Language category:

any *)

123 entries.

Rank	Name	Share	Last month's share	Last year's share
1	C	17.668%	15.868%	16.825%
2	Java	14.720%	15.450%	20.381%
3	Objective-C	8.230%	8.516%	9.221%
4	C++	6.770%	7.544%	7.912%
5	Basic	5.457%	5.955%	7.592%
6	PHP	4.401%	4.144%	4.247%
7	Python	3.658%	3.363%	3.616%
8	C#	3.269%	3.444%	4.598%
9	Perl	2.566%	2.455%	2.459%
10	Ruby	1.918%	1.392%	1.576%

Izrada izvršnog programa u C-u



```
#include <stdio.h>

int main()
{
    printf("Dobar dan!");
    system("pause");
    return 0;
}
```

1

C datoteka sadrži izvorni C programski kod razumljiv čovjeku. Programski kod možete pisati u bilo kojem programu za unos teksta.



```
File Edit Window Help Compile
> gcc dobar_dan.c -o dobar_dan
```

2

Izvorni kod se prevodi pomoću programa prevoditelja (*compiler*). Prevoditelj pronalazi i prijavljuje greške, a kada je sve u redu stvara izvršni kod.



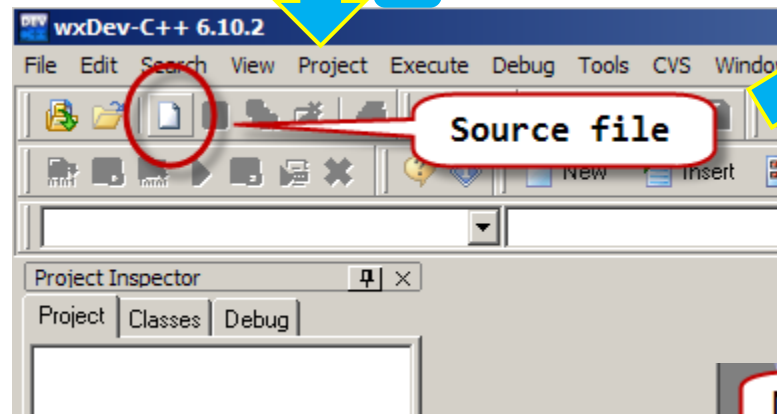
3

Izvršni kod (**.exe** datoteka na MS Win) razumljiv je računalu, može se izvesti na određenom OS-u. Sam izvršni kod je niz binarnih znamenki.

Izrada izvršnog programa (Dev-C++)



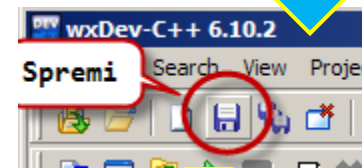
1



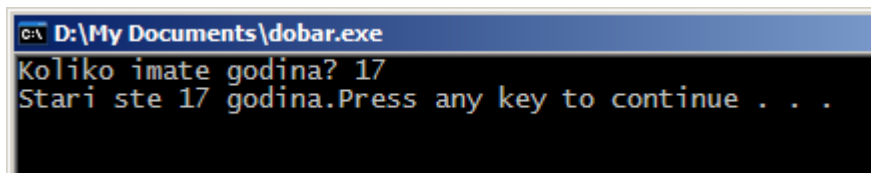
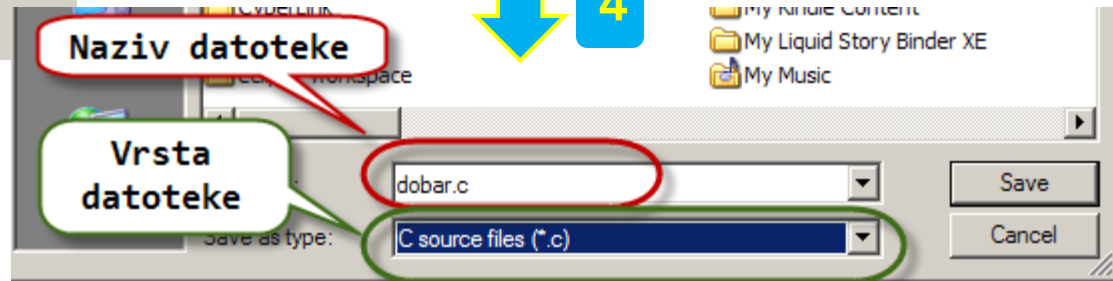
2

```
#include <stdio.h>
int main(){
    int starost;
    printf("Koliko imate godina? ");
    scanf("%d", &starost);
    printf("Stari ste %d godina.", starost);
    system ("pause");
    return 0;
}
```

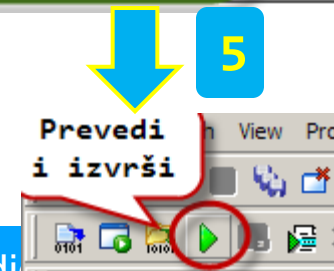
3



4

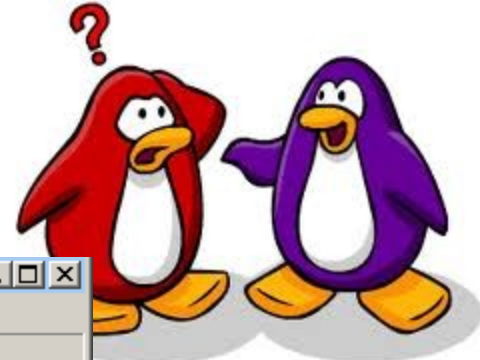


6



5

Ako se dogodi greška?



Broj linije u datoteci programskog koda u kojoj se dogodila greška. OPREZ! često je niz grešaka posljedica jedne greška koja se dogodila. Kod istraživanja grešaka (debugiranje), krenite od prve, ispravite ju pa pokušajte ponovno prevesti i izvesti program.

```
#include <stdio.h>
int main()
{
    in a,opseg;
    printf("Unesite duljinu stranice kvadrata: ");
    scanf("%d", a);
    opseg=4*a
    printf("Opseg pravokutnika je %d.\n", opseg);
    system ("pause");
}
```

Compiler prozor prikazuje informacije o greškama tijekom prevođenja programa.

Line	File	Message
4	d:\My Documents\UNIDU\srednjosko ..	In function `main': 'in' undeclared (first use in this function) (Each undeclared identifier is reported only once for each function it appears in.)
4	d:\My Documents\UNIDU\srednjosko ..	syntax error before "a"
6	d:\My Documents\UNIDU\srednjosko ..	'a' undeclared (first use in this function)
7	d:\My Documents\UNIDU\srednjosko ..	'opseg' undeclared (first use in this function)
8	d:\My Documents\UNIDU\srednjosko ..	syntax error before "printf"

Dvoklik na redak s porukom o grešci odvesti će vas na tu liniju programskog koda

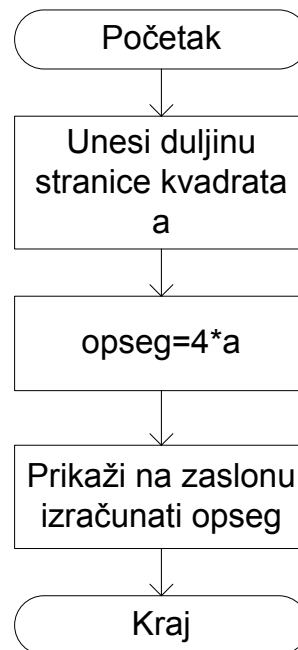
Opis greške.

4: 1 Insert 12 Lines in file

Opseg kvadrata



Algoritam



Rezultat izvođenja



```
C:\> D:\My Documents\UNIDU\srednjoskolci\kvadrat1.exe
Unesite duljinu stranice: 3
Opseg kvadrata je 12.
Press any key to continue . . .
```



Anatomija programa

C je jako mali jezik i većina funkcionalnosti se izvodi pomoću vanjskih biblioteka.

#include naredba programu kaže koju vanjsku biblioteku (*header*) treba pogledati za traženu funkcionalnost.

stdio.h sadrži opis funkcija za ulazno-izlazne operacije.

stdlib.h sadrži opis standardnih često korištenih funkcija (primjerice **system**)

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

```
#include <stdlib.h>
```

```
int main()
```

Main funkcija je početna funkcija programa.

```
{
```

Točka-zarez na kraju svakog retka!

```
int str, opseg;
```

Rezervira se memorijski prostor za pohranu 2 cijela broja.

```
printf("Unesite duljinu stranice:");
```

Ispis fiksnog teksta na zaslon

```
scanf("%d", &str);
```

Unos cijelog broja. Operator **&** daje adresu na kojoj je varijabla **str** smještena.

```
opseg = str * 4;
```

Izračun opsega.

```
printf("Opseg kvadrata je %d.\n", opseg);
```

Ispis teksta i opsega na zaslonu
%d ispisuje cijeli broj

```
system("pause");
```

Pauza u izvođenju programa (kako bismo vidjeli rezultat izvođenja).

```
return 0;
```

Main funkcija vraća cijeli broj pozivatelju (operacijskom sustavu).
0 znači da se program uspješno izvršio.

```
}
```

Tijelo funkcije je omeđeno vitičastim zagradama

Opseg kvadrata

```
#include <stdio.h>
#include <stdlib.h>
int main()
{
    int str, opseg;
    printf("Unesite duljinu stranice:");
    scanf("%d", &str);
    opseg = str * 4;
    printf("Opseg kvadrata je %d.\n", opseg);
    system ("pause");
    return 0;
}
```



1. zadatak: Gledatelji



Toni je otišao s prijateljem Dinom na košarkašku utakmicu. Za vrijeme utakmice je, po običaju, Toni stalno prigovarao Dinu. Na poluvremenu je njihov sukob dosegao vrhunac.

Toni: “Jao, Dino! Pa ti uopće ne pratiš utakmicu!”

Dino (nezainteresirano): “Nije istina, ja obožavam košarku!”

Toni: “Lažljivče! Došao si ovdje samo gledati navijačice! Sumnjam da uopće znaš pravila.”

Dino: “Nije istina! Ja odlično znam pravila!”

Toni: “Hajdemo se onda okladiti da ne znaš reći koliki je rezultat ako je prva ekipa zabila T1 ‘trica’, D1 ‘dvica’ i S1 ‘slobodnjaka’, a druga ekipa T2, D2 i S2 istih!”

Pomozite Dinu izračunati rezultat!

Za one koji ne znaju pravila: zabijanje jedne ‘trice’ vrijedi 3 boda, ‘dvica’ 2 boda, a ‘slobodnjaka’ 1 bod.

ULAZNI PODACI

- 6 cijelih brojeva T1, D1, S1, T2, D2 i S2.

IZLAZNI PODACI

- Rezultat ispišite u obliku A:B, gdje je A broj bodova prve, a B broj bodova druge ekipe.

PRIMJER:

ULAZ: 1 2 3 1 2 4

2 2 2 1 4 3

IZLAZ: 10:11

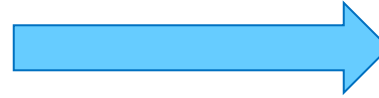
12:14

Vrijeme



Rješenje 1. zadatka

Algoritam



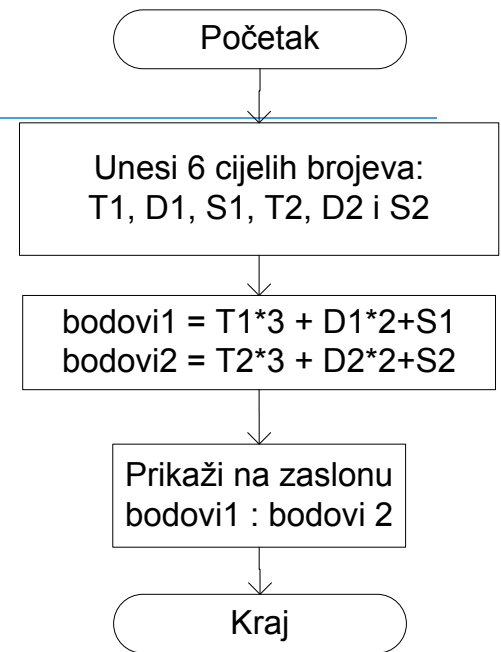
```
#include <stdio.h>
#include <stdlib.h>
int main()
```

```
{
    int t1, d1, s1, t2, d2, s2, bod1, bod2;
    printf("Unesite 6 cijelih brojeva: T1, D1, S1, T2, D2 i S2:");
    scanf("%d %d %d %d %d %d", &t1, &d1, &s1, &t2, &d2, &s2);
    bod1 = t1*3 + d1*2 + s1;
    bod2 = t2*3 + d2*2 + s2;
    printf("Rezultat je %d:%d.\n", bod1, bod2);
    system("pause");
    return 0;
}
```

Programski kod



Rezultat izvođenja



gledatelji.c



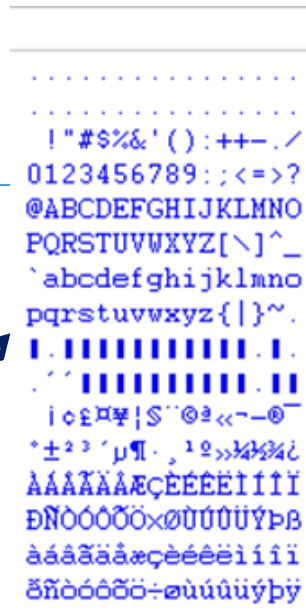
```
Unesite 6 cijelih brojeva: T1, D1, S1, T2, D2 i S2:1 2 3 1 2 4
Rezultat je 10:11.
```



Osnovni tipovi podataka

Tip podatka	Uobičajeni raspon vrijednosti
char	0 → 255
int	-2.147.483.648 → +2.147.483.647
float	$1 \times 10^{-37} \rightarrow 1 \times 10^{37}$
double	$1 \times 10^{-308} \rightarrow 1 \times 10^{308}$

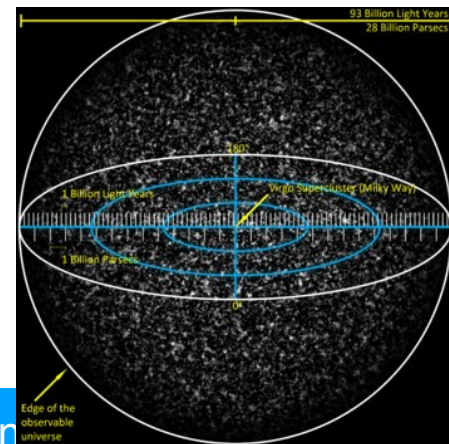
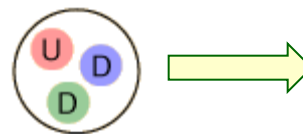
ASCII kodovi znakova



Cijeli brojevi u intervalu od cca minus do plus dvije milijarde. Npr. broj stanovnika neke države (ali ne i čitavog svijeta!)

Sve trenutno poznate fizičke veličine izražene u metrima:

- od promjera neutrona (2×10^{-23})
- do promjera vidljivog svemira (1×10^{26})



Ako trebamo iskazati brojeve manje od 10^{-37} ili veće od 10^{37}



Osnovni operatori

Prioritet operatora:

++, **--**, **!**

*****, **/** i **%**

+ i **-**

==, **<**, **>** **<=**, **>=**, **!=**

&& i **||**

□ aritmetički

- Zbrajanje $c = a + b$
- Oduzimanje $c = a - b$
- Množenje $c = a * b$
- Dijeljenje $c = a / b$
- Modulo $c = a \% b$ (ostatak dijeljenja)
- Inkrementiranje $a++$ (isto što i $a = a + 1$)
- Dekrementiranje $a--$ (isto što i $a = a - 1$)

□ logički

- I $a \&\& b$ (istinito ako su **i a i b** istiniti)
- ILI $a \|\| b$ (istinito ako su **ili a ili b** istiniti)
- NE **!**a (**negacija** istinito ako je a neistinit)

□ uspoređivanja

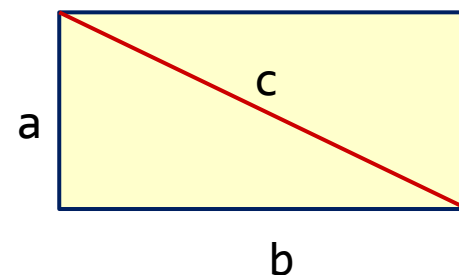
- $a==b$, $a<b$, $a>b$ $a<=b$, $a>=b$, $a!=b$


Duljina hipotenuze



- Podijelimo pravokutnik na 2 jednaka trokuta
- Kako ćemo izračunati duljinu hipotenuze?

$$c = \sqrt{a^2 + b^2}$$

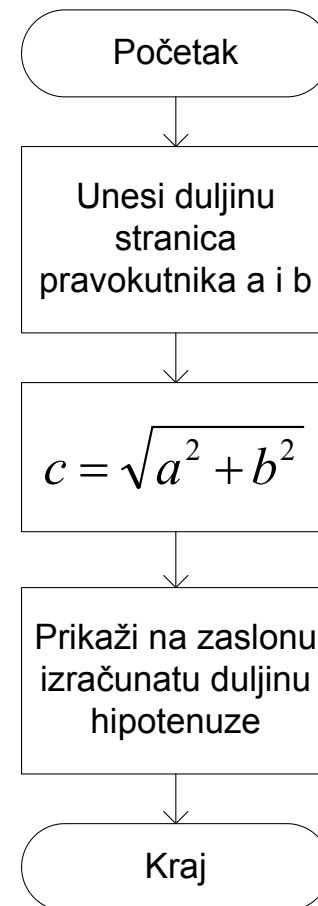
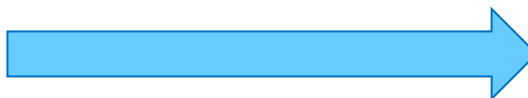


- Matematičke funkcije su definirane u datoteci **math.h**  `#include <math.h>`
 - Vađenje korijena: **sqrt(broj)**
 - Potenciranje: **pow(baza, potencija)**
 - Apsolutna vrijednost: **fabs(broj)...**
- Te funkcije rade s realnim brojevima (float, double)

Duljina hipotenuze



Algoritam



Rezultat
izvođenja

```
C:\ D:\My Documents\UNIDU\srednjoskolci\zadnje\  
Unesite duljine stranica a i b: 4 6  
Duljina hipotenuze je 7.21  
Press any key to continue . . . .
```



Anatomija programa

math.h sadrži opis matematičkih funkcija (sqrt, pow, fabs, log, log10,...)

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
```

Rezultat ne mora biti cijeli broj!
Matematičke funkcije definirane u math.h rade s realnim brojevima (**float** i **double**)

Unos sa standardnog ulaza (tipkovnice).
Oznake počinju znakom %
%d – unos cijelog broja (int)
%f – unos realnog broja (float)
%c – unos jednog znaka (char)

```
int main(){
    int a, b;
    float c;
    printf("Unesite duljine stranica a i b: ");
    scanf("%d %d", &a, &b);
    c = sqrt((float)a*a + b*b);
    printf("Duljina hipotenuze je %.2f\n", c);
    system ("pause");
    return 0;
}
```

sqrt(izraz) funkcija koja vadi korijen izraza unutar zagrada. Funkcija prima i vraća realan broj. Ako su parametri cijeli brojevi, treba barem jedan od njih pretvoriti u realan broj.

Ne zaboravite točku-zarez na kraju svakog retka

%.2f ispisuje realni broj (s decimalama), zaokružen na **2 decimale**.
Ako napišemo samo **%f**, ispisati će realni broj na defaultnih 6 decimala.

Duljina hipotenuze

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
int main(){
    int a, b;
    float c;
    printf("Unesite duljine stranica a i b: ");
    scanf("%d %d", &a, &b);
    c = sqrt((float)a*a + b*b);
    printf("Duljina hipotenuze je %.2f\n", c);
    system ("pause");
    return 0;
}
```



2. Zadatak - Brzina



Markov otac obožava brzo voziti, što Marka jako ljuti jer se boji policije. Stoga Marko tijekom vožnje stalno prati znakove ograničenja brzine pokraj ceste i brzinu kojom vozi njegov otac, te ocu redovito govori koliko vozi iznad dozvoljene brzine.

Umorio se od računanja pa te moli za pomoć.

Tvoj zadatak je napisati program koji će na osnovu maksimalne dozvoljene brzine i brzine kojom se kreće Markov otac, izračunati za koliko posto je Markov otac premašio maksimalnu dozvoljenu brzinu.

ULAZNI PODACI:

- cijeli broj M, maksimalna dozvoljena brzina vožnje.
- cijeli broj N, brzina kojom vozi Markov otac. Brzina je uvijek veća od maksimalno dozvoljene brzine vožnje.

IZLAZNI PODACI:

- Decimalni broj P, postotak za koliko je Markov otac premašio maksimalnu dozvoljenu brzinu.

$$P = ((N - M) / M) * 100$$

PRIMJER:

ULAZ: 60 80

50 90

IZLAZ: 33.33

80.00

Vrijeme



Rezultat bez decimala?



- Sve ste „dobro” napisali ali dobivate rezultat bez decimala?

```
Unesite maksimalnu dozvoljenu brzinu M i stvarnu brzinu N:60 80
Brzina je premasena za 0.00%.
Press any key to continue . . .
```

- Problem je što program ne pokazuje ispravno znamenke iza decimalne točke $20/60 = 0.33333$
 - Oba broj su cijeli (int) pa se odrađuje „cjelobrojno” dijeljenje (rezultat je cijeli broj)
 - Rješenje je **pretvorba cjelobrojnog podatka u realni**
- Da bi se dobio realni rezultat, potrebno je u realni broj pretvoriti barem 1 cijeli broj iz izraza

$$p = ((\text{float})n - m) / m$$

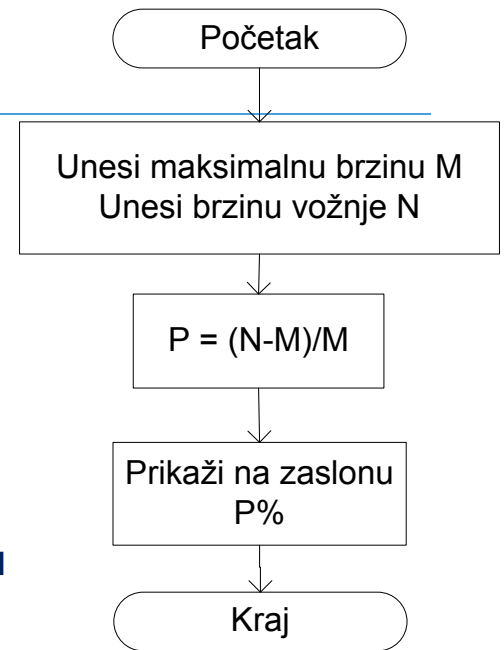
Rješenje 2. zadatka

```
#include <stdio.h>
#include <stdlib.h>
int main() {
    int m, n;
    float p;
    printf("Unesite maksimalnu dozvoljenu brzinu
           M i stvarnu brzinu N:");
    scanf("%d %d", &m, &n);
    p = (((float)n - m) / m) * 100;
    printf("Brzina je premasena za %.2f%%.\n", p);
    system("pause");
    return 0;
}
```

Programski kod

Rezultat izvođenja

Algoritam



brzina.c

Znak % služi za formatiranje ispisa (sjetite se %d, %f,...)
Želite li na zaslonu ispisati znak %, **trebate ga napisati dva puta: %%**



```
Unesite maksimalnu dozvoljenu brzinu M i stvarnu brzinu N:60 80
Brzina je premasena za 33.33%.
```


Kvadratna jednadžba

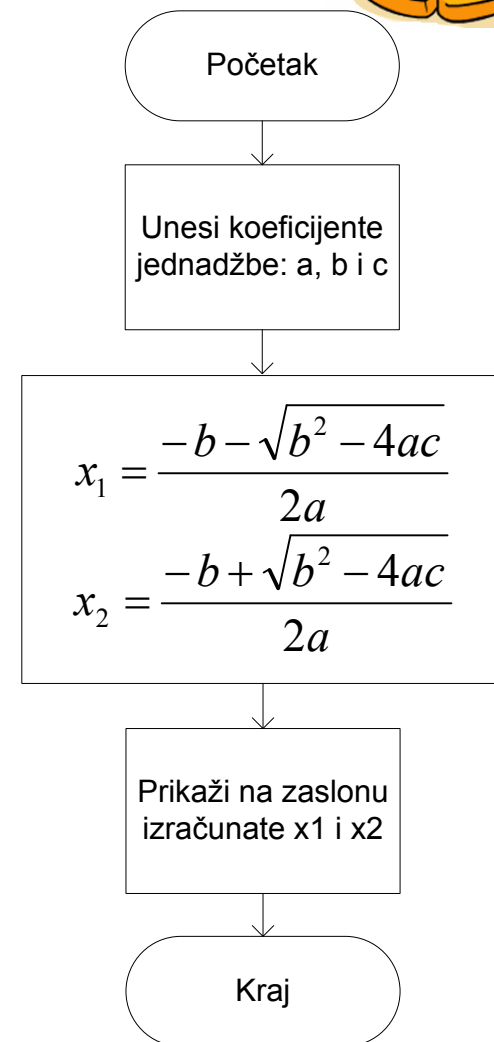
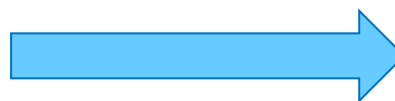


- Opća kvadratna jednadžba je izraz

$$ax^2 + bx + c = 0; a \neq 0$$

- Za slučaj kada je diskriminanta (izraz $b^2 - 4ac$) veća od nule ima dva rješenja: x_1 i x_2

Algoritam



Rezultat izvođenja



```
Unesite koeficijente a b i c: 2 4 -4
Rjesenja kvadratne jednadzbe su -2.73 i 0.73
Press any key to continue . . .
```



Anatomija programa

math.h sadrži opis matematičkih funkcija (sqrt, pow, fabs, log, log10,...)

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
int main(){
    int a, b, c;
    float x1, x2;
    printf("Unesite koeficijente a b i c: ");
    scanf("%d %d %d", &a, &b, &c);
    x1 = (-1*b - sqrt(pow((float)b,2)-4*a*c)) / (2*a);
    x2 = (-1*b + sqrt(pow((float) b,2)-4*a*c)) / (2*a);
    printf("Rjesenja su %.2f i %.2f\n", x1, x2);
    system("pause");
    return 0;
}
```

Rezultat ne mora biti cijeli broj!
Matematičke funkcije definirane u math.h rade s realnim brojevima (**float** i **double**)

sqrt(*izraz*) funkcija koja vadi korijen izraza unutar zagrada.
pow(*baza*,*stupanj*) funkcija potencira prvi parametar na potenciju drugoga.

%.2f ispisiuje realni broj (s decimalama), zaokružen **na 2 decimale**.
Ako napišemo samo **%f**, ispisiati će realni broj na defaultnih 6 decimala.
%s ispisiuje niz znakova, **%c** jedan znak, a **%d** cijeli broj

Kvadratna jednačina v1

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
int main(){
    int a, b, c;
    float x1, x2;
    printf("Unesite koeficijente a b i c: ");
    scanf("%d %d %d", &a, &b, &c);
    x1 = (-1*b - sqrt(pow((float)b,2)-4*a*c)) / (2*a);
    x2 = (-1*b + sqrt(pow((float)b,2)-4*a*c)) / (2*a);
    printf("Rjesenja su %.2f i %.2f\n", x1, x2);
    system("pause");
    return 0;
}
```



Kvadratna jednačba v2



□ Što ako netko unese nulu na mjesto prvog koeficijenta?

PAZI!

$a == 0$ uspoređuje vrijednost a s nulom

$a = 0$ pridružuje nulu varijabli a

□ Jednačba postaje linearna: $bx + c = 0$ $x = \frac{-c}{b}$

...

```
if (a == 0) {  
    x = -1 * (float)c / b;  
}  
else {  
    x1 = (-1*b - sqrt(pow((float)b,2)-4*a*c)) / (2*a);  
    x2 = (-1*b + sqrt(pow((float)b,2)-4*a*c)) / (2*a);  
}
```

Izraz čija istinitost se ispituje.

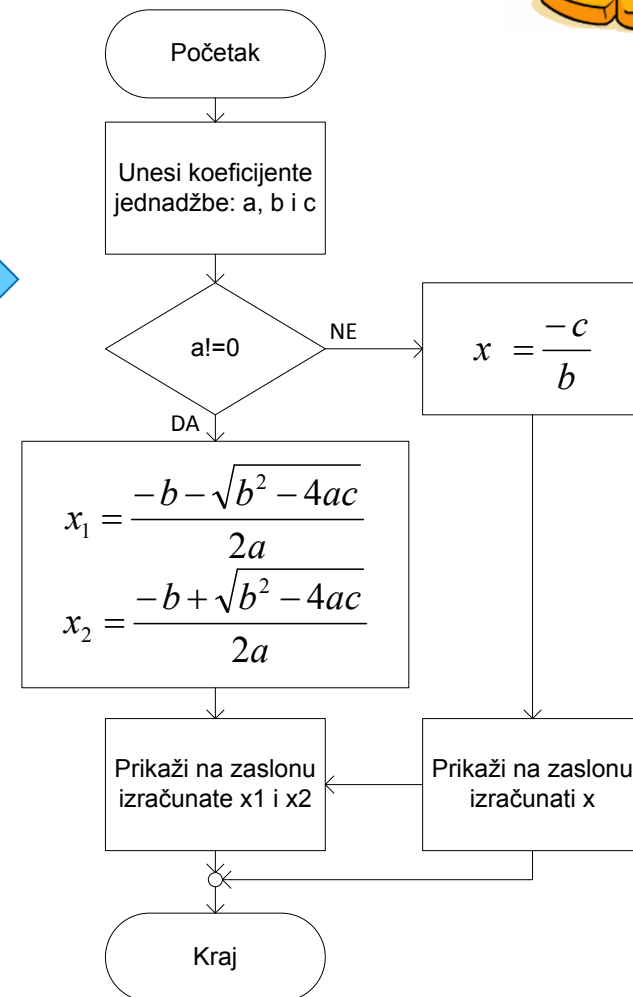
Ako je izraz u zagradama iza **if** istinit, rješava se linearna jednačba i izvršava **ovaj blok instrukcija**.

... inače se rješava kvadratna jednačba i izvršava blok instrukcija iza **else**

Kvadratna jednačina v2



Algoritam



Rezultat izvođenja



```
Unesite koeficijente a b i c: 0 6 -3  
Rjesenje jednačine je 0.50  
Press any key to continue . . .
```



Anatomija programa

math.h sadrži opis matematičkih funkcija (sqrt, pow, fabs, log, log10,...)

Rezultat ne mora biti cijeli broj!
Matematičke funkcije definirane u math.h rade s realnim brojevima (**float** i **double**)

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

int main(){
    int a, b, c;
    float x1, x2;
    printf("Unesite koeficijente a b i c: ");
    scanf("%d %d %d", &a, &b, &c);
    if (a == 0) {
        x1 = ((-1 * (float)c) / b);
        printf("Rjesenje jednadzbe je %.2f\n", x1);
    }
    else {
        x1 = (-1*b - sqrt(pow((float)b,2)-4*a*c)) / (2*a);
        x2 = (-1*b + sqrt(pow((float)b,2)-4*a*c)) / (2*a);
        printf("Rjesenja jednadzbe su %.2f i %.2f\n", x1, x2);
    }
    system("pause");
    return 0;
}
```

Provjera je li prvi koeficijent jednak nuli

... ako jest, računa se linearna jednadžba

... inače se računaju koeficijenti kvadratne jednadžbe

Ne zaboravite točku-zarez na kraju svakog retka

Kvadratna jednačina v2

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
int main(){
    int a, b, c;
    float x1, x2;
    printf("Unesite koeficijente a b i c: ");
    scanf("%d %d %d", &a, &b, &c);
    if (a == 0) {
        x1 = ((-1 * (float)c) / b);
        printf("Rjesenje jednačine je %.2f\n", x1);
    }
    else {
        x1 = (-1*b - sqrt(pow((float)b,2)-4*a*c)) / (2*a);
        x2 = (-1*b + sqrt(pow((float)b,2)-4*a*c)) / (2*a);
        printf("Rjesenja jednačine su %.2f i %.2f\n", x1, x2);
    }
    system("pause");
    return 0;
}
```



Kako obraditi više uvjeta?



- naredbe grananja se mogu ugnježdjavati

```
if (a==0){
```

Samo onda ako je **a** jednak nuli slijedi ispitivanje vrijednosti **b**

```
  if(b<0)
```

Ako je **a** nula, a **b** je manji od nule

```
    printf("b je manji od a");
```

```
  else if (b==0)
```

Ako su i **a** i **b** nula

```
    printf("a i b su jednaki");
```

```
  else
```

Ako je **a** nula, a **b** nije niti nula niti manji od nule

```
    printf("b je veći od a");
```

```
}
```

Ovaj **else** se odnosi na **prvi if** i izvodi se onda kada je **a** različit od nule

```
else {
```

```
  printf("a nije nula!");
```

```
}
```

Ako iza naredbe grananja slijedi **samo jedna naredba**, vitičaste zagrade ne treba pisati. Inače sav programski kod treba uokviriti unutar para vitičastih zagrada.

Kako obraditi složeni uvjet?



- Primjerice kako provjeriti je li broj paran i manji od nule?
 - Primjenom binarnih logičkih operatora **|** i **||**

Logički | operator. Istinito je ako je izraz lijevo od operatora | izraz desno od operatora istiniti. U ovom primjeru, izraz je istinita i ako je broj manji od nule i ako je paran

```
if ((broj < 0) && (broj % 2 == 0)) {...}
```

- Ako je dovoljno da barem jedan od izraza bude istinit, koristi se logički || operator

```
if ((broj < 0) || (broj % 2 == 0)) {...}
```

Logički || operator. Istinito je ako je || izraz lijevo od operatora || izraz desno od operatora istiniti. U ovom primjeru, izraz je istinit i ako je broj manji od nule || ako je paran.

3. Zadatak - Finale



Hrvatska nogometna reprezentacija igra finale svjetskog nogometnog prvenstva 2018. godine protiv Brazila. Zadatak je napisati program za obradu podataka s natjecanja. Poznat je broj golova A koji je postigla Hrvatska i broj golova B koji je postigao Brazil, te prosječan broj golova u finalu svjetskih prvenstava.

ULAZNI PODACI:

- cijeli brojevi A i B, broj golova koje su dali Hrvatska i Brazil u finalu.
- decimalni broj P, prosjek golova u finalu.

IZLAZNI PODACI:

- Oznaku države koja je pobijedila (1=Hrvatska, 2=Brazil, 0=neriješeno), te broj -1, 0 ili 1 ovisno o tome je li zbroj golova u finalu manji, jednak ili veći u odnosu na prosjek golova u finalu.

PRIMJER

ULAZ: 3 2 2.75

1 1 2.25

IZLAZ: 1 1

0 -1

Vrijeme





Rješenje 3. zadatka

```

#include <stdio.h>
#include <stdlib.h>
int main(){
    int a, b, uk, pob = 1, us = 1;
    float p, x2;
    printf("Unesite rezultat u finalu A B i prosjek P: ");
    scanf("%d %d %f", &a, &b, &p);
    uk = a + b;
    if (uk < p)
        us = -1;
    else if (uk == p)
        us = 0;
    if (a == b)
        pob = 0;
    else if (a < b)
        pob = 2;
    printf("Pobjednik %d uspjeh %d\n", pob, us);
    system("pause");
    return 0;
}

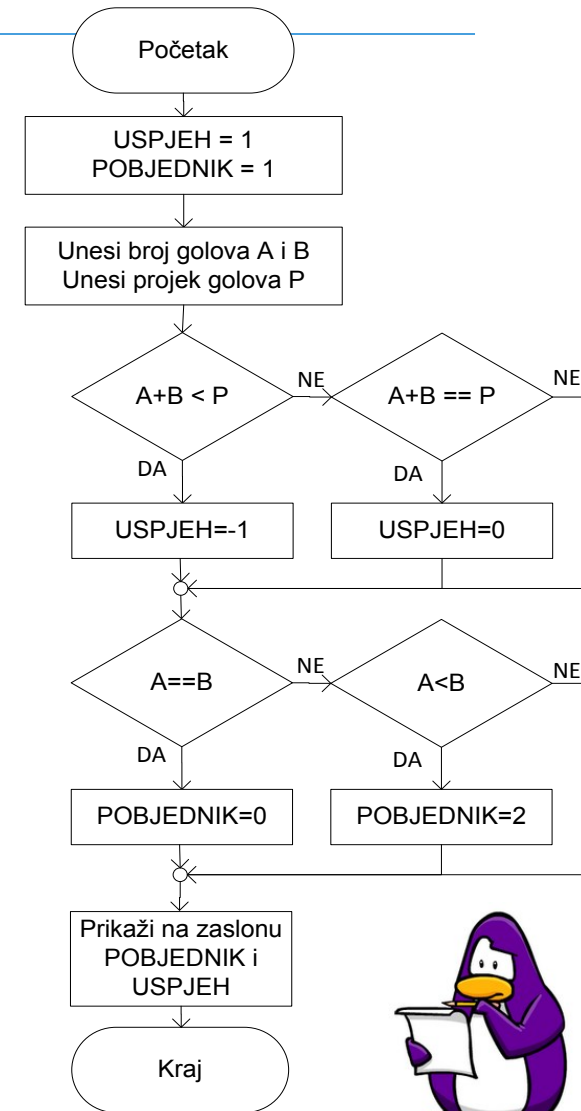
```

Algoritam



Programski kod

Rezultat izvođenja



2-tjedni izazov



Program treba na osnovu zadanog broja zaključenih petica, četvorki, trojki, dvojki i jedinica iz nastavnih predmeta odrediti i ispisati prosječnu vrijednost svih zaključenih ocjena te dodatno ispisati odgovarajuću ukupnu zaključnu ocjenu u opisnom obliku.

Ukupna zaključna ocjena “nedovoljan” zaključuje se kada je učenik imao barem jednu zaključenu jedinicu. Inače se ukupna zaključna ocjena zaključuje na sljedeći način:

“odličan” se zaključuje kada je prosjek nenegativnih dobivenih zaključnih ocjena iz svih predmeta između $4.50 \leq p \leq 5.00$, ocjena “vrlo dobar” za $3.50 \leq p < 4.50$, ocjena “dobar” za $2.50 \leq p < 3.50$ te ocjena “dovoljan” za $2.00 \leq p < 2.50$

ULAZ

- Cijeli broj O1 ($0 \leq O1$), broj zaključenih jedinica;
- Cijeli broj O2 ($0 \leq O2$), broj zaključenih dvojki;
- Cijeli broj O3 ($0 \leq O3$), broj zaključenih trojki;
- Cijeli broj O4 ($0 \leq O4$), broj zaključenih četvorki;
- Cijeli broj O5 ($0 \leq O5$), broj zaključenih petica;

IZLAZ

- U prvi redak izlaza treba ispisati prosječnu vrijednost svih zaključenih ocjena ispisanu na dvije decimale;
- U drugi redak treba ispisati jednu od sljedećih poruka: “odlican”, “vrlo dobar”, “dobar”, “dovoljan” ili “nedovoljan”.

ULAZ: 0 2 4 5 5

IZLAZ: 3.81

vrlo dobar

2 3 4 5 5

3.42

nedovoljan