



Funkcije, polja i nizovi znakova

Tomo Sjekavica

Ožujak 2012.



Zaštićeno licencom <http://creativecommons.org/licenses/by-nc-sa/3.0/hr/>

Creative Commons



- o **slobodno smijete:**

- n **dijeliti** — umnožavati, distribuirati i javnosti priopćavati djelo

- n **remiksirati** — prerađivati djelo

- o **pod slijedećim uvjetima:**

- n **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).

- n **nekomercijalno.** Ovo djelo ne smijete koristiti u komercijalne svrhe.

- n **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/>.



Funkcija `main`

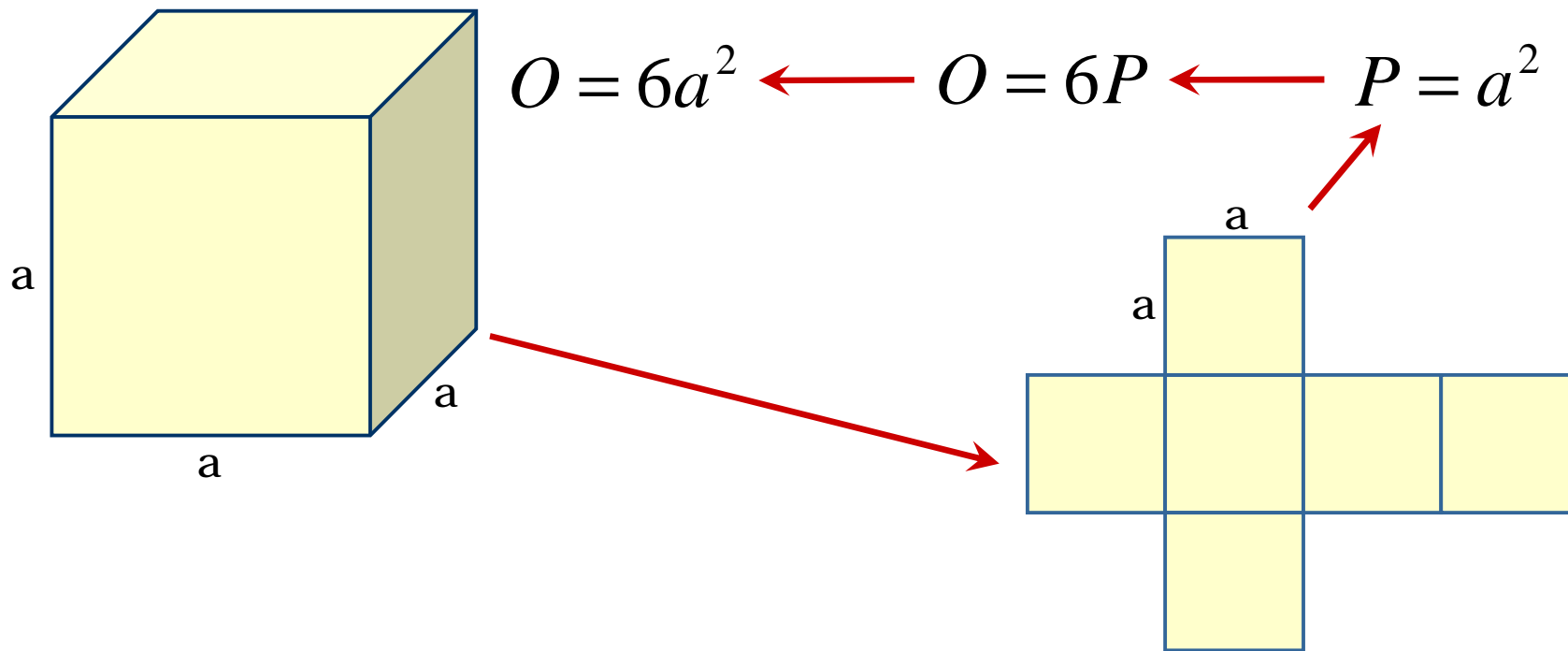
```
int main() { ... }
```

- `main` funkcija je početna ili glavna funkcija svakog C programa
- svaki C program mora imati jednu `main` funkciju unutar koje se piše izvorni kod
- jedan C program ne može imati više `main` funkcija
- sve naredbe unutar funkcije se pišu unutar vitičastih zagrada { ... }



Oplošje kocke

- kocka se sastoji od 6 jednakih kvadrata
- svaki kvadrat se sastoji od 4 jednake stranice
- kako ćemo izračunati oplošje kocke?





Oplošje kocke v0.1

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

```
int main() {  
    int a, o1, o2;
```

```
    printf("Unesite duljinu stranice kocke: ");  
    scanf("%d", &a);
```

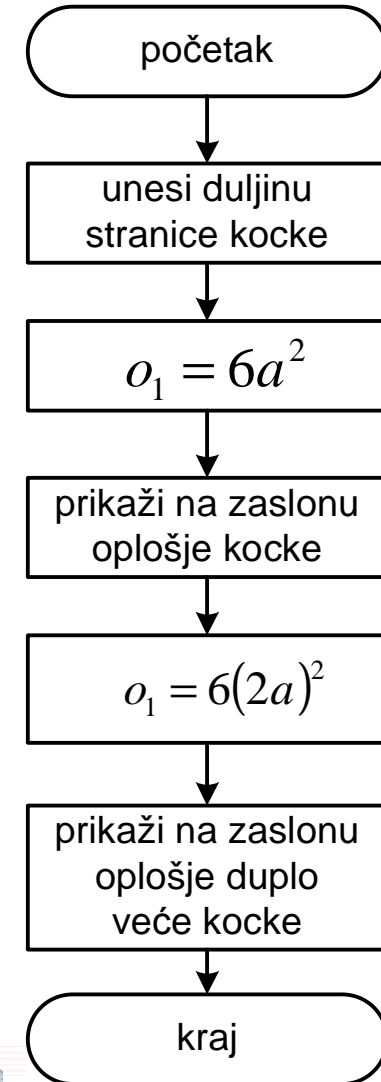
```
    o1 = 6 * a * a;  
    printf("Oplosje kocke je: %d. \n", a, o1);
```

```
    o2 = 6 * 2*a * 2*a;  
    printf("Oplosje duplo vece kocke je: %d. \n", o2);
```

```
    system("pause");  
    return 0;
```

```
}
```

```
C:\D:\Tomo\Fakultet_Tomo\Radionice_progra...  
Unesite duljinu stranice kocke: 5  
Oplosje kocke je: 150.  
Oplosje duplo vece kocke je: 600.  
Press any key to continue . . .
```



oplosje01.c



Anatomija C programa

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

```
int main() {
```

```
    int a, o1, o2;
```

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

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

```
    o1 = 6 * a * a;
```

```
    printf("Oplosje kocke je: %d. \n", a, o1);
```

```
    o2 = 6 * 2*a * 2*a;
```

```
    printf("Oplosje duplo vece kocke je: %d. \n", o2);
```

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

```
    return 0;
```

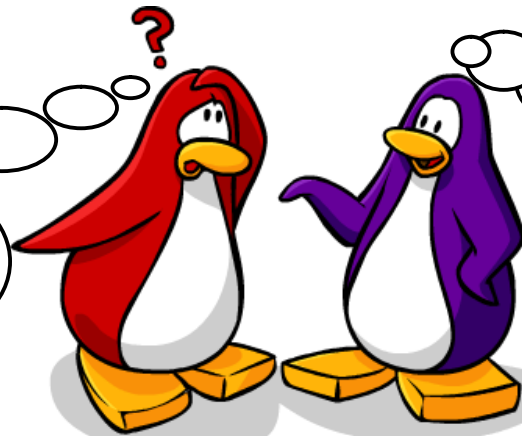
```
}
```

Rezervacija memorijskog prostora za pohranu 3 cijela broja.

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

Isti dio koda se ponavlja dva puta u programu.

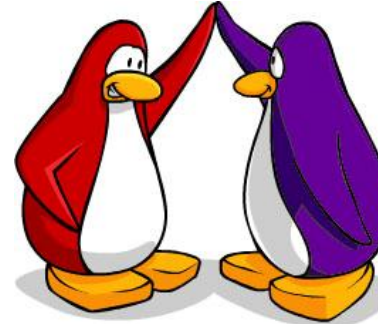
Ako trebam dio koda koristiti više puta u istom programu, da li ga kopirati na svako mjesto? Je li to dobra ideja?



Ne! Ako nešto promijeniš, onda to moraš promijeniti na svakom mjestu. Zar nisi čuo za funkcije? Možeš taj kod spremi u funkciju i onda samo pozvati funkciju kad ti treba taj dio koda!



Oplošje kocke v0.2 (s funkcijom)

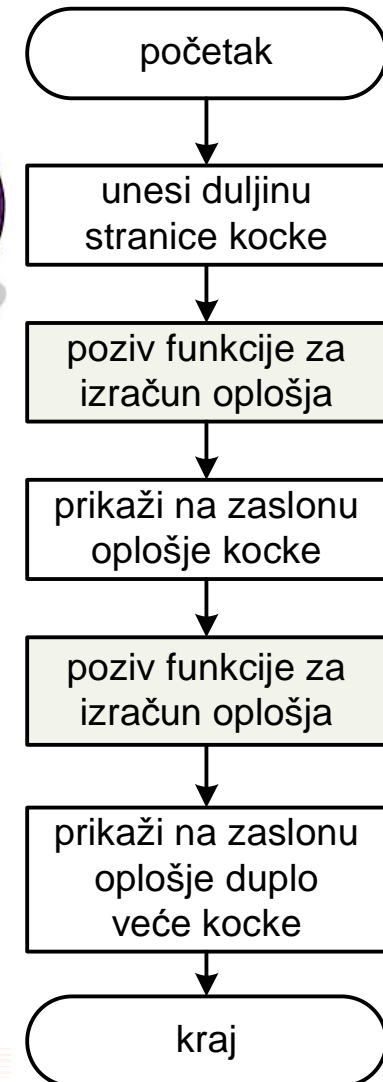


```
#include<stdio.h>
```

```
int oplosje(int str){  
    int op;  
    op = 6 * str * str;  
    return op;  
}
```

```
int main() {  
    int a, o1, o2;  
    printf("Unesite duljinu stranice kocke: ");  
    scanf("%d", &a);  
    o1 = oplosje(a);  
    printf("Oplosje kocke je: %d.\n", o1);  
    o2 = oplosje(2*a);  
    printf("Oplosje duplo vece kocke je: %d.\n", o2);  
    system("pause");  
    return 0;  
}
```

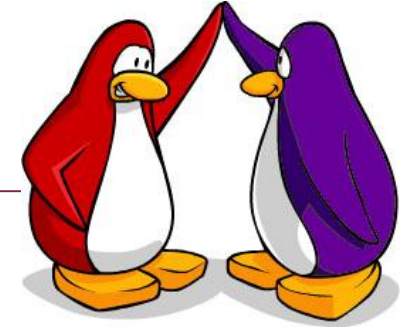
```
C:\D:\Tomo\Fakultet_Tomo\Radionice_progra...  
Unesite duljinu stranice kocke: 4  
Oplosje kocke je: 96.  
Oplosje duplo vece kocke je: 384.  
Press any key to continue . . .
```



oplosje02.c



Anatomija C programa



Kao i kod main funkcije
tijelo funkcije je omeđeno
vitičastim zagradama.

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

```
int oplosje(int str) {  
    int op;  
    op = 6 * str * str;  
    return op;  
}
```

Povratna vrijednost funkcije je cijeli broj.

Deklaracija i definicija funkcije
oplosje za izračun oplošja kocke.
Funkcija u C-u mora biti
deklarirana prije funkcije main.

Funkcija kao parametar prima jedan cijeli broj.

Funkcija oplosje glavnom programu vraća
izračunato oplošje kocke kao povratnu vrijednost.

```
int main() {  
    int a, o1, o2;  
    printf("Unesite duljinu stranice kocke: ");  
    scanf("%d", &a);  
    o1 = oplosje(a);  
    printf("Oplosje kocke je: %d. \n", o1);  
    o2 = oplosje(2*a);  
    printf("Oplosje duplo veće kocke je: %d. \n", o2);  
    system("pause");  
    return 0;  
}
```

Poziv funkcije oplosje. Povratna
vrijednost funkcije se sprema u
varijablu o1, odnosno u varijablu o2.



Funkcije - zaključak

- Funkcije u C-u mogu primiti **0, 1 ili više parametara**:
 - n kod predavanja parametara kod deklaracije funkcije obavezno treba navesti **tip** i **naziv** svakog parametra:

```
int oplosje(int stranica) { ... }
```



Funkcije - zaključak

- Funkcije u C-u mogu vraćati glavnom programu **povratnu vrijednost**:
 - n ispred naziva funkcije se navodi **tip** povratne vrijednosti funkcije:

```
int oplosje(int stranica) { ... }
```

- n ako funkcija nema povratnu vrijednost, onda će **tip** povratne vrijednosti funkcije biti **void**:

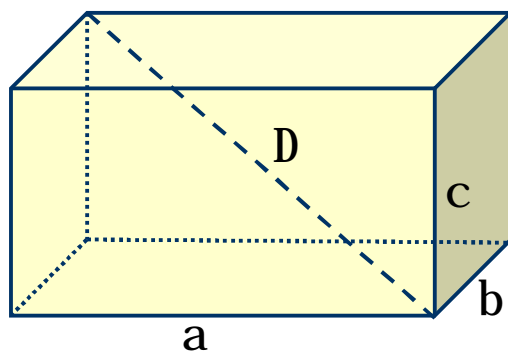
```
void funkcija(int broj) { ... }
```



Oplošje i dijagonala kvadra



- **Izmijenite** program za izračun oplošja kocke, tako da ima **dvije funkcije**. Prva funkcija računa **oplošje kvadra**, a druga funkcija računa **dijagonalu kvadra**.



$$O = 2(ab + bc + ca)$$

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

Hmm, koji će mi biti tip povratne vrijednosti funkcije za izračun dijagonale kvadra?

Dijagonalu prikazati sa 3 decimale.

```

c:\ D:\Tomo\Fakultet_Tomo\Radionice_programiran...
Unesite sve 3 stranice kvadra: 10 5 3
Oplosje kvadra je: 190.
Dijagonala kvadra je: 11.576.
Press any key to continue . . . .
  
```



Polja



○ kako kreirati niz od 10 cijelih brojeva?

○ 1. pokušaj:

```
int broj 1, broj 2, ..., broj 10;
```

○ 2. pokušaj:

```
int brojevi [10];
```

Ako kod deklaracije polja odmah želimo pridružiti vrijednosti treba ih unijeti unutar { }, npr.:
`int brojevi[10] = {0}`

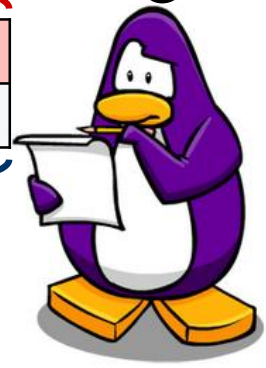
To je jednostavno, pa samo kreiramo 10 cjelobrojnih varijabli. Ali što ako nam bude trebalo npr. 100 cjelobrojnih varijabli? Pa nećemo valjda morati svaku posebno kreirati!

Za pohranu niza podataka istoga tipa, bolje je koristiti polja. Pored naziva polja u uglatim zagradama samo navedete koliko elemenata ima polje!

Brojevi (elementi polja).

0	1	2	3	4	5	6	7	8	9

Indeksi elemenata polja.
Indeksi idu od 0 do n-1
(u ovom primjeru do 10-1 = 9).





Unos i pristup elementima polja

Deklaracija polja koje može sačuvati 5 cjelobrojnih vrijednosti.

unos brojeva sa tipkovnice i spremanje u polje pomoću for petlje

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

```
int main() {
```

```
int polje[5], i;
```

```
for(i=0; i<5; i=i+1) {  
    scanf("%d", &polje[i]);
```

```
}
```

```
printf("Ispis elemenata polja: ");
```

```
for(i=0; i<5; i=i+1) {  
    printf("%d ", polje[i]);
```

```
}
```

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

```
return 0;
```

```
}
```

i = 0
polje[0] = 5

i = 1
polje[1] = 3

i = 2
polje[2] = 9

i = 3
polje[3] = 4

i = 4
polje[4] = 2

i = 5
kraj for petlje

pristup elementima polja pomoću for petlje i ispis na ekran.

5	3	9	4	2
0	1	2	3	4

```
C:\D:\Tomo\Fakultet_Tomo\Radionice_progr... - _ _ X
Unesite elemente polja:
Unesite [0]. element polja: 5
Unesite [1]. element polja: 3
Unesite [2]. element polja: 9
Unesite [3]. element polja: 4
Unesite [4]. element polja: 2
Ispis elemenata polja: 5 3 9 4 2
Press any key to continue . . .
```



polje.c

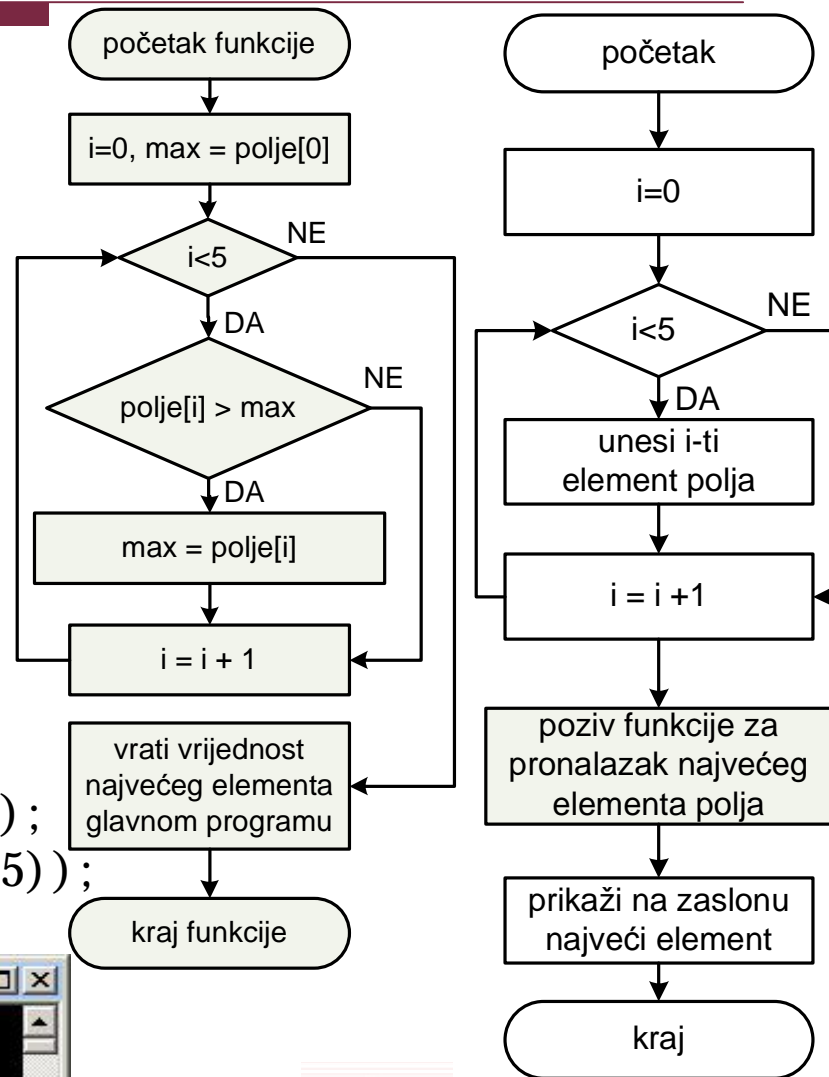
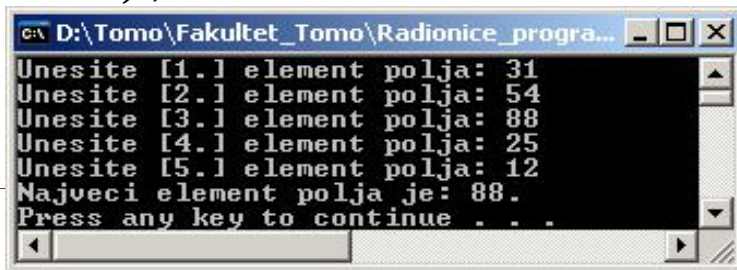


Najveći element polja pomoću funkcije

```

#include<stdio.h>
int najveci (int polje[], int brel) {
    int i, max = polje[0];
    for(i=0; i<brel; i=i+1)
        if(polje[i] > max)
            max = polje[i];
    return max;
}
int main() {
    int polje[5], i;
    for(i=0; i<5; i=i+1){
        scanf("%d", &polje[i]);
    }
    printf("Najveci element polja je: ");
    printf("%d. \n", najveci (&polje[0], 5));
    system("pause");
    return 0;
}

```



polje_max.c



Anatomija C programa



funkcija najveći

```
#include <stdio.h>
int najveći (int polje[], int brel) {
    int i, max = polje[0];
    for (i=0; i<brel; i=i+1)
        if (polje[i] > max)
            max = polje[i];
    return max;
}
```

Varijabli max moramo postaviti početnu vrijednost prije prvog korištenja. Kao najveći element ćemo postaviti vrijednost 1. elementa polja.

Provjera da li je trenutni element veći od najvećeg.

Ako je veći onda vrijednost tog elementa postaje najveća.

Funkcija vraća vrijednost najvećeg elementa polja.

Deklaracija polja koje može sačuvati 5 cjelobrojnih vrijednosti.

```
int main() {
    int polje[5];
    int i;
    for (i=0; i<5; i=i+1) {
        scanf ("%d", &polje[i]);
    }
    printf ("Najveći element polja je: ");
    printf ("%d. \n", najveći (&polje[0], 5));
    system ("pause");
    return 0;
}
```

for petlja omogućuje da se instrukcije izvrše više puta.

Provjerava se da li je i<5

Ako uvjet nije zadovoljen, petlja se nastavlja izvoditi. Izvršavaju se instrukcije unutar bloka omeđenog vitičastim zagradama.

Kad se izvrše, vrijednost varijable i se povećava za 1.

Funkciji se predaje adresa prvog elementa polja. Već smo rekli da je indeks prvog elementa 0, pa radi toga predajemo &polje[0].

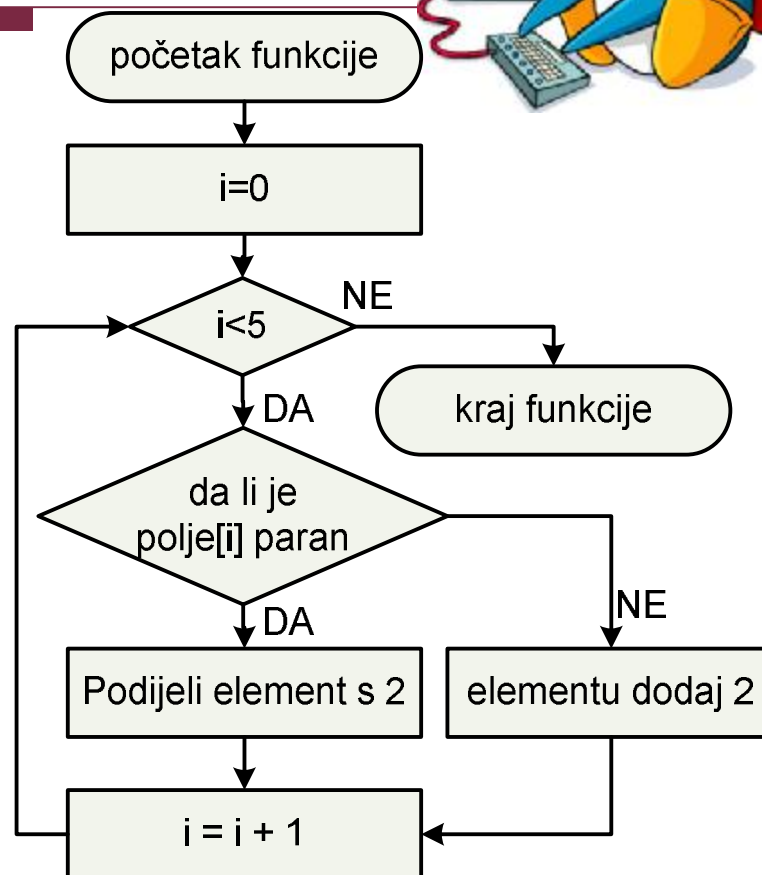
Poziv funkcije najveći. Funkcija prima 2 parametra:



Parni i neparni elementi polja



- Napisati funkciju koja mijenja vrijednosti elementima zadanog polja na slijedeći način:
 - n** parne brojeve dijeli s 2
 - n** neparnim brojevima dodaje broj 2
- Funkcija nema povratnu vrijednost.
- U glavnom programu se unose vrijednosti elemenata polja, poziva funkcija i ispisuje polje s novim vrijednostima.



```
c:\D:\Tomo\Fakultet_Tomo\Radionice_progra...
Unesite [1.] element polja: 15
Unesite [2.] element polja: 22
Unesite [3.] element polja: 34
Unesite [4.] element polja: 19
Unesite [5.] element polja: 35
17 11 17 21 37
Press any key to continue . . . . -
```


Nizovi znakova

○ kako spremiti niz znakova u C-u?

```
char niz[] = "Radionica";
```

Sigurno postoji neki tip podatka **string** ili nešto slično u koji se mogu spremiti nizovi znakova kao varijable!

U C-u ne postoji tip podatka **string**, nizovi znakova u C-u se spremaju kao polja znakova (**char** polja).

Niz znakova

R	a	d	i	o	n	i	c	a	\0
0	1	2	3	4	5	6	7	8	9



10

Indeksi elemenata niza znakova.
Indeksi idu od 0 do n-1
(u ovom primjeru do 10-1 = 9).

Program koji čita niz znakova zna da je došao do kraja niza kad pročita znak '\0'. Taj znak ima vrijednost 0.
Svaki niz znakova na kraju ima znak '\0', a taj znak se automatski dodaje kod spremanja niza znakova.

Odlično, ali kad čitamo niz znakova kako ćemo znati kada smo došli do kraja niza?



Unos i ispis imena učenika

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

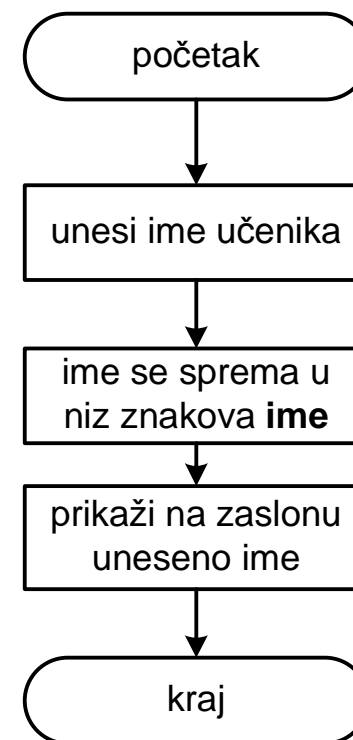
```
int main() {  
    char ime[20+1];
```

```
    printf("Unesite ime ucenika: ");  
    scanf("%s", ime);
```

```
    printf("Uneseno ime je: %s\n", ime);
```

```
    system("pause");  
    return 0;
```

```
}
```



```
D:\Tomo\Fakultet_Tomo\Radionice_progr...  
Unesite ime ucenika/ce: Pero  
Uneseno ime je: Pero  
Press any key to continue . . .
```



ime.c

18

Anatomija C programa



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

Deklaracija polja `ime` kao niz znakova u koji se može spremiti 20 znakova. Onaj +1 je rezerviran za '\0'.

```
int main() {  
    char ime[20+1];  
    printf("Unesite ime ucenika: ");  
    scanf("%20s", ime);
```

Unos teksta sa tipkovnica i spremanje u niz znakova `ime`. `%20s` znači da će se u niz spremiti samo prvih 20 unesenih znakova.

Hmm, pa naučili smo da funkcija `scanf` prima adresu varijable. Zašto u ovom programu nema znaka `&` ispred varijable `ime`? Hoće li program ovako raditi ispravno?

```
    printf("Uneseno ime je: %s\n", ime);  
    system("pause");  
    return 0;
```

`%s` se koristiti za ispis niza znakova.



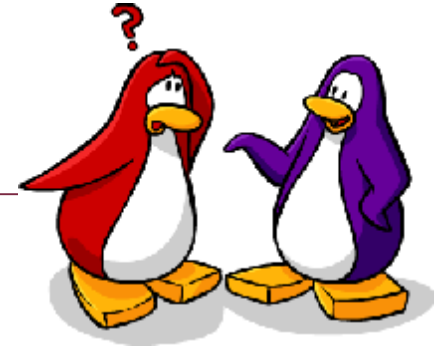
Program će raditi ispravno. Varijabla `ime` predstavlja adresu u memoriji gdje počinje niz znakova. To je isto kao da smo napisali `&ime[0]`.

`&ime[0]` `ime`

T	o	m	o	\0
0	1	2	3	4



Unos imena i prezimena v0.1



- Izmjenili smo program da može spremiti 50 znakova u niz znakova što bi trebalo biti dovoljno za spremanje imena i prezimena.

To je zato što **%s** u niz znakova sprema sve znakove do prve praznine (tipka **Space**).

```
C:\D:\Tomo\Fakultet_Tomo\Radionice_programiranja\I... - _ _ X
Unesite ime i prezime ucenika/ce: Pero Peric
Uneseno ime i prezime je: Pero.
Press any key to continue . . . . _
```

Zašto mi se u niz znakova **imeprez** sprema samo ime, a ne i prezime?

Najčešće korišteni formati unosa kod scanf funkcije:

samo znamenke	<code>%[0123456789]</code> ili <code>%[0-9]</code>
samo samoglasnici	<code>%[aeiouAEIOU]</code>
samo mala slova	<code>%[abcdefghijklmnopqrstuvwxyz]</code> ili <code>%[a-z]</code>
samo velika slova	<code>%[ABCDEFGHIJKLMNOPQRSTUVWXYZ]</code> ili <code>%[A-Z]</code>
slova i praznina	<code>%[a-zA-Z]</code>
sve osim znamenki	<code>%[^0123456789]</code> ili <code>%[^1-9]</code>
sve osim Enter	<code>%[^\n]</code>



imeprez1.c



Unos imena i prezimena v0.2

- U funkciji `scanf` smo umjesto `%50s` stavili `%50[^\n]` i sada možemo unositi sve znakove uključujući i prazninu sve dok ne unesemo “**Enter**”.

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

```
int main() {
```

```
    char imeprez[50+1];
```

```
    printf("Unesite ime i prezime ucenika: ");
```

```
    scanf("%50[^\n]", imeprez);
```

```
    printf("Uneseno ime i prezime je: ");
```

```
    printf("%s. \n", imeprez);
```

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

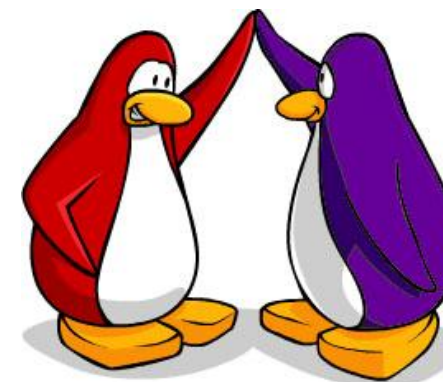
```
    return 0;
```

```
}
```

2011/2012

Funkcije, polja i nizovi znakova

```
C:\D:\Tomo\Fakultet_Tomo\Radionice_programiranja\I... _ _ X
Unesite ime i prezime ucenika/ce: Pero Peric
Uneseno ime i prezime je: Pero Peric.
Press any key to continue . . . _
```



imeprez2.c

21



Unos imena i prezimena v0.3

- Nakon unosa imena i prezimena, treba još unijeti i spol.

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

```
int main() {
```

```
    char imeprez[50+1];
```

```
    char spol;
```

```
    printf("Unesite ime i prezime ucenika/ce: ");
```

```
    scanf("%50[^\n]", imeprez);
```

```
    printf("Unesite spol ucenika/ce: ");
```

```
    scanf("%c", &spol);
```

```
    printf("Uneseno ime i prezime je: %s.\n", imeprez);
```

```
    printf("Spol ucenika/ce je: %c.\n", spol);
```

```
    system("pa
```

```
    return 0;
```

```
}
2011/2012
```

Zašto ne mogu unijeti spol nakon unosa imena i prezimena?

Unio si niz znakova i na kraju **Enter**. Računalo taj **Enter** shvaća kao jedan uneseni znak pa ga učitava u slijedećoj **scanf** funkciji!

Unos jednog znaka



```

C:\D:\Tomo\Fakultet_Tomo\Radionice_programiranja\II_radionica\zadaci\im...
Unesite ime i prezime ucenika/ce: Pero Peric
Unesite spol ucenika/ce: Uneseno ime i prezime je: Pero Peric.
Spol ucenika/ce je:
Press any key to continue . . .

```



meprez3.c

Unos imena i prezimena v0.4

- Kako spriječiti spremanje '\n' u slijedećoj scanf funkciji

```
printf("Unesite ime i prezime ucenika/ce: ");  
scanf("%50[^\n]", imeprez);  
printf("Unesite spol ucenika/ce: ");  
scanf(" %c", &spol); ← postavljanje praznine prije %c  
printf("Uneseno ime i prezime je: %s.\n", imeprez);  
printf("Spol ucenika/ce je: %c.\n", spol);
```



```
D:\Tomo\Fakultet_Tomo\Radionice_programiranja\I...  
Unesite ime i prezime ucenika/ce: Pero Peric  
Unesite spol ucenika/ce: M  
Uneseno ime i prezime je: Pero Peric.  
Spol ucenika/ce je: M.  
Press any key to continue . . . _
```

Ako ima više scanf funkcija u jednom programu poželjno je ostaviti **prazninu** prije navođenja prve oznake formata unosa da se spriječi učitavanje znaka **Enter**.



imeprez4.c

ASCII tablica

○ Kako računalo razumije koji smo znak unijeli?

Decimal	Char	Decimal	Char	Decimal	Char	Decimal	Char
0	[NULL]	32	[SPACE]	64	@	96	`
1	[START OF HEADING]	33	!	65	A	97	a
2	[START OF TEXT]	34	"	66	B	98	b
3	[END OF TEXT]	35	#	67	C	99	c
4	[END OF TRANSMISSION]	36	\$	68	D	100	d
5	[ENQUIRY]	37	%	69	E	101	e
6	[ACKNOWLEDGE]	38	&	70	F	102	f
7	[BELL]	39	'	71	G	103	g
8	[BACKSPACE]	40	(72	H	104	h
9	[HORIZONTAL TAB]	41)	73	I	105	i
10	[LINE FEED]	42	*	74	J	106	j
11	[VERTICAL TAB]	43	+	75	K	107	k
12	[FORM FEED]	44	,	76	L	108	l
13	[CARRIAGE RETURN]	45	-	77	M	109	m
14	[SHIFT OUT]	46	.	78	N	110	n
15	[SHIFT IN]	47	/	79	O	111	o
16	[DATA LINK ESCAPE]	48	0	80	P	112	p
17	[DEVICE CONTROL 1]	49	1	81	Q	113	q
18	[DEVICE CONTROL 2]	50	2	82	R	114	r
19	[DEVICE CONTROL 3]	51	3	83	S	115	s
20	[DEVICE CONTROL 4]	52	4	84	T	116	t
21	[NEGATIVE ACKNOWLEDGE]	53	5	85	U	117	u
22	[SYNCHRONOUS IDLE]	54	6	86	V	118	v
23	[ENG OF TRANS. BLOCK]	55	7	87	W	119	w
24	[CANCEL]	56	8	88	X	120	x
25	[END OF MEDIUM]	57	9	89	Y	121	y
26	[SUBSTITUTE]	58	:	90	Z	122	z
27	[ESCAPE]	59	;	91	[123	{
28	[FILE SEPARATOR]	60	<	92	\	124	
29	[GROUP SEPARATOR]	61	=	93]	125	}
30	[RECORD SEPARATOR]	62	>	94	^	126	~
31	[UNIT SEPARATOR]	63	?	95	_	127	[DEL]

Svaki znak ima svoju numeričku vrijednost koja je zadana pomoću ASCII tablice.



○ Kako ćemo saznati je li neki znak znamenka?

```
if(niz[i] >= '0' && niz[i] <= '9')
```

ili

```
if(niz[i] >= 48 && niz[i] <= 57)
```



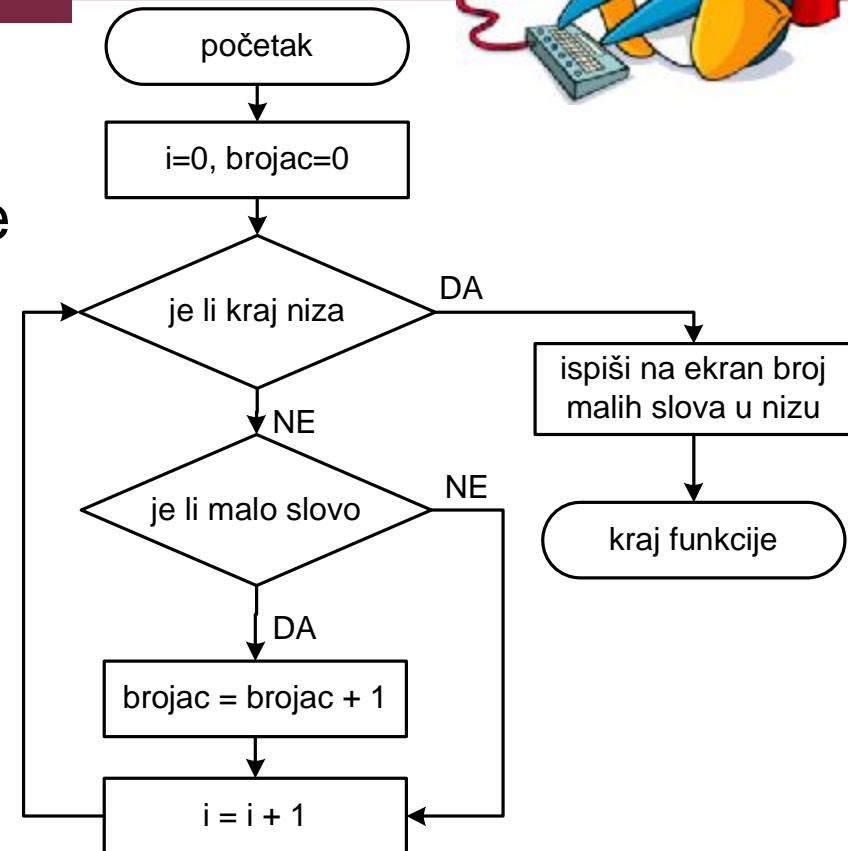

Broj malih slova u nizu



- Napisati program u kojem se može unijeti niz znakova koji može sadržavati najviše 100 znakova uključujući praznine.
- Izbrojati koliko uneseni niz sadrži malih slova.

Dobro je znati, možda će vam trebati:

Jednostruki i dvostruki navodnici nisu isto, pa se koriste za različite stvari.
jednostruki navodnici označavaju samo jedan znak, npr: 'R'
(podatak tipa **char**).
dvostruki navodnici označavaju niz znakova, npr: "Radi onice programiranja"
(polje znakova).



```
C:\D:\Tomo\Fakultet_Tomo\Radionice_programiranja\II_radi...
Unesite niz znakova:UNIDU radionice programiranja
Broj malih slova u unesenom nizu je: 22.
Press any key to continue . . .
```