INFORMATIČKI KLUB

FUTURA

```
#include<stdio.h>
int main()
{
    printf("Hello World!");
    return 0;
}
```

C

RADIONICE PROGRAMIRANJA

ZA SREDNJOŠKOLCE I STUDENTE

RADIONICE PROGRAMIRANJA ZA SREDNJE ŠKOLE

3. RADIONICA - POČETNICI

Informatički klub FUTURA

Dubrovnik, 2016.

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/

Rad s nizovima znakova(eng. string)

- U C-u se niz znakova bilježi kao polje podataka tipa **char**.
  - char se čita kao "***kar***" (od "*karakter*"), a NE "***čar***" (nije "čaračter" ☺)
- Polje završava posebnim znakom "završetka niza" - **\0**
- Primjerice ako je niz znakova „**Futura**”
  - varijabla je **polje duljine 7 znakova** (6 za tekst i 1 za oznaku kraja niza) **char niz[6+1];** odnosno **char niz[7];**
- Pristup jednom po jednom znaku niza obično se radi **while** petljom

char niz[7];

niz[0]	niz[1]	niz[2]	niz[3]	niz[4]	niz[5]	niz[6]
F	u	t	u	r	a	\0

```
while (niz[i] != '\0') {
    printf("%c", niz[i]);
    if (niz[i] == 'x')
        break;
    i++;
}
```

While petlja se izvodi **sve dok je uvjet u zagradama istinit**. Ako na početku taj uvjet nije istinit, petlja se neće izvesti niti jednom.

Iz petlje se može izaći u bilo kojem trenutku pozivom naredbe **break**.

Rad s nizovima znakova

- Unos niza kod deklaracije

```
char niz[] = “Radionica”;
```

- Unos niza s prazninama sa standardnog ulaza (sve do „entera”)

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

- Kopiranje **niza1** (stari niz) u **niz2** (novi niz)

```
#include<string.h>
strcpy (niz2, niz1);
```

- Lijepljenje **niza1** u **niz2** (novi niz)

```
#include<string.h>
strcat (niz2, niz1);
```

- Uspoređivanje dva niza **niza1 i niz2**

```
#include<string.h>
int strcmp (niz1, niz2);
```

  - Ako su oba niza jednaka, funkcija vraća 0
  - Ako je prvi niz manji od drugoga, funkcija vraća broj < 0
  - Ako je prvi niz veći od drugoga, funkcija vraća broj > 0

Zadatak 5 – niz znakova

- Unutar ASCII kodne tablice, kodovi malih i velikih slova engleske abecede se razlikuju za 32. Npr. ‘A’ ima kod 65, ‘a’ ima kod 97, ‘B’ ima kod ‘66’, ‘b’ ima kod ‘98’,...

- Zadatak:
- Unijeti niz znakova s tipkovnice maksimalne duljine 100 znakova.
- Prebacite sva velika slova u mala i ispišite niz.

- Primjer:
- Unesite niz: Danas je lijep dan.
- Rezultat: danas je lijep dan.

Zadatak 6 – uspoređivanje nizova

- Usporedite dva niza znakova. Prvi postavite prilikom deklariranja varijable, a drugi unesite preko tipkovnice.

Funkcije – radni zadatak

- Unijeti tri broja i ispisati zbroj
  - Ispišite poruku na ekran: ”Unesite broj”
  - Koristeći scanf unijeti prvi broj
  - Ponovno ispišite poruku na ekran: ”Unesite broj”
  - Koristeći scanf unijeti drugi broj
  - Ponovno ispišite poruku na ekran: ”Unesite broj”
  - Koristeći scanf unijeti treći broj
  - Ispisati zbroj tri broja

Funkcije

- Što se može primijetiti u prethodnom primjeru?
- Omogućuju ponovnu iskoristivost programskog koda.
- Zasebni odsječci programa koji imaju svoju funkcionalnost.
- Olakšavaju preglednost programskog koda.
- Primjeri funkcija koje smo koristili:
  - printf
  - scanf
  - strcpy

Funkcije – definicija funkcije

Tip funkcije – određuje kojeg je tipa povratna vrijednost funkcije.

Ime funkcije – identifikator kojeg koristimo kod pozivanja funkcije.

Lista argumenata funkcije – varijable koje sadrže vrijednosti koje se predaju funkciji kod njenog pozivanja. **Formalni argumenti**.

Sintaksa zapisa argumenata je: **tip naziv**

Više argumenata u funkciji odvaja se zarezom:

int funkcija(float arg1, char arg2, float arg3);

```
double potenciraj(int broj, int potencija) {
  int i;
  double umnozak=1;

  for(i=1; i<=potencija; i++) {
      umnozak = umnozak * broj;
    }

  return umnozak;
}
```

Tijelo funkcije

Deklaracija lokalnih varijabli. Lokalne varijable vidljive su samo unutar tijela funkcije u kojoj su deklarirane.

return – naredba za povratak vrijednosti funkcije i prekid njenog izvršavanja.

Definicija funkcije predstavlja cijeli programski kod kojim je opisana njezina funkcionalnost.

Funkcije

Void – funkcija tipa **void** ne vraća vrijednost.

Lista argumenata funkcije – funkcija ne mora primati argumente ako nije potrebno.

```
void pozdrav() {

  printf("Dobar dan.\n");

}
```

Tijelo funkcije

return – primjećujemo da funkcija ne vraća vrijednost. Nema naredbe **return**.

Funkcije – poziv funkcije

```
int zbroji(int x, int y) {
  int suma = 0;
  suma = x + y;
  return suma;
}

int main() {
  int a, b, c;
  printf("Unesite dva broja [a b]:
");
  scanf("%d %d", &a, &b);

  c = zbroji(a, b);

  printf("%d + %d = %d\n", a, b, c);
  return 0;
}
```

Poziv funkcije – navodimo ime funkcije i navodimo listu stvarnih argumenata.

Funkcije – radni zadatak

- Koristeći znanje o funkcijama izmijeniti radni zadatak tako da izbjegnemo dupliciranje koda
  - Izdvojiti ispis poruke i unos broja u funkciju
  - Funkcija treba vraćati unesenu vrijednost
  - U main funkciji pozvati pomoćnu funkciju
  - Konačno zbrojiti i ispisati rezultat

Funkcije – main funkcija

Osnovna funkcija s kojom počinje izvršavanje programa.

Tip funkcije – uvijek mora biti tipa **int** jer vraća cjelobrojnu vrijednost operacijskom sustavu.

Ime funkcije – main() označava glavnu funkciju koja se pokreće pozivanjem programa. Može biti samo jedna **main()** funkcija.

```
int main() {
   ...
   return 0;
}
```

```
void main() {
   ...
}
```

return – povratna vrijednost se prosljeđuje operacijskom sustavu prilikom završetka rada programa. Može označavati uredan kraj programa, a može predstavljati kôd greške.

Funkcije – Zadatak 2 (prvi dio)

- Napisati dvije funkcije
- 1. char pretvoriumalo(char znak);
- 2. char pretvoriuveliko(char znak);
- Funkcije trebaju malo slovo pretvoriti u veliko odnosno veliko u malo

Funkcije – Zadatak 2 (drugi dio)

- Napisati treću funkciju koja provjera slovo
- int provjeri(char znak);
- Funkcija treba vratiti:
  - 0 ako je znak malo slovo
  - 1 ako je znak veliko slovo
  - 2 ako znak nije slovo
- U main funkciji provjeriti jeli uneseno slovo
  - Ako jest pozvati priklanu pretvori funkciju
  - Ako nije ispisati poruku ”Uneseni znak nije slovo”

Funkcije – Zadatak 3

Napišite program za računanje zbroja brojeva u zadanom intervalu.

Zbrajanje brojeva u zadanom intervalu napravite u obliku funkcije, a unos brojeva i ispis rezultata napravite u glavnom programu.

```
INTERVAL: [5 8]
ALGORITAM: 5+6+7+8 = 26

PRAVILO: A <= B
```

ULAZ:	IZLAZ:
2 8	35

ULAZ:	IZLAZ:
8 8	8

ULAZ:	IZLAZ:
9 3	Greška!