11. Serial monitor i Stringovi

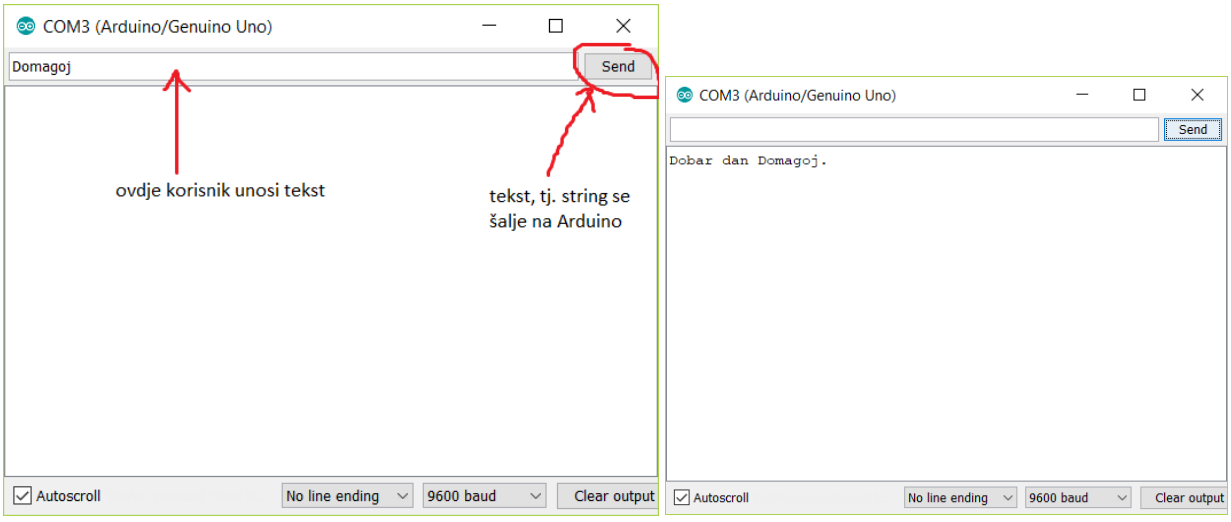
ZADATAK 1. Prepiši sljedeći kod i testiraj ga. Pronađi na internetu tablicu ASCII kodova te usporedi. Umjesto argumenta DEC, napiši HEX i promatraj što se dobiva.

```
void setup() {
   Serial.begin(9600);
}
void loop() {
   if (Serial.available() > 0) {
      int incomingByte = Serial.read();
      Serial.print("I received: ");
      Serial.println(incomingByte, DEC);
   }
}
```

HINT: ***Serial Monitor*** možeš brzo pokrenuti sa Ctrl+Shift+M

ZADATAK 2. U prethodnom zadatku zamijeni `Serial.read()` sa `String ulazni=Serial.readString();`.
Na taj način moguće je odjednom pročitati cijeli niz znakova. Testiraj program sada.

ZADATAK 3. Definiraj varijablu String ime = Serial.readString();
Ona će sadržavati korisnikovo ime koje se unosi preko *Serial Monitora,* a na kojem se zatim ispisuje poruka „Dobar dan [ime].“
Nakon ispisa poruke, Arduino čeka na novi unos, te se ništa dodatno ne ispisuje.



ZADATAK 4. Modificiraj prethodni program tako da Arduino pita korisnika koliko je sati (u 24-h formatu HH:MM, npr. 17:10). Prema tome, prilagodi prethodnu poruku (npr. Ako korisnik unese 20:05, treba ispisati „Dobra večer [ime]“).
Jutro je do 10:00, dan je od 10:00 do 19:00, a večer je nakon 19:00 sati.
Iz pročitanog stringa potrebno je izvući podatak o satima. Kako bi si olakšao, googlaj „String() arduino“ te nađi popis funkcije nad stringovima u Arduinu.

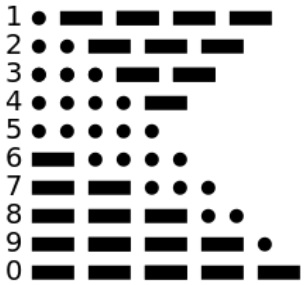
Korisne funkcije bi ti mogle biti: `substring()`, `toInt()`, `indexOf()` itd.

ZADATAK 5. Spoji 2 LED-ice različite boje na Arduino, crvenu i plavu. Preko *Serial Monitor-a* javi korisniku da upiše koju LED-icu želi upaliti/ugasiti. Dakle, korisnik upisuje „**naredba boja“** (pr. „ugasi crvena“, „upali plava“). Naredbe mogu biti samo „upali“ i „ugasi“, a boja može biti samo „crvena“ i „plava“. Prema takvom korisnikovom unosu mijenjaj stanje LED-ica.

ZADATAK 6. Modificiraj prethodni program tako da ne ovisi o malim/velikim slovima, tj. ako korisnik napiše „Ugasi Crvena“ ili „UGASI crvena“, rezultat je jednak kao da je napisao „ugasi crvena“.
Korisne bi mogle biti funkcije `toUpperCase()`, `toLowerCase()` i `compareTo()`.

ZADATAK 7. Proširi prethodni kod novom naredbom „*blinkaj boja PERIOD*“ (period je vrijeme u ms između dva blinkanja, npr. za „blinkaj crvena 500“, crvena ledica svijetli 250ms, zatim ne svijetli 250ms, pa onda opet svijetli 250ms itd.)
Isprobaj novu naredbu, te zatim probaj unijeti neku narebu iz prethodnog zadatka.
Što se dogodilo? Možeš li to objasniti?

ZADATAK 8. Morseov kod je izmislio Samuel Morse za šifriranje i prenošenje slova, brojeva i znakova koristeći odgovarajući niz udaraca, tonova ili svjetlosnih treptaja različitih dužina trajanja. Morseovom kodom se koriste spasilačke službe, radioamateri i vojnici.
Napravi program kojem ćemo preko Serial Monitora unijeti višeznamenkasti broj, a zatim će program ispisati broj koji smo unijeli, ali kodiran Morseovim kodom.

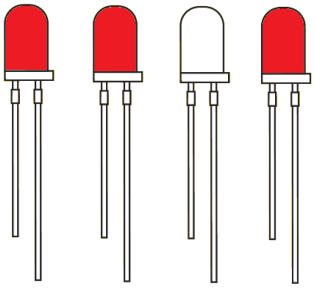


Cijeli dio programa gdje se ulazni String brojeva pretvara u izlazni String Moreseovih znakova potrebno je napraviti u obliku funkcije koja se poziva iz loopa. Npr.

```
String toMorse(String inputString){}
```

Preporuka je da tablicu kodiranja brojeva 0-9 u Morseove znakove staviš u dva niza.
Za stvaranje izlaznog niza preporučeno je koristiti funkciju `concat()`.

ZADATAK 9. Spoji 4 LED-ice na Arduino. Njih ćemo koristiti za binarni prikaz broja. Na *Serial Monitor* je potrebno javiti korisniku da unese jedan broj koji želi pretvoriti u binarni zapis. Pošto imamo četiri LED-ice na raspolaganju, neka taj broj bude u rasponu od 0 do 15.
Ako korisnik unese broj izvan raspona, na *Serial Monitor* se ispisuje poruka korisniku da ponovno upiše broj iz zadanog raspona.
Uneseni broj pretvori u binarni (koristi operacije nad bitovima) te zatim prikaži na LED-icama (1 svijetli, 0 ne svijetli).



PRIMJER IZLAZA ZA BROJ 13

Za operacije nad bitovima googlaj „bitwise operators arduino“, a posebno „`bitwise AND`“ i „`bitshift LEFT`“. Koristan ti može biti i „Bit Math Tutorial“ (prva stvar koju dobiješ kad to upišeš u google).

ZADATAK 10. ZA BRZE: Modificiraj prethodni program tako da se pretvaranje broja u stanja 4 ledice radi u funkciji `toLedSignals` koja je definirana na način:

```
void toLedSignals (int num, int* led1, int* led2, int* led3,
int* led4) {}
```

Unutar funkcije stanja led1, led2 itd mijenjaš pristupajući im koristeći pointere:

```
*led1=...;
```

Fukciju `toLedSignals` iz loopa ćeš pozvati predajući adrese led1, led2, led3, i led 4 na način:

```
toLedSignals (inputNum, &led1, &led2, &led3, &led4);
```

ZADATAK 11. ZA BRZE: Napravi program sličan programu iz 8 zadatka, ali sada ti unosi niz Morseovih znakova ('.' i '-'), koje ćeš odvajati razmakom ' '. Program mora dekodirati broj poslan Morseovim znakovima i ispisati koji je to bio broj. Također napravi da cijeli proces dekodiranja radi funkcija koju ćeš napisati npr. `int MorseToNum (String inputString){}`.

ZADATAK 12. ZA BRZE: Na Arduino spoji 2 LED-ice različite boje i potenciometar. Potenciometar postavi u nasumičnu poziciju. Funkcijom **map()** prebaci vrijednost potenciometra u interval [0, 20] i spremi u varijablu „blago“.

Sada je vrijeme za pogađanje. Korisnik preko *Serial Monitora* unosi broj između 0 i 20.

Ako je korisnik pogodio broj, tj. ako je unos korisnika == blago, obje LED-ice se pale.

Ako je unešeni broj <u>veći</u> od „blaga“, pali se jedna LED-ica, npr. crvena.

Ako je unešeni broj <u>manji</u> od „blaga“, pali se druga LED-ica, npr. plava.

Povećaj sada interval na [0, 100]. Koliko ti je trebalo da pogodiš broj? ☺

*Dodatno, traži od korisnika da nakon svakog pogotka generira novi nasumični broj, tj. ponovno pomakne potenciometar. Ponavljaj taj postupak proizvoljan broj puta za određen interval, pritom spremajući broj pokušaja prije svakog pogotka. Izračunaj prosjek broja pokušaja te ga ispiši na *Serial Monitor.* Slaže li se taj broj sa tvojim očekivanjem? Ako ne pokušaj objasniti zašto.