

Niz znakova (string)

- pohranjuje se kao jednodimenzionalno polje znakova
- zadnji znak je NUL ZNAK `\0`

npr.

J	A	B	U	K	A	\0
---	---	---	---	---	---	----

- funkcije za rad s nizovima znakova:

gets () → učitava se niz znakova s standardnog ulaza

strlen () → daje duljinu zadanog niza znakova

* **Primjer 1.**: Upisati rečenicu. Prebrojati i ispisati koliko sadrži slova **a**!

```
#include <stdio.h>
#include <string.h>
main(){
char r[50];
int i, d, br=0;
printf ("\n Upišite rečenicu max duljine 50 znakova:");
gets(r);
d=strlen(r);
for (i = 0; i < d; i++)
{
if (r[i]=='a' || r[i]=='A')
br++;
}
printf ("\n Rečenica sadrži %d slova a.", br);
}
```

* **Primjer 2.:** Upisati riječ. Izračunati i ispisati njenu duljinu – bez korištenja naredbe *strlen*!

```
#include <stdio.h>
#include <string.h>
main(){
    char r[20];
    int i, br=0;
    printf ("\n Upišite riječ max duljine 20 znakova:");
    scanf ("%s", r);
    for (i = 0; i < 20; i++)
    {
        if (r[i]!='\0')
            br++;
        else
            break;
    }
    printf ("\n Duljina riječi je %d.", br);
}
```

Riješi sljedeće zadatke:

1. Gornji zadatak riješi koristeći naredbu **while**!
2. Učitati riječ. Ispisati je s razmakom među znakovima (napomena – u naredbi za ispis ispred ili iza formata %c staviti razmak)!
npr. Upis: jabuka
Ispis: j a b u k a
3. Učitati riječ. Ispisati svako slovo u novi red (napomena – u naredbi za ispis ispred ili iza formata %c staviti \n)!
npr. Upis: jabuka
Ispis: j
a
b
u
k
a
4. Učitati riječ. Ispisati je naopako (napomena – naredba for za ispis riječi treba biti: for (i=d-1; i>=0; i--))!
5. Učitati rečenicu. Prebrojati i ispisati koliko sadrži riječi (napomena – broje se razmaci)!
6. Učitati rečenicu. Ispisati svaku riječ u novi red (napomena – kad se dođe do razmaka ispisati \n)!
7. Učitati rečenicu. Ispisati prvu riječ (napomena – kad se dođe do razmaka upotrijebiti naredbu **break** za izlaz iz petlje)!