

Napisati **rekurzivnu** funkciju koja zadani niz znakova ispisuje unatrag (npr. ako je zadan niz "Abeceda", funkcija treba ispisati "adecebA"). Funkcija treba imati prototip:

```
void ispisi_unatrag(char *niz);
```

```
void ispisi_unatrag(char *niz) {
    if (*niz) {
        ispisi_unatrag(niz+1);
        printf("%c", *niz);
    }
}
```

---

Napisati rekurzivnu funkciju koja će za zadani realan broj  $x > 0.5$  i zadani broj članova reda izračunati  $\ln x$  pomoću sljedećeg reda:

$$\ln x = \frac{x-1}{x} + \frac{(x-1)^2}{2 \cdot x^2} + \frac{(x-1)^3}{3 \cdot x^3} + \dots + \frac{(x-1)^i}{i \cdot x^i} + \dots$$

Koristeći te funkcije za rad sa stogom napisati **rekurzivnu** funkciju, koja će prepisati sve elemente iz jednog stoga u drugi, ali tako da se očuva redoslijed elemenata (nakon obavljanja treće funkcije prvi stog ostaje prazan). Stogovi se trećoj funkciji zadaju preko argumenata.

```
void prepisi(Stog *s1, Stog *s2) {
    int e;
    if (skini(s2, &e)) {
        prepisi(s1, s2);
        stavi(s1, e);
    }
}
```

Napisati **rekurzivnu** funkciju koja će pronaći indeks najvećeg elementa u zadanom cjelobrojnom polju. Napisati drugu **rekurzivnu** funkciju koja će, koristeći prvu funkciju, urediti zadano polje po veličini. Druga funkcija treba raditi slijedeće:

- pronaći najveći element (koristeći prvu funkciju)
- zamijeniti ga s elementom na početku polja
- pozvati se rekurzivno za ostatak polja

```
// vraća index maksimalnog elementa u polju
// Moram ići odozad jer ne smijem mijenjati polje tako da mi
vrijednosti indeksa
// budu očuvane
int max_rek(long *polje, int N) {
    int max2;
    // polje od jednog elementa
    if (N == 1) return 0;

    // najveći za polje bez zadnjeg elementa
    max2 = max_rek(polje, N-1);

    if (polje[N-1] > polje[max2]) return N-1;
    else return max2;
}

// sortira polje
void sort_rek(long *polje, int N) {
    int max;
    long pom;
    // polje od jednog elementa
    if (N == 1) return;

    // tražim najveći element
    max = max_rek(polje, N);

    // stavljam najvećeg na prvo mjesto
    pom = polje[max];
    polje[max] = polje[0];
    polje[0] = pom;

    // rekursivno se pozivam za ostatak polja
    sort_rek(polje+1, N-1);
}
```

Napisati funkciju koja će stvoriti novi stog koji će sadržavati sve one brojeve iz zadanog stoga koji su veći ili jednaki prosječnoj vrijednosti elemenata iz zadanog stoga. Poredak tih brojeva

u novom stogu nije bitan. Redoslijed podataka na zadanom stogu mora ostati nepromijenjen. Funkcija mora imati prototip:

```
Stog veci_od_prosjeka (Stog *zadani);
```