

Programiranje 2 – prvi kolokvij, 22. 4. 2016.

Rezultati i uvidi u kolokvije: Rezultati u četvrtak, 28.4., navečer na webu, a uvidi u ponedjeljak, 2.5., u 14 sati.

Upute: Na kolokviju je dozvoljeno koristiti samo pribor za pisanje i brisanje, te službeni podsjetnik. Kalkulatori, razne neslužbene tablice, papiri i sl., nisu dozvoljeni! **Mobitele ugasite i spremite!** Sva rješenja napišite isključivo na papire sa zadacima, jer jedino njih predajete. Ne zaboravite se **potpisati** na svim papirima! Skice smijete raditi i na drugim papirima koje će vam dati dežurni asistent. U svim zadacima zabranjeno je korištenje dodatnih nizova i standardne matematičke biblioteke (zaglavlje `math.h`), osim ako je u zadatku drugačije navedeno.

Zadatak 1. (20 + 5 bodova) Karta područja je zapisana u matrici $K \in \mathbb{Z}^{m,n}$ na način da svaki $k_{i,j}$ sadrži informaciju o lokaciji (i, j) . Planinar Luka želi doći od lokacije $A = (i_1, j_1)$ do lokacije $B = (i_2, j_2)$, krećući se proizvoljno između susjednih lokacija na karti. U jednom koraku, iz neke lokacije može se doći u (najviše) 8 susjednih lokacija ($\uparrow, \downarrow, \leftarrow, \rightarrow, \swarrow, \searrow, \nearrow, \nwarrow$). Planinar se ne smije kretati izvan karte, jer bi se u tom slučaju mogao izgubiti.

Međutim, planinar ima ograničenu količinu zaliha u svojoj planinarskoj torbi. Ukoliko je $k_{i,j} < 0$, planinar potroši $k_{i,j}$ zaliha da dođe do lokacije (i, j) iz jedne od susjednih lokacija. Ako je $k_{i,j} > 0$, onda se na toj lokaciji nalazi planinarski dom, te planinar, umjesto da potroši određenu količinu namirnica, može nadomjestiti točno $k_{i,j}$ namirnica, s tim da vrijedi

$$|k_{i,j}| < \min\{|k_{i-1,j}|, |k_{i+1,j}|, |k_{i,j-1}|, |k_{i,j+1}|, |k_{i-1,j-1}|, |k_{i+1,j-1}|, |k_{i-1,j+1}|, |k_{i+1,j+1}|\}.$$

Dva planinarska doma se nikada ne nalaze na susjednim lokacijama na karti, da maksimiziraju profit. Iz istog razloga se planinarski domovi nikada ne nalaze na rubu karte. Ukoliko je $k_{i,j} = 0$, na toj lokaciji se nalazi neprohodan teren kojeg planinar mora zaobići.

- Napišite rekurzivnu funkciju koja računa broj načina na koje Luka može doći od lokacije A do lokacije B sa zadanom početnom količinom namirnica $namirnica \in \mathbb{N}$.
- Demonstrirajte poziv funkcije u glavnom programu, s time da se parametri $namirnica$, dimenzije karte, početna i ciljna lokacija učitavaju kao argumenti komandne linije.

Za rješavanje drugog podzadatka nije nužno da riješite prvi, ali **je nužno** da napišete barem zaglavlje funkcije iz prvog podzadatka.

Programiranje 2 – prvi kolokvij, 22. 4. 2016.

Zadatak 2. (15 bodova) Neka su m i n prirodni brojevi, takvi da je $m \leq 50$, $n \leq 30$. Napišite funkciju `izbacirdx` koja prima prirodne brojeve m , n i x , te matricu prirodnih brojeva tipa $m \times n$. Funkcija `izbacirdx` treba izbaciti sve retke kod kojih je najveći zajednički djelitelj svih elemenata u tom retku jednak broju x . Izbaciti redak znači pomaknuti preostale retke i popraviti broj redaka.

Napišite i odsječak glavnog programa koji sadrži samo deklaraciju matrice za sve dozvoljene m i n , te poziv funkcije na toj matrici za $m = 17$, $n = 5$.

Programiranje 2 – prvi kolokvij, 22. 4. 2016.

Zadatak 3. (7 + 8 bodova)

Napišite program koji učitava broj prodavača n jedne trgovine, a zatim, redom, iznose računa koje je pojedini prodavač naplatio. Iznos je realni broj strogo veći od nula. Između iznosa svaka dva prodavača unosi se 0.

- (a) Potrebno je kreirati dinamički alocirano polje `prodavaci`, u kojem će, u i -tom retku, na prvom mjestu biti oznaka prodavača (na početku je to upravo i), zatim iznosi naplaćenih računa i -tog prodavača, $i = 0, \dots, n - 1$, a posljedni element svakog retka ima vrijednost 0 i označava kraj retka. Pri tom, alocirajte točno onoliko memorije koliko vam je potrebno.
- (b) Napišite funkciju `prodavaci_sort` u kojoj ćete sortirati polje `prodavaci`, tako da poredak redaka bude *silazan* prema ukupnom naplaćenom iznosu pojedinog prodavača. Potom treba ispisati oznake prodavača, silazno po tom kriteriju (vrijednosti ukupno naplaćenih računa). Ukoliko neki prodavač nije imao niti jedan naplaćeni račun, stavite ga na kraj polja. Uz svakog prodavača, dodatno treba ispisati iznose svih naplaćenih računa u poretku u kojem su učitavani. Funkciju trebate pozvati u glavnom programu.

Argumente funkcije i povratnu vrijednost sami određujete. Ukoliko ne znate napisati funkciju, napišite prototip funkcije i pozovite ju u glavnom programu.

U ovom zadatku možete koristiti dodatne funkcije.

Programiranje 2 – prvi kolokvij, 22. 4. 2016.

Zadatak 4. (15 bodova) Napišite funkciju

```
char *cenzura(const char *tekst, const char *rijec)
```

koja vraća pokazivač na string dobiven iz stringa `tekst`, tako da je u njemu svako pojavljivanje stringa `rijec` zamijenjeno znakovima '*' (onoliko mnogo znakova, kolika je duljina stringa `rijec`). Kod cenzure ne treba praviti razliku između malih i velikih slova u tekstu.

Primjer: Za stringove `tekst="Luka Kovac radi u kovacnici."` i `rijec="Kovac"`, treba vratiti pokazivač na string `"Luka ***** radi u *****nici."`.

Napomene: Smijete koristiti funkcije iz biblioteka `string.h` i `cctype.h`. Ne smijete mijenjati stringove koji su argumenti funkcije.

Programiranje 2 – prvi kolokvij, 22. 4. 2016.

Rezultati i uvidi u kolokvije: Rezultati u četvrtak, 28.4., navečer na webu, a uvidi u ponedjeljak, 2.5., u 14 sati.

Upute: Na kolokviju je dozvoljeno koristiti samo pribor za pisanje i brisanje, te službeni podsjetnik. Kalkulatori, razne neslužbene tablice, papiri i sl., nisu dozvoljeni! **Mobitele ugasite i spremite!** Sva rješenja napišite isključivo na papire sa zadacima, jer jedino njih predajete. Ne zaboravite se **potpisati** na svim papirima! Skice smijete raditi i na drugim papirima koje će vam dati dežurni asistent. U svim zadacima zabranjeno je korištenje dodatnih nizova i standardne matematičke biblioteke (zaglavlje `math.h`), osim ako je u zadatku drugačije navedeno.

Zadatak 1. (20 + 5 bodova) Kapetan Toni upravlja putničkim brodom. On posjeduje kartu koju reprezentiramo matricom $K \in \mathbb{Z}^{m,n}$ na način da svaki $k_{i,j}$ sadrži informaciju o lokaciji (i, j) . Kapetan Toni želi doći od lokacije $A = (i_1, j_1)$ do lokacije $B = (m - 1, n - 1)$, vozeći brod proizvoljno između susjednih lokacija na karti. U jednom koraku, iz neke lokacije može se doći u (najviše) 8 susjednih lokacija ($\uparrow, \downarrow, \leftarrow, \rightarrow, \swarrow, \searrow, \nearrow, \nwarrow$). Kapetan ne smije voziti po putevima izvan karte, pošto ne poznaje te vode i može nasukati brod.

Ukoliko je $k_{i,j} > 0$, onda vožnja iz susjedne lokacije u lokaciju (i, j) traje $k_{i,j}$ vremena. Ukoliko je $k_{i,j} = 0$, tada su vode na zadanoj lokaciji pune grebena i pličaka koje kapetan Toni definitivno treba izbjeći. Sve lokacije s $k_{i,j} < 0$ sadrže vrtlog, zbog čega kapetan Toni trenutno stiže iz susjednih lokacija na lokaciju (i, j) . Međutim, put iz vrtloga (i, j) u svaku susjednu lokaciju (p, q) , oduzima $2 \cdot |k_{i,j}| + k_{p,q}$ vremena. Dva vrtloga se nikada ne nalaze na susjednim lokacijama na karti.

- Napišite rekurzivnu funkciju koja računa broj načina na koje kapetan Toni može prevesti putnike s lokacije A do cilja B , ako to mora napraviti u zadanom maksimalnom vremenu $t \in \mathbb{N}$.
- Demonstrirajte poziv funkcije u glavnom programu, s time da se parametri t , dimenzije karte i koordinate početne luke učitavaju kao argumenti komandne linije.

Za rješavanje drugog podzadatka nije nužno da riješite prvi, ali **je nužno** da napišete barem zaglavlje funkcije iz prvog podzadatka.

Programiranje 2 – prvi kolokvij, 22. 4. 2016.

Zadatak 2. (15 bodova) Neka su m i n prirodni brojevi, takvi da je $m \leq 40$, $n \leq 20$. Napišite funkciju `izbacisvx` koja prima prirodne brojeve m , n i x , te matricu prirodnih brojeva tipa $m \times n$. Funkcija `izbacisvx` treba izbaciti sve stupce kod kojih je najmanji zajednički višekratnik svih elemenata u tom stupcu jednak broju x . Izbaciti stupac znači pomaknuti preostale stupce i popraviti broj stupaca.

Napišite i odsječak glavnog programa koji sadrži samo deklaraciju matrice za sve dozvoljene m i n , te poziv funkcije na toj matrici za $m = 7$, $n = 17$.

Programiranje 2 – prvi kolokvij, 22. 4. 2016.

Zadatak 3. (7 + 8 bodova)

Napišite program koji učitava broj vozača n koji sudjeluju na kvalifikacijama za jednu kružnu utrku, a zatim, redom, vremena za odvožen jedan krug pojedinog vozača. Svaki vozač ima pravo proizvoljno mnogo puta odvoziti krug. Vrijeme je realni broj strogo veći od nula. Između vremena svaka dva vozača unosi se 0.

- (a) Potrebno je kreirati dinamički alocirano polje `vozaci`, u kojem će, u i -tom retku, na prvom mjestu biti oznaka vozača (na početku je to upravo i), zatim vremena odvoženih krugova i -tog vozača, $i = 0, \dots, n - 1$, a posljedni element svakog retka ima vrijednost 0 i označava kraj retka. Pri tom, alocirajte točno onoliko memorije koliko vam je potrebno.
- (b) Napišite funkciju `ispis_vozaci` u kojoj ćete sortirati polje `vozaci`, tako da poredak redaka bude *uzlazan* prema minimalnom vremenu za jedan odvožen krug pojedinog vozača. Potom treba ispisati oznake vozača, uzlazno po tom kriteriju (minimalno vrijeme za jedan odvožen krug). Ukoliko neki vozač nije imao niti jedan odvožen krug, stavite ga na kraj polja. Uz svakog vozača, dodatno treba ispisati vremena svih odvoženih krugova u poretku u kojem su učitavani. Funkciju trebate pozvati u glavnom programu.

Argumente funkcije i povratnu vrijednost sami određujete. Ukoliko ne znate napisati funkciju, napišite prototip funkcije i pozovite ju u glavnom programu.

U ovom zadatku možete koristiti dodatne funkcije.

Programiranje 2 – prvi kolokvij, 22. 4. 2016.

Zadatak 4. (15 bodova) Napišite funkciju

```
char *naglasici(const char *tekst, const char *rijec)
```

koja vraća pokazivač na string dobiven iz stringa `tekst`, tako da je u njemu svako pojavljivanje stringa `rijec`, koji će se sastojati samo od malih slova (ne treba provjeravati), zamijenjeno odgovarajućim velikim slovima. Kod naglašavanja riječi ne treba praviti razliku između malih i velikih slova u tekstu.

Primjer: Za stringove `tekst="vatra prekinula koncert grupe Vatra"` i `rijec="vatra"`, treba vratiti pokazivač na string `"VATRA prekinula koncert grupe VATRA"`.

Napomene: Smijete koristiti funkcije iz biblioteka `string.h` i `ctype.h`. Ne smijete mijenjati stringove koji su argumenti funkcije.

Programiranje 2 – prvi kolokvij, 22. 4. 2016.

Rezultati i uvidi u kolokvije: Rezultati u četvrtak, 28.4., navečer na webu, a uvidi u ponedjeljak, 2.5., u 14 sati.

Upute: Na kolokviju je dozvoljeno koristiti samo pribor za pisanje i brisanje, te službeni podsjetnik. Kalkulatori, razne neslužbene tablice, papiri i sl., nisu dozvoljeni! **Mobitele ugasite i spremite!** Sva rješenja napišite isključivo na papire sa zadacima, jer jedino njih predajete. Ne zaboravite se **potpisati** na svim papirima! Skice smijete raditi i na drugim papirima koje će vam dati dežurni asistent. U svim zadacima zabranjeno je korištenje dodatnih nizova i standardne matematičke biblioteke (zaglavlje `math.h`), osim ako je u zadatku drugačije navedeno.

Zadatak 1. (20 + 5 bodova) Pero se kreće po labirintu zapisanom u matrici $A \in \mathbb{Z}^{m,n}$ na način da svaki $a_{i,j}$ sadrži informaciju o lokaciji (i, j) . Pero želi doći od početka labirinta $A = (0, 0)$ do kraja labirinta $B = (m - 1, n - 1)$, krećući se proizvoljno između susjednih lokacija na karti. U jednom koraku, iz neke lokacije može se doći u (najviše) 8 susjednih lokacija ($\uparrow, \downarrow, \leftarrow, \rightarrow, \swarrow, \searrow, \nearrow, \nwarrow$). Labirint je okružen velikom provalijom, stoga ga Pero ne smije napuštati.

Pero ima određenu količinu energije koja mu omogućava da prijeđe preko prepreka koje se nalaze na određenim lokacijama u labirintu. Ukoliko je $a_{i,j} < 0$, na toj lokaciji je prepreka koja Peri oduzima $a_{i,j}$ energije. Ukoliko je $a_{i,j} = 0$, na toj lokaciji se nalazi prepreka preko koje Pero ne može proći. Ukoliko je $a_{i,j} > 0$, ta lokacija sadrži voće koje Peri daje $a_{i,j}$ energije, s tim da vrijedi

$$|a_{i,j}| < \min\{|a_{i-1,j}|, |a_{i+1,j}|, |a_{i,j-1}|, |a_{i,j+1}|, |a_{i-1,j-1}|, |a_{i+1,j-1}|, |a_{i-1,j+1}|, |a_{i+1,j+1}|\}.$$

Voće se nikada ne nalazi na dvije susjedne lokacije ili na rubovima labirinta.

- Napišite rekursivnu funkciju koja računa broj načina na koje Pero može doći od početka labirinta A do kraja labirinta B , sa zadanom početnom količinom energije $energija \in \mathbb{N}$.
- Demonstrirajte poziv funkcije u glavnom programu, s time da se parametri $energija$ i dimenzije labirinta učitavaju kao argumenti komandne linije.

Za rješavanje drugog podzadatka nije nužno da riješite prvi, ali **je nužno** da napišete barem zaglavlje funkcije iz prvog podzadatka.

Programiranje 2 – prvi kolokvij, 22. 4. 2016.

Zadatak 2. (15 bodova) Neka su m i n prirodni brojevi, takvi da je $m \leq 30$, $n \leq 50$. Napišite funkciju `izbacisdx` koja prima prirodne brojeve m , n i x , te matricu prirodnih brojeva tipa $m \times n$. Funkcija `izbacisdx` treba izbaciti sve stupce kod kojih je najveći zajednički djelitelj svih elemenata u tom stupcu jednak broju x . Izbaciti stupac znači pomaknuti preostale stupce i popraviti broj stupaca.

Napišite i odsječak glavnog programa koji sadrži samo deklaraciju matrice za sve dozvoljene m i n , te poziv funkcije na toj matrici za $m = 19$, $n = 7$.

Programiranje 2 – prvi kolokvij, 22. 4. 2016.

Zadatak 3. (7 + 8 bodova)

Napišite program koji učitava broj ronioca n koji sudjeluju na natjecanju držanja daha pod vodom (tzv. statika), a zatim, redom, postignuta vremena pojedinog ronioca. Svaki ronIOC ima pravo na proizvoljno mnogo pokušaja. Vrijeme je realni broj strogo veći od nula. Između vremena svaka dva ronioca unosi se 0.

- (a) Potrebno je kreirati dinamički alocirano polje `ronioci`, u kojem će, u i -tom retku, na prvom mjestu biti oznaka ronioca (na početku je to upravo i), zatim postignuta vremena držanja daha pod vodom i -tog ronioca, $i = 0, \dots, n - 1$, a posljedni element svakog retka ima vrijednost 0 i označava kraj retka. Pri tom, alocirajte točno onoliko memorije koliko vam je potrebno.
- (b) Napišite funkciju `sort_ronioci` u kojoj ćete sortirati polje `ronioci`, tako da poredak redaka bude *silazan* prema maksimalnom vremenu držanja daha (u jednom pokušaju) pojedinog ronioca. Potom treba ispisati oznake ronioca, silazno po tom kriteriju (maksimalno vrijeme držanja daha). Ukoliko neki ronIOC nije imao niti jedan pokušaj, stavite ga na kraj polja. Uz svakog ronioca, dodatno treba ispisati vremena svih pokušaja u poretku u kojem su učitavani. Funkciju trebate pozvati u glavnom programu.

Argumente funkcije i povratnu vrijednost sami određujete. Ukoliko ne znate napisati funkciju, napišite prototip funkcije i pozovite ju u glavnom programu.

U ovom zadatku možete koristiti dodatne funkcije.

Programiranje 2 – prvi kolokvij, 22. 4. 2016.

Zadatak 4. (15 bodova) Napišite funkciju

```
char *spijuni(const char *tekst, const char *rijec, const char* sifra)
```

koja vraća pokazivač na string dobiven iz stringa `tekst`, tako da je u njemu svako pojavljivanje stringa `rijec` zamijenjeno stringom `sifra`. Kod šifriranja riječi ne treba praviti razliku između malih i velikih slova u tekstu. Stringovi `rijec` i `sifra` bit će iste duljine i to ne treba provjeravati.

Primjer: Za stringove `tekst="Unisten tenk iz Tenkovske divizije"`, `rijec="tenk"` i `sifra="slon"`, treba vratiti pokazivač na string `"Unisten slon iz slonovske divizije"`.

Napomene: Smijete koristiti funkcije iz biblioteka `string.h` i `ctype.h`. Ne smijete mijenjati stringove koji su argumenti funkcije.

Programiranje 2 – prvi kolokvij, 22. 4. 2016.

Rezultati i uvidi u kolokvije: Rezultati u četvrtak, 28.4., navečer na webu, a uvidi u ponedjeljak, 2.5., u 14 sati.

Upute: Na kolokviju je dozvoljeno koristiti samo pribor za pisanje i brisanje, te službeni podsjetnik. Kalkulatori, razne neslužbene tablice, papiri i sl., nisu dozvoljeni! **Mobitele ugasite i spremite!** Sva rješenja napišite isključivo na papire sa zadacima, jer jedino njih predajete. Ne zaboravite se **potpisati** na svim papirima! Skice smijete raditi i na drugim papirima koje će vam dati dežurni asistent. U svim zadacima zabranjeno je korištenje dodatnih nizova i standardne matematičke biblioteke (zaglavlje `math.h`), osim ako je u zadatku drugačije navedeno.

Zadatak 1. (20 + 5 bodova) Pilot Marko upravlja putničkim avionom. Navigacijski uređaj u pilotskoj kabini sadrži kartu reprezentiranu matricom $N \in \mathbb{Z}^{m,n}$ na način da svaki $n_{i,j}$ sadrži informaciju o lokaciji (i, j) . Pilot Marko želi prevesti putnike od lokacije $A = (i_1, j_1)$ do lokacije $B = (i_2, j_2)$, krećući se proizvoljno između susjednih lokacija na karti. U jednom koraku, iz neke lokacije može se doći u (najviše) 8 susjednih lokacija ($\uparrow, \downarrow, \leftarrow, \rightarrow, \swarrow, \searrow, \nearrow, \nwarrow$). Pilot ne smije letjeti izvan ruta definiranih u karti navigacijskog uređaja, jer u tom slučaju mora tražiti pomoć od iznimno zaposlenih kontrolora leta.

Ukoliko je $n_{i,j} > 0$, onda vožnja iz susjedne lokacije u lokaciju (i, j) troši količinu kerozina $n_{i,j}$. Ukoliko je $n_{i,j} = 0$, tada je na tom području snažno nevrijeme koje pilot mora izbjeći. Ukoliko je $n_{i,j} < 0$, tada iz svake susjedne lokacije možemo doći u lokaciju (i, j) , uz potrošnju kerozina $\lfloor \frac{n_{i,j}}{2} \rfloor$.

- Napišite rekurzivnu funkciju koja računa broj načina na koje pilot Marko može prevesti putnike s lokacije A do cilja B , ako avion ima zadanu količinu kerozina $v \in \mathbb{N}$, koju smije potrošiti.
- Demonstrirajte poziv funkcije u glavnom programu, s time da se parametri v , dimenzije karte, koordinate početne i ciljne zračne luke učitavaju kao argumenti komandne linije.

Za rješavanje drugog podzadatka nije nužno da riješite prvi, ali **je nužno** da napišete barem zaglavlje funkcije iz prvog podzadatka.

Programiranje 2 – prvi kolokvij, 22. 4. 2016.

Zadatak 2. (15 bodova) Neka su m i n prirodni brojevi, takvi da je $m \leq 20$, $n \leq 40$. Napišite funkciju `izbacirvx` koja prima prirodne brojeve m , n i x , te matricu prirodnih brojeva tipa $m \times n$. Funkcija `izbacirvx` treba izbaciti sve retke kod kojih je najmanji zajednički višekratnik svih elemenata u tom retku jednak broju x . Izbaciti redak znači pomaknuti preostale retke i popraviti broj redaka.

Napišite i odsječak glavnog programa koji sadrži samo deklaraciju matrice za sve dozvoljene m i n , te poziv funkcije na toj matrici za $m = 5$, $n = 19$.

Programiranje 2 – prvi kolokvij, 22. 4. 2016.

Zadatak 3. (7 + 8 bodova)

Smještaj u pojedinom apartmanu moguće je rezervirati preko n turističkih agencija. Napišite program koji učitava broj agencija n , a zatim, redom, cijene rezervacija koje su napravljene preko pojedine agencije za razne apartmane. Svaka agencija može nuditi proizvoljno mnogo apartmana. Cijena rezervacije je realni broj strogo veći od nula. Između cijena rezervacija svake dvije agencije unosi se 0.

- (a) Potrebno je kreirati dinamički alocirano polje `agencije`, u kojem će, u i -tom retku, na prvom mjestu biti oznaka agencije (na početku je to upravo i), zatim cijene rezervacija za apartmane kod i -te agencije, $i = 0, \dots, n - 1$, a posljedni element svakog retka ima vrijednost 0 i označava kraj retka. Pri tom, alocirajte točno onoliko memorije koliko vam je potrebno.
- (b) Napišite funkciju `agencije_ispis` u kojoj ćete sortirati polje `agencije`, tako da poredak redaka bude *silazan* prema zbroju cijena svih rezervacija za apartmane pojedine agencije. Potom treba ispisati oznake agencija, silazno po tom kriteriju (zbroj cijena svih rezervacija). Ukoliko neka agencija nije imala niti jednu rezervaciju, stavite ju na kraj polja. Uz svaku agenciju, dodatno treba ispisati cijene svih rezervacija u poretku u kojem su učitavane. Funkciju trebete pozvati u glavnom programu.

Argumente funkcije i povratnu vrijednost sami određujete. Ukoliko ne znate napisati funkciju, napišite prototip funkcije i pozovite ju u glavnom programu.

U ovom zadatku možete koristiti dodatne funkcije.

Programiranje 2 – prvi kolokvij, 22. 4. 2016.

Zadatak 4. (15 bodova) Napišite funkciju

```
char *prekrizi(const char *tekst, const char *rijec)
```

koja vraća pokazivač na string dobiven iz stringa `tekst`, tako da je u njemu svako pojavljivanje stringa `rijec` zamijenjeno znakovima '-' (onoliko mnogo znakova, kolika je duljina stringa `rijec`). Kod križanja ne treba praviti razliku između malih i velikih slova u tekstu.

Primjer: Za stringove `tekst="Ana voli Milovana"` i `rijec="Ana"`, treba vratiti pokazivač na string `--- voli Milov---`.

Napomene: Smijete koristiti funkcije iz biblioteka `string.h` i `cctype.h`. Ne smijete mijenjati stringove koji su argumenti funkcije.