

Programiranje 2 – prvi kolokvij, 28. 4. 2017.

Rezultati i uvidi u kolokvije: Rezultati u četvrtak, 4.5., navečer na webu, a uvidi u petak, 5.5., u 12 sati u (104).

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 isključite 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 bodova) Napišite program koji učitava prirodne brojeve m i n , te ispisuje koliko ima “dobrih” rastava oblika

$$(x_1, y_1) + (x_2, y_2) + \dots + (x_{2k}, y_{2k}) = (m, n),$$

pri čemu je k prirodan broj, te su x_i i y_i , $i = 1, \dots, 2k$, prirodni brojevi. Poredak sumanada u parovima s lijeve strane nije bitan, tj. možete uzeti da su x_i nepadajući i y_i nepadajući. Nadalje, u svakom “dobrom” rastavu postoji par (x_i, y_i) tako da su x_i i y_i relativno prosti.

Programiranje 2 – prvi kolokvij, 28. 4. 2017.

Zadatak 2. (15 bodova) Za matricu X tipa $m \times n$, definiramo njezinu glavnu podmatricu reda k , $k \in \mathbb{N}$, kao matricu tipa $k \times k$ čiji elementi su elementi x_{ij} matrice X , takvi da je $1 \leq i, j \leq k$. Napišite funkciju `BlokDijagonalna` koja za ulazne parametre ima matricu realnih brojeva X , njezin broj redaka m i broj stupaca n , gdje je $m \leq 55$, $n \leq 35$. Funkcija treba ispisati najveću glavnu podmatricu matrice X , koja je reda $2k$ i oblika $\left[\begin{array}{c|c} A & 0 \\ \hline 0 & A \end{array} \right]$, gdje je A neka kvadratna matrica reda k , za $k \geq 1$. Ako takva podmatrica ne postoji, treba ispisati odgovarajuću poruku.

Programiranje 2 – prvi kolokvij, 28. 4. 2017.

Zadatak 3. (14 + 6 = 20 bodova) Napišite program koji implementira igru *pronađi mrkve*. Ploča je polje znakova \$ i *. Korisnik u svakom koraku izabere točno jedan indeks koji je sadržan u polju (duljina polja se ispisuje kao poruka na ekran). Ukoliko je izabran znak \$, tada je otkrivena mrkva. Igra generira novu ploču koja je identična staroj, samo ne sadrži izabrani element. Npr., za ploču \$**\$\$* i indeks 3, nova ploča izgleda \$**\$*. Ukoliko je izabran znak *, otkriven je busen trave koji sadrži mrkvinu muhu. Igra treba generirati novu ploču na sljedeći način: izabrani element, prvi lijevi i prvi desni element (ako postoje) treba izbaciti iz polja. Ukoliko se na lijevoj ili desnoj poziciji nalazi ponovo mrkvina muha, postupak treba ponoviti. Za identičan primjer ploče i odabrani indeks 1, program treba generirati polje \$*. Igra završava kada na ploči više nema mrkvi.

- Napišite funkciju

```
char **igraj(char **buseni, int *korak, int *br_mrkvi, int **dim, int **index)
```

koja simulira opisanu igru, te sprema sve novodobivene ploče u niz nizova `buseni`, dok u dodatna jednodimenzionalna polja: `dim` - sprema veličinu svake ploče i `index` - sprema odabrane indekse od strane korisnika (parametri funkcije su adrese pokazivača na prvi element tih polja). Pokazivači `korak` i `br_mrkvi` sadrže adrese varijabli koje koristite za spremanje informacija o trenutnoj iteraciji igre i o broju preostalih mrkvi na ploči. Početna ploča je spremljena u polju koje se nalazi na indeksu 0 u nizu nizova `buseni`, pripadna duljina je spremljena na indeksu 0 u polju `dim`, broj mrkvi je spremljen na adresi `br_mrkvi`, a početna vrijednost na adresi `korak` je 0.

Funkcija treba igru odigrati do kraja (tj. dok na trenutnoj ploči nema više mrkvi). U svakom koraku, funkcija učitava indeks `s` ulaza (ne treba provjeravati postoji li učitani indeks u odgovarajućem polju), a nakon toga se računa nova ploča i odgovarajuća duljina. Funkcija vraća završni niz nizova, nakon dodavanja svih ploča.

- Napišite funkciju `void ispisi(char **buseni, int *dim, int *index, int br_koraka)` koja ispisuje sve korake igre u kojima je izabrana mrkva. Funkcija, za svaki takav korak, ispisuje stanje ploče prije i nakon izabira mrkve.

Programiranje 2 – prvi kolokvij, 28. 4. 2017.

Zadatak 4. (12 + 3 = 15 bodova) U jednom stringu čuva se niz riječi. Riječi su odvojene znakom '+' koji dolazi **iza** svake riječi (uključujući zadnju riječ u nizu). Pritom, kao riječ smatramo svaki niz znakova koji ne sadrži znak '+'.

- (a) Napišite funkciju `char* izbaci(char *recenica)` koja prima opisani string, te iz njega izbacuje riječ koja sadrži najveći broj velikih slova. Ukoliko je string duljine nula, potrebno je vratiti vrijednost `NULL`. Funkcija mora alocirati novi string te zapisati izbrisani riječ u njega. Funkcija vraća pokazivač na taj string.
- (b) Napišite glavni program u kojem pozivate funkciju `izbaci` sve dok ne ispraznite neki početni string. Nakon svakog poziva funkcije potrebno je ispisati vraćenu riječ te osloboditi zauzetu memoriju.

Primjer: Za dani string "OVO+ko1OKvij+rijec+", funkcija će vratiti "OVO", a string će nakon poziva biti "ko1OKvij+rijec+".

Napomena: Za rješavanje drugog podzadatka nije nužno da riješite prvi. **Zabranjeno** je korištenje funkcija iz zaglavlja `string.h`, a smijete koristiti funkcije iz zaglavlja `ctype.h`.

Programiranje 2 – prvi kolokvij, 28. 4. 2017.

Rezultati i uvidi u kolokvije: Rezultati u četvrtak, 4.5., navečer na webu, a uvidi u petak, 5.5., u 12 sati u (104).

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 isključite 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 bodova) Napišite program koji učitava prirodne brojeve m i n , te ispisuje koliko ima “dobrih” rastava oblika

$$(x_1, y_1) + (x_2, y_2) + \dots + (x_{2k-1}, y_{2k-1}) = (m, n),$$

pri čemu je k prirodan broj, te su x_i i y_i , $i = 1, \dots, 2k - 1$, prirodni brojevi. Poredak sumanada u parovima s lijeve strane nije bitan, tj. možete uzeti da su x_i nepadajući i y_i nepadajući. Nadalje, u svakom “dobrom” rastavu postoji par (x_i, y_i) tako da su x_i i y_i relativno prosti.

Programiranje 2 – prvi kolokvij, 28. 4. 2017.

Zadatak 2. (15 bodova) Za matricu X tipa $m \times n$, definiramo njezinu glavnu podmatricu reda k , $k \in \mathbb{N}$, kao matricu tipa $k \times k$ čiji elementi su elementi x_{ij} matrice X , takvi da je $1 \leq i, j \leq k$. Napišite funkciju `BlokGornjeTrokutasta` koja za ulazne parametre ima matricu realnih brojeva X , njezin broj redaka m i broj stupaca n , gdje je $m \leq 55$, $n \leq 35$. Funkcija treba ispisati najveću glavnu podmatricu matrice X , koja je reda $2k$ i oblika $\left[\begin{array}{c|c} B & B \\ \hline 0 & B \end{array} \right]$, gdje je B neka kvadratna matrica reda k , za $k \geq 1$. Ako takva podmatrica ne postoji, treba ispisati odgovarajuću poruku.

Programiranje 2 – prvi kolokvij, 28. 4. 2017.

Zadatak 3. (14 + 6 = 20 bodova) Napišite program koji implementira igru *pronadi krumpira*. Ploča je polje znakova o i -. Korisnik u svakom koraku izabere točno jedan indeks koji je sadržan u polju (duljina polja se ispisuje kao poruka na ekran). Ukoliko je izabran znak o, tada je otkriven krumpir. Igra generira novu ploču koja je identična staroj, samo ne sadrži izabrani element. Npr., za ploču o - - o - o o - i indeks 3, nova ploča izgleda o - - - o o -. Ukoliko je izabran znak -, otkriven je busen trave koji sadrži krumpirovu zlaticu. Igra treba generirati novu ploču na sljedeći način: izabrani element, i sljedeća dva elementa s desne strane (ako postoje) treba izbaciti iz polja. Ukoliko se na nekoj od sljedećih pozicija nalazi ponovo krumpirova zlatica, postupak treba ponoviti (tj. treba izbaciti sljedeća dva elementa, krećući od indeksa elementa koji sadrži krumpirovu zlaticu). Za identičan primjer ploče i odabrani indeks 1, program treba generirati polje o -. Igra završava kada na ploči više nema krumpira.

- Napišite funkciju

```
char **igraj(char **buseni, int *korak, int *br_krumpira, int **dim, int **index)
```

koja simulira opisanu igru, te sprema sve novodobivene ploče u niz nizova `buseni`, dok u dodatna jednodimenzionalna polja: `dim` - sprema veličinu svake ploče i `index` - sprema odabrane indekse od strane korisnika (parametri funkcije su adrese pokazivača na prvi element tih polja). Pokazivači `korak` i `br_krumpira` sadrže adrese varijabli koje koristite za spremanje informacija o trenutnoj iteraciji igre i o broju preostalih krumpira na ploči. Početna ploča je spremljena u polju koje se nalazi na indeksu 0 u nizu nizova `buseni`, pripadna duljina je spremljena na indeksu 0 u polju `dim`, broj krumpira je spremljen na adresi `br_krumpira`, a početna vrijednost na adresi `korak` je 0.

Funkcija treba igru odigrati do kraja (tj. dok na trenutnoj ploči nema više krumpira). U svakom koraku, funkcija učitava indeks s ulaza (ne treba provjeravati postoji li učitani indeks u odgovarajućem polju), a nakon toga se računa nova ploča i odgovarajuća duljina. Funkcija vraća završni niz nizova, nakon dodavanja svih ploča.

- Napišite funkciju `void ispisi(char **buseni, int *dim, int *index, int br_koraka)` koja ispisuje sve korake igre u kojima je izabrana krumpirova zlatica. Funkcija, za svaki takav korak, ispisuje stanje ploče prije i nakon izabira krumpirove zlatice.

Programiranje 2 – prvi kolokvij, 28. 4. 2017.

Zadatak 4. (12 + 3 = 15 bodova) U jednom stringu čuva se niz riječi. Riječi su odvojene znakom ' '; ' koji dolazi **ispred** svake riječi (uključujući prvu riječ u nizu). Pritom, kao riječ smatramo svaki niz znakova koji ne sadrži znak ' '; '.

- (a) Napišite funkciju `char* filtriraj(char *niz_rj)` koja prima opisani string, te iz njega izbacuje riječ koja sadrži najmanji broj dekadskih znamenaka. Ukoliko je string duljine nula, potrebno je vratiti vrijednost `NULL`. Funkcija mora alocirati novi string te zapisati izbrisani riječ u njega. Funkcija vraća pokazivač na taj niz.
- (b) Napišite glavni program u kojem pozivate funkciju `filtriraj` sve dok ne ispraznite neki početni string. Nakon svakog poziva funkcije potrebno je ispisati vraćenu riječ te osloboditi zauzetu memoriju.

Primjer: Za dani string `" ;a123;12rr32;rijec"`, funkcija će vratiti `"rijec"`, a string će nakon poziva biti `" ;a123;12rr32"`.

Napomena: Za rješavanje drugog podzadatka nije nužno da riješite prvi. **Zabranjeno** je korištenje funkcija iz zaglavlja `string.h`, a smijete koristiti funkcije iz zaglavlja `cctype.h`.

Programiranje 2 – prvi kolokvij, 28. 4. 2017.

Rezultati i uvidi u kolokvije: Rezultati u četvrtak, 4.5., navečer na webu, a uvidi u petak, 5.5., u 12 sati u (104).

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 isključite 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 bodova) Napišite program koji učitava prirodne brojeve m i n , te ispisuje koliko ima “dobrih” rastava oblika

$$(x_1, y_1) + (x_2, y_2) + \dots + (x_{2k}, y_{2k}) = (m, n),$$

pri čemu je k prirodan broj, te su x_i i y_i , $i = 1, \dots, 2k$, prirodni brojevi. Poredak sumanada u parovima s lijeve strane nije bitan, tj. možete uzeti da su x_i nepadajući i y_i nepadajući. Nadalje, u svakom “dobrom” rastavu postoji par (x_i, y_i) tako da je najmanji zajednički višekratnik od x_i i y_i manji ili jednak $2x_i + 2y_i$.

Programiranje 2 – prvi kolokvij, 28. 4. 2017.

Zadatak 2. (15 bodova) Za matricu X tipa $m \times n$, definiramo njezinu glavnu podmatricu reda k , $k \in \mathbb{N}$, kao matricu tipa $k \times k$ čiji elementi su elementi x_{ij} matrice X , takvi da je $1 \leq i, j \leq k$. Napišite funkciju `BlokSporedna` koja za ulazne parametre ima matricu realnih brojeva X , njezin broj redaka m i broj stupaca n , gdje je $m \leq 55$, $n \leq 35$. Funkcija treba ispisati najveću glavnu podmatricu matrice X , koja je reda $2k$ i oblika $\left[\begin{array}{c|c} 0 & C \\ \hline C & 0 \end{array} \right]$, gdje je C neka kvadratna matrica reda k , za $k \geq 1$. Ako takva podmatrica ne postoji, treba ispisati odgovarajuću poruku.

Programiranje 2 – prvi kolokvij, 28. 4. 2017.

Zadatak 3. (14 + 6 = 20 bodova) Napišite program koji implementira igru *pronađi rotkvicu*. Ploča je polje znakova & i \$. Korisnik u svakom koraku izabere točno jedan indeks koji je sadržan u polju (duljina polja se ispisuje kao poruka na ekran). Ukoliko je izabran znak &, tada je otkrivena rotkvica. Igra generira novu ploču koja je identična staroj, samo ne sadrži izabrani element. Npr., za ploču &\$\$\$&&\$ i indeks 3, nova ploča izgleda &\$\$\$\$&\$. Ukoliko je izabran znak \$, otkriven je busen trave koji sadrži buhače. Igra treba generirati novu ploču na sljedeći način: izabrani element, i sljedeća dva elementa s lijeve strane (ako postoje) treba izbaciti iz polja. Ukoliko se na nekoj od sljedećih pozicija nalaze ponovo buhači, postupak treba ponoviti (tj. treba izbaciti sljedeća dva elementa, krećući od indeksa elementa koji sadrži buhače). Za identičan primjer ploče i odabrani indeks 4, program treba generirati polje &&\$. Igra završava kada na ploči više nema rotkvica.

- Napišite funkciju

```
char **igraj(char **buseni, int *korak, int *br_rotkvica, int **dim, int **index)
```

koja simulira opisanu igru, te sprema sve novodobivene ploče u niz nizova `buseni`, dok u dodatna jednodimenzionalna polja: `dim` - sprema veličinu svake ploče i `index` - sprema odabrane indekse od strane korisnika (parametri funkcije su adrese pokazivača na prvi element tih polja). Pokazivači `korak` i `br_rotkvica` sadrže adrese varijabli koje koristite za spremanje informacija o trenutnoj iteraciji igre i o broju preostalih rotkvica na ploči. Početna ploča je spremljena u polju koje se nalazi na indeksu 0 u nizu nizova `buseni`, pripadna duljina je spremljena na indeksu 0 u polju `dim`, broj rotkvica je spremljen na adresi `br_rotkvica`, a početna vrijednost na adresi `korak` je 0.

Funkcija treba igru odigrati do kraja (tj. dok na trenutnoj ploči nema više rotkvica). U svakom koraku, funkcija učitava indeks s ulaza (ne treba provjeravati postoji li učitani indeks u odgovarajućem polju), a nakon toga se računa nova ploča i odgovarajuća duljina. Funkcija vraća završni niz nizova, nakon dodavanja svih ploča.

- Napišite funkciju `void ispisi(char **buseni, int *dim, int *index, int br_koraka)` koja ispisuje sve korake igre u kojima je izabrana rotkvica. Funkcija, za svaki takav korak, ispisuje stanje ploče prije i nakon izabira rotkvice.

Programiranje 2 – prvi kolokvij, 28. 4. 2017.

Zadatak 4. (12 + 3 = 15 bodova) U jednom stringu čuva se niz riječi. Riječi su odvojene znakom '*' koji dolazi **iza** svake riječi (uključujući zadnju riječ u nizu). Pritom, kao riječ smatramo svaki niz znakova koji ne sadrži znak '*'.

- (a) Napišite funkciju `char* makni(char *nizrj)` koja prima opisani string, te iz njega izbacuje riječ koja sadrži najmanji broj znakova koji nisu slova niti dekadске znamenke. Ukoliko je string duljine nula, potrebno je vratiti vrijednost `NULL`. Funkcija mora alocirati novi string te zapisati izbrisani riječ u njega. Funkcija vraća pokazivač na taj niz.
- (b) Napišite glavni program u kojem pozivate funkciju `makni` sve dok ne ispraznite neki početni string. Nakon svakog poziva funkcije potrebno je ispisati vraćenu riječ te osloboditi zauzetu memoriju.

Primjer: Za dani string `"0-V-0*pos.ebna*neka-duga-rijec"`, funkcija će vratiti `"pos.ebna"`, a string će nakon poziva biti `"0-V-0*neka-duga-rijec"`.

Napomena: Za rješavanje drugog podzadatka nije nužno da riješite prvi. **Zabranjeno** je korištenje funkcija iz zaglavlja `string.h`, a smijete koristiti funkcije iz zaglavlja `ctype.h`.

Programiranje 2 – prvi kolokvij, 28. 4. 2017.

Rezultati i uvidi u kolokvije: Rezultati u četvrtak, 4.5., navečer na webu, a uvidi u petak, 5.5., u 12 sati u (104).

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 isključite 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 bodova) Napišite program koji učitava prirodne brojeve m i n , te ispisuje koliko ima “dobrih” rastava oblika

$$(x_1, y_1) + (x_2, y_2) + \dots + (x_{2k-1}, y_{2k-1}) = (m, n),$$

pri čemu je k prirodan broj, te su x_i i y_i , $i = 1, \dots, 2k - 1$, prirodni brojevi. Poredak sumanada u parovima s lijeve strane nije bitan, tj. možete uzeti da su x_i nepadajući i y_i nepadajući. Nadalje, u svakom “dobrom” rastavu postoji par (x_i, y_i) tako da je najmanji zajednički višekratnik od x_i i y_i manji ili jednak $2x_i + 2y_i$.

Programiranje 2 – prvi kolokvij, 28. 4. 2017.

Zadatak 2. (15 bodova) Za matricu X tipa $m \times n$, definiramo njezinu glavnu podmatricu reda k , $k \in \mathbb{N}$, kao matricu tipa $k \times k$ čiji elementi su elementi x_{ij} matrice X , takvi da je $1 \leq i, j \leq k$. Napišite funkciju `BlokDonjeTrokutasta` koja za ulazne parametre ima matricu realnih brojeva X , njezin broj redaka m i broj stupaca n , gdje je $m \leq 55$, $n \leq 35$. Funkcija treba ispisati najveću glavnu podmatricu matrice X , koja je reda $2k$ i oblika $\left[\begin{array}{c|c} D & 0 \\ \hline D & D \end{array} \right]$, gdje je D neka kvadratna matrica reda k , za $k \geq 1$. Ako takva podmatrica ne postoji, treba ispisati odgovarajuću poruku.

Programiranje 2 – prvi kolokvij, 28. 4. 2017.

Zadatak 3. (14 + 6 = 20 bodova) Napišite program koji implementira igru *pronadi tikve*. Ploča je polje znakova ! i -. Korisnik u svakom koraku izabere točno jedan indeks koji je sadržan u polju (duljina polja se ispisuje kao poruka na ekran). Ukoliko je izabran znak !, tada je otkrivena tikva. Igra generira novu ploču koja je identična staroj, samo ne sadrži izabrani element. Npr., za ploču ! - - ! - ! - i indeks 3, nova ploča izgleda ! - - - ! ! -. Ukoliko je izabran znak -, otkriven je busen trave koji sadrži grinje. Igra treba generirati novu ploču na sljedeći način: izabrani element i prvi element s desne strane izabranog elementa (ako postoji) treba izbaciti iz polja. Ukoliko se na desnoj poziciji ponovo nalaze grinje, postupak treba ponoviti. Za identičan primjer ploče i odabrani indeks 1, program treba generirati polje ! - ! ! -. Igra završava kada na ploči više nema tikvi.

- Napišite funkciju

```
char **igraj(char **buseni, int *korak, int *br_tikvi, int **dim, int **index)
```

koja simulira opisanu igru, te sprema sve novodobivene ploče u niz nizova `buseni`, dok u dodatna jednodimenzionalna polja: `dim` - sprema veličinu svake ploče i `index` - sprema odabrane indekse od strane korisnika (parametri funkcije su adrese pokazivača na prvi element tih polja). Pokazivači `korak` i `br_tikvi` sadrže adrese varijabli koje koristite za spremanje informacija o trenutnoj iteraciji igre i o broju preostalih tikvi na ploči. Početna ploča je spremljena u polju koje se nalazi na indeksu 0 u nizu nizova `buseni`, pripadna duljina je spremljena na indeksu 0 u polju `dim`, broj tikvi je spremljen na adresi `br_tikvi`, a početna vrijednost na adresi `korak` je 0.

Funkcija treba igru odigrati do kraja (tj. dok na trenutnoj ploči nema više tikvi). U svakom koraku, funkcija učitava indeks `s` ulaza (ne treba provjeravati postoji li učitani indeks u odgovarajućem polju), a nakon toga se računa nova ploča i odgovarajuća duljina. Funkcija vraća završni niz nizova, nakon dodavanja svih ploča.

- Napišite funkciju `void ispisi(char **buseni, int *dim, int *index, int br_koraka)` koja ispisuje sve korake igre u kojima su izabrane grinje. Funkcija, za svaki takav korak, ispisuje stanje ploče prije i nakon izabira grinja.

Programiranje 2 – prvi kolokvij, 28. 4. 2017.

Zadatak 4. (12 + 3 = 15 bodova) U jednom stringu čuva se niz riječi. Riječi su odvojene znakom '-' koji dolazi **ispred** svake riječi (uključujući prvu riječ u nizu). Pritom, kao riječ smatramo svaki niz znakova koji ne sadrži znak '-'.

- (a) Napišite funkciju `char* obrisi(char *rijeci)` koja prima opisani string, te iz njega izbacuje riječ koja sadrži najveći broj slova. Ukoliko je string duljine nula, potrebno je vratiti vrijednost `NULL`. Funkcija mora alocirati novi string te zapisati izbrisanu riječ u njega. Funkcija vraća pokazivač na taj niz.
- (b) Napišite glavni program u kojem pozivate funkciju `obrisi` sve dok ne ispraznite neki početni string. Nakon svakog poziva funkcije potrebno je ispisati vraćenu riječ te osloboditi zauzetu memoriju.

Primjer: Za dani string `"-aVa0-1sp1ti-r1j3c"`, funkcija će vratiti `"1sp1ti"`, a string će nakon poziva biti `"-aVa0-r1j3c"`.

Napomena: Za rješavanje drugog podzadatka nije nužno da riješite prvi. **Zabranjeno** je korištenje funkcija iz zaglavlja `string.h`, a smijete koristiti funkcije iz zaglavlja `ctype.h`.