

## Programiranje 2 – prvi kolokvij, 13. travnja 2012.

Ime i prezime: \_\_\_\_\_ JMBAG: \_\_\_\_\_

**Upute:** Na kolokviju je dozvoljeno koristiti samo pribor za pisanje i brisanje, te službeni šalabahter. Kalkulatori, mobiteli, razne neslužbene tablice, papiri i sl., nisu dozvoljeni! 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.

**Rezultati i uvid u zadaće:** petak, 20. travnja u 14:00 sati.

### 1. zadatak

(15 bodova) Napišite program koji računa vrijednost funkcije  $f : \mathbb{N}_0 \rightarrow \mathbb{N}_0$ , zadane s

$$f(n) = \begin{cases} 5n + 3, & n < 15, \\ g(f(2n)), & n \text{ neparan i } n > 15, \\ f(g(n+1)), & \text{inače,} \end{cases}$$

gdje je  $g(n)$  broj prostih znamenaka od  $n$  u bazi 12.

Argument funkcije  $f$  prima se preko komandne linije. Ako u komandnoj liniji nije naveden točno jedan argument nakon imena programa, i to prirodan broj (niz decimalnih znamenki), program treba ispisati odgovarajuću poruku o grešci i vratiti neku vrijednost različitu od 0 operacijskom sustavu. U suprotnom, treba ispisati poruku o rezultatu, oblika

$$f(34) = 13.$$

Prirodni brojevi predstavljeni su tipom `unsigned`.

## Programiranje 2 – prvi kolokvij, 13. travnja 2012.

Ime i prezime: \_\_\_\_\_

JMBAG: \_\_\_\_\_

### 2. zadatak

(15 bodova) Neka je  $N \leq 10$  prirodan broj i neka je  $A$  matrica dimenzije  $N \times N$  čiji je svaki element neka znamenka od 1 do 7. Na temelju podataka iz matrice  $A$  gradimo novu matricu  $X$  čiji su elementi znakovi: prebrojimo koliko puta se javlja koja znamenka u matrici  $A$ , te za svaku znamenku formiramo jedan stupac od  $X$ . Prvi element tog stupca je ta znamenka, a ispod nje je onoliko zvjezdica koliko se ona puta javlja. Eventualne nepopunjene elemente u stupcu treba postaviti na znak točka (vidi primjer).

- Napišite funkciju koja prima matricu  $A$ , te vraća matricu  $X$ . Za matricu  $X$  potrebno je alocirati točno onoliko memorije koliko je minimalno potrebno. Funkcija smije primiti i dodatne (potencijalno “varijabilne”) argumente.
- Napišite program koji učitava prirodni broj  $N$  i matricu  $A$ , te poziva funkciju iz podzadatka (a). Program nakon toga treba ispisati dobivenu matricu  $X$ .

Dozvoljeno je korištenje **pomoćnih nizova**, ali **nije** dozvoljeno korištenje **globalnih** varijabli! Ako ne znate napisati funkciju iz (a), za 3 boda manje, proces formiranja matrice  $X$  možete uklopiti u glavni program. Primjer: ako je  $N = 4$  i

$$A = \begin{bmatrix} 3 & 2 & 5 & 2 \\ 4 & 1 & 2 & 5 \\ 7 & 3 & 1 & 3 \\ 1 & 4 & 2 & 5 \end{bmatrix}, \quad \text{onda je} \quad X = \begin{bmatrix} '1' & '2' & '3' & '4' & '5' & '6' & '7' \\ '*' & '*' & '*' & '*' & '*' & '.' & '*' \\ '*' & '*' & '*' & '*' & '*' & '.' & '.' \\ '*' & '*' & '*' & '.' & '*' & '.' & '.' \\ '.' & '*' & '.' & '.' & '.' & '.' & '.' \end{bmatrix}.$$

## Programiranje 2 – prvi kolokvij, 13. travnja 2012.

Ime i prezime: \_\_\_\_\_

JMBAG: \_\_\_\_\_

### 3. zadatak

(15 bodova) Napišite funkciju

```
char* basename(const char* path, char separator, const char* suffix);
```

koja će vratiti string nastao tako da se iz stringa `path` ukloni početni komad sve do (uključivo) posljednjeg pojavljivanja znaka `separator`. Nadalje, ako string `path` završava stringom `suffix`, ukloni se i taj završetak. U slučaju da se unutar stringa `path` ne pojavljuje znak `separator`, s početka stringa se ne uklanja ništa. Ako se završetak stringa `path` ne podudara sa stringom `suffix`, s kraja stringa se ne uklanja ništa.

U slučaju da je putem nekog od argumenata `path` i `suffix` funkciji prosljeđen `NULL` pokazivač, funkcija mora vratiti `NULL`. Također, ako se unutar stringa `suffix` pojavljuje znak `separator`, funkcija treba vratiti `NULL`.

Primjeri:

- `basename("/usr/include/stdio.h", '/', ".h")` vraća `"stdio"`,
- `basename("/usr/bin/gcc", '/', ".exe")` vraća `"gcc"`,
- `basename("libiberty.a", '/', ".a")` vraća `"libiberty"`,
- `basename("Hello world!", 'o', "world!")` vraća `NULL`.

Za string koji vraćate iz funkcije obavezno rezervirajte točno onoliko memorije koliko je potrebno.

**Napomene:** Uočite da **ne smijete** mijenjati stringove koje dobijete kao argumente funkcije. Smijete koristiti funkcije iz `string.h`.

## Programiranje 2 – prvi kolokvij, 13. travnja 2012.

Ime i prezime: \_\_\_\_\_

JMBAG: \_\_\_\_\_

### 4. zadatak

(20 bodova) U memoriji je na neki način pohranjen popis zaposlenih u raznim kompanijama te je zadana funkcija

```
int radnik(int id, int *sef, int *placa, int *n, int podredjeni[]);
```

Zaposlene razlikujemo prema oznaci (pozitivni cijeli broj), pa funkcija, za zadanu oznaku `id`, vraća 1 ako takav zaposlenik postoji, a 0 ako ne postoji. Dodatno, preko varijabilnih argumenata funkcija, redom, vraća oznaku nekog drugog, zaposleniku nadređenog, zaposlenika (šefa), s tim da vraća 0 ako zaposlenik nema nadređenog, plaću zaposlenika, te polje oznaka njegovih podređenih (`n` je duljina tog polja), pri čemu svaki zaposlenik ima najviše 68 direktno podređenih. Svaka kompanija ima točno jednog “glavnog šefa”, tj. zaposlenika koji nema nadređenih.

Koristeći navedenu funkciju `radnik`, napišite:

- (a) [5 bodova] funkciju `int glavni_sef(int id)` koja vraća oznaku “glavnog šefa” kompanije u kojoj radi zaposleni sa zadanom oznakom. Ako zadani zaposlenik ne postoji, funkcija vraća 0;
- (b) [15 bodova] funkciju `svi_sefovi`, koja kroz argument (uz koji smiju postojati i drugi argumenti) prima oznaku nekog zaposlenika te ispisuje oznake i plaće svih šefova (zaposlenika koji imaju barem jednog podređenog) u kompaniji u kojoj radi zadani zaposlenik (tj. imaju istog glavnog šefa).

Obavezno navedite i primjer poziva funkcije `svi_sefovi`.

## Programiranje 2 – prvi kolokvij, 13. travnja 2012.

Ime i prezime: \_\_\_\_\_ JMBAG: \_\_\_\_\_

**Upute:** Na kolokviju je dozvoljeno koristiti samo pribor za pisanje i brisanje, te službeni šalabahter. Kalkulatori, mobiteli, razne neslužbene tablice, papiri i sl., nisu dozvoljeni! 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.

**Rezultati i uvid u zadaće:** petak, 20. travnja u 14:00 sati.

### 1. zadatak

(15 bodova) Napišite program koji računa vrijednost funkcije  $f : \mathbb{N}_0 \rightarrow \mathbb{N}_0$ , zadane s

$$f(n) = \begin{cases} n^3 + 3, & n < 13, \\ g(f(4n)), & n \text{ neparan i } n > 13, \\ f(g(n + 32)), & \text{inače,} \end{cases}$$

gdje je  $g(n)$  zbroj složenih znamenaka od  $n$  u bazi 13.

Argument funkcije  $f$  prima se preko komandne linije. Ako u komandnoj liniji nije naveden točno jedan argument nakon imena programa, i to prirodan broj (niz decimalnih znamenki), program treba ispisati odgovarajuću poruku o grešci i vratiti neku vrijednost različitu od nule operacijskom sustavu. U suprotnom, treba ispisati poruku o rezultatu, oblika

$$f(53) = 12.$$

Prirodni brojevi predstavljeni su tipom `unsigned`.

## Programiranje 2 – prvi kolokvij, 13. travnja 2012.

Ime i prezime: \_\_\_\_\_

JMBAG: \_\_\_\_\_

### 2. zadatak

(15 bodova) Neka je  $N \leq 20$  prirodan broj i neka je  $B$  matrica dimenzije  $N \times N$  čiji je svaki element neka znamenka od 5 do 9. Na temelju podataka iz matrice  $B$  gradimo novu matricu  $Y$  čiji su elementi znakovi: prebrojimo koliko puta se javlja koja znamenka u matrici  $B$ , te za svaku znamenku formiramo jedan redak od  $Y$ . Zadnji element tog retka je ta znamenka, a lijevo od nje je onoliko znakova  $x$  koliko se ona puta javlja. Eventualne nepopunjene elemente u retku treba postaviti na znak minus (vidi primjer).

- Napišite funkciju koja prima matricu  $B$ , te vraća matricu  $Y$ . Za matricu  $Y$  potrebno je alocirati točno onoliko memorije koliko je minimalno potrebno. Funkcija smije primiti i dodatne (potencijalno “varijabilne”) argumente.
- Napišite program koji učitava prirodni broj  $N$  i matricu  $B$ , te poziva funkciju iz podzadatka (a). Program nakon toga treba ispisati dobivenu matricu  $Y$ .

Dozvoljeno je korištenje **pomoćnih nizova**, ali **nije** dozvoljeno korištenje **globalnih** varijabli! Ako ne znate napisati funkciju iz (a), za 3 boda manje, proces formiranja matrice  $Y$  možete uklopiti u glavni program. Primjer: ako je  $N = 4$  i

$$B = \begin{bmatrix} 8 & 5 & 9 & 8 \\ 5 & 7 & 8 & 5 \\ 8 & 8 & 9 & 8 \\ 7 & 9 & 7 & 5 \end{bmatrix}, \quad \text{onda je } Y = \begin{bmatrix} '- & '- & 'x' & 'x' & 'x' & 'x' & '5' \\ '- & '- & '- & '- & '- & '- & '6' \\ '- & '- & '- & 'x' & 'x' & 'x' & '7' \\ 'x' & 'x' & 'x' & 'x' & 'x' & 'x' & '8' \\ '- & '- & '- & 'x' & 'x' & 'x' & '9' \end{bmatrix}.$$

## Programiranje 2 – prvi kolokvij, 13. travnja 2012.

Ime i prezime: \_\_\_\_\_

JMBAG: \_\_\_\_\_

### 3. zadatak

(15 bodova) Napišite funkciju

```
char* detox_and_trim(const char* str, const char* ws, const char* repl);
```

koja će vratiti string nastao tako da se iz stringa `str` s početka i kraja uklone znakovi koji se nalaze unutar stringa `ws`, te se u ostatku stringa znakovi iz `ws` zamijene znakovima iz `repl`, tako da se `ws[i]` zamjenjuje s `repl[i]`.

U slučaju da je putem nekog od argumenata funkciji prosljeđen `NULL` pokazivač, funkcija mora vratiti `NULL`. Također, ako su stringovi `ws` i `repl` različitih duljina, ili ako se neki znak iz stringa `ws` pojavljuje unutar stringa `repl`, funkcija treba vratiti `NULL`.

Primjeri:

- `detox_and_trim("132A1B2C22", "123", "!@#")` vraća "A!B@C";
- `detox_and_trim("132A1B2C22", "123", "ab")` vraća `NULL`;
- `detox_and_trim("132A1B2C22", "123", "ab1")` vraća `NULL`.

Za string koji vraćate iz funkcije obavezno rezervirajte točno onoliko memorije koliko je potrebno.

**Napomene:** Uočite da **ne smijete** mijenjati stringove koje dobijete kao argumente funkcije. Smijete koristiti funkcije iz `string.h`.

## Programiranje 2 – prvi kolokvij, 13. travnja 2012.

Ime i prezime: \_\_\_\_\_

JMBAG: \_\_\_\_\_

### 4. zadatak

(20 bodova) Održana su natjecanja u kojima su se natjecale grupe natjecatelja (svaki u točno jednom natjecanju) na način da iz svake grupe jedan pobjednik "ide dalje", te se bori s drugim pobjednicima. Grupe ne moraju nužno biti jednake veličine, a svako natjecanje je završilo kad je ostao jedan pobjednik. Podaci o natjecanjima su na neki način pohranjeni u memoriji računala te je zadana funkcija

```
int natjecatelj(int id, int *n, int pobijedjeni[], int *pobjednik);
```

Natjecatelje razlikujemo prema oznaci (pozitivni cijeli broj), pa funkcija, za zadanu oznaku `id`, vraća 1 ako takav natjecatelj postoji, a 0 ako ne postoji. Dodatno, preko varijabilnih argumenata funkcija, redom, vraća niz oznaka natjecatelja koje je natjecatelj pobijedio (`n` je duljina tog polja), pri čemu znamo da niti jedan natjecatelj nije (direktno) pobijedio više od 51 protivnika, i oznaku natjecatelja koji je pobijedio zadanog natjecatelja, s tim da vraća 0 ako `id` označava upravo konačnog pobjednika.

Koristeći navedenu funkciju `natjecatelj`, napišite:

- (a) [5 bodova] funkciju `int pobjednik(int id)` koja vraća oznaku konačnog pobjednika natjecanja u kojem je sudjelovao natjecatelj s oznakom `id`. Ako zadani natjecatelj ne postoji, funkcija vraća 0;
- (b) [15 bodova] funkciju `prvoporazeni`, koja kroz argument (uz koji smiju postojati i drugi argumenti) prima oznaku nekog natjecatelja te ispisuje oznake svih natjecatelja koji su ispali u prvom krugu (nisu nikoga pobijedili), a sudjelovali su u istom natjecanju kao i natjecatelj sa zadanom oznakom (tj. imaju istog konačnog pobjednika).

Obavezno navedite i primjer poziva funkcije `prvoporazeni`.



## Programiranje 2 – prvi kolokvij, 13. travnja 2012.

Ime i prezime: \_\_\_\_\_ JMBAG: \_\_\_\_\_

**Upute:** Na kolokviju je dozvoljeno koristiti samo pribor za pisanje i brisanje, te službeni šalabahter. Kalkulatori, mobiteli, razne neslužbene tablice, papiri i sl., nisu dozvoljeni! 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.

**Rezultati i uvid u zadaće:** petak, 20. travnja u 14:00 sati.

### 1. zadatak

(15 bodova) Napišite program koji računa vrijednost funkcije  $f : \mathbb{N}_0 \rightarrow \mathbb{N}_0$ , zadane s

$$f(n) = \begin{cases} (2n + 1)^3, & n \text{ jednoznamenkast,} \\ g(f(5n)), & \text{vodeće dvije znamenke od } n \text{ su jednake,} \\ f(g(n + 1)), & \text{vodeće dvije znamenke od } n \text{ su različite,} \end{cases}$$

gdje je  $g(n) = \lfloor \sqrt[3]{n} \rfloor$ .

Argument funkcije  $f$  prima se preko komandne linije. Ako u komandnoj liniji nije naveden točno jedan argument nakon imena programa, i to prirodan broj (niz decimalnih znamenki), program treba ispisati odgovarajuću poruku o grešci i vratiti neku vrijednost različitu od nule operacijskom sustavu. U suprotnom, treba ispisati poruku o rezultatu, oblika

$$f(117) = 17.$$

Prirodni brojevi predstavljeni su tipom `unsigned`.

## Programiranje 2 – prvi kolokvij, 13. travnja 2012.

Ime i prezime: \_\_\_\_\_

JMBAG: \_\_\_\_\_

### 2. zadatak

(15 bodova) Neka je  $N \leq 30$  prirodan broj i neka je  $C$  matrica dimenzije  $N \times N$  čiji je svaki element neko od slova A, B, C, D, E, F. Na temelju podataka iz matrice  $C$  gradimo novu matricu  $Z$  čiji su elementi znakovi: prebrojimo koliko puta se javlja koje slovo u matrici  $C$ , te za svako slovo formiramo jedan stupac od  $Z$ . Zadnji element tog stupca je to slovo, a iznad njega je onoliko povisilica koliko se ono puta javlja. Eventualne nepopunjene elemente u stupcu treba postaviti na slovo o (vidi primjer).

- Napišite funkciju koja prima matricu  $C$ , te vraća matricu  $Z$ . Za matricu  $Z$  potrebno je alocirati točno onoliko memorije koliko je minimalno potrebno. Funkcija smije primiti i dodatne (potencijalno “varijabilne”) argumente.
- Napišite program koji učitava prirodni broj  $N$  i matricu  $C$ , te poziva funkciju iz podzadatka (a). Program nakon toga treba ispisati dobivenu matricu  $Z$ .

Dozvoljeno je korištenje **pomoćnih nizova**, ali **nije** dozvoljeno korištenje **globalnih** varijabli! Ako ne znate napisati funkciju iz (a), za 3 boda manje, proces formiranja matrice  $Z$  možete uklopiti u glavni program. Primjer: ako je  $N = 4$  i

$$C = \begin{bmatrix} F & C & F & D \\ A & E & D & A \\ C & A & F & D \\ D & A & C & F \end{bmatrix}, \quad \text{onda je} \quad Z = \begin{bmatrix} \# & o & o & \# & o & \# \\ \# & o & \# & \# & o & \# \\ \# & o & \# & \# & o & \# \\ \# & o & \# & \# & \# & \# \\ A & B & C & D & E & F \end{bmatrix}.$$

## Programiranje 2 – prvi kolokvij, 13. travnja 2012.

Ime i prezime: \_\_\_\_\_

JMBAG: \_\_\_\_\_

### 3. zadatak

(15 bodova) Napišite funkciju

```
char* cut(const char* str, int f, const char* delim);
```

koja promatra string `str` kao niz "riječi" razdvojenih znakovima navedenim u stringu `delim`. Ako se unutar stringa `str` jedan do drugog nalaze dva znaka iz stringa `delim`, smatramo da oni omeđuju praznu riječ. Ako se na početku, odnosno, na kraju stringa `str` nalazi znak iz niza `delim`, smatramo da string `str` započinje, odnosno, završava praznom riječi. Funkcija treba vratiti `f`-tu riječ iz stringa `str`, pri čemu riječi brojimo od 1, a prazne riječi se broje kao riječi.

U slučaju da je putem nekog od argumenata `str` i `delim` funkciji proslijeđen `NULL` pokazivač, funkcija mora vratiti `NULL`. Također, ako tražena `f`-ta riječ ne postoji u stringu, funkcija mora vratiti `NULL`.

Primjeri:

- `cut("JMBAG;Ime,Prezime,Ocjena", 3, ",;")` vraća "Prezime";
- `cut(",A,,B,C,D", 4, ",")` vraća "B";
- `cut("A,B,C,D", 17, ",")` vraća `NULL`.

Za string koji vraćate iz funkcije obavezno rezervirajte točno onoliko memorije koliko je potrebno.

**Napomene:** Uočite da **ne smijete** mijenjati stringove koje dobijete kao argumente funkcije. Smijete koristiti funkcije iz `string.h`.

## Programiranje 2 – prvi kolokvij, 13. travnja 2012.

Ime i prezime: \_\_\_\_\_

JMBAG: \_\_\_\_\_

### 4. zadatak

(20 bodova) Baza podataka sadržava popis doktora matematike, raspoređenih prema mentorima. Doktorande razlikujemo prema oznakama (pozitivni cijeli broj), a svaki od njih ima najviše jednog poznatog mentora. Njihovim podacima pristupamo pomoću funkcije

```
int drsc(int id, int *mentor, int studenti[], int *n);
```

koja, za zadanog doktora matematike s oznakom `id`, vraća 1 ako takav postoji, a 0 ako ne postoji. Dodatno, preko varijabilnih argumenata funkcija, redom, vraća oznaku mentora (za doktore kojima je mentor nepoznat vrijednost argumenta `mentor` postavlja na nulu), te polje oznaka doktorskih studenata (`n` je duljina tog polja) kojima je zadani doktor bio mentor. Poznato je da niti jedan doktor matematike nije imao više od 85 doktorskih studenata.

Koristeći navedenu funkciju `drsc`, napišite:

- (a) [5 bodova] funkciju `int dr_adam(int id)` koja vraća oznaku “prvog mentorskog pretka” doktora matematike s oznakom `id` (gledamo mentora, pa njegovog mentora, itd. dok ne dođemo do osobe čiji mentor nije poznat). Ako zadani doktor matematike ne postoji, funkcija vraća 0;
- (b) [15 bodova] funkciju `max`, koja kroz argument (uz koji smiju postojati i drugi argumenti) prima oznaku nekog doktora matematike te vraća 0 ako takav ne postoji, odnosno, broj koliko je najviše doktorskih studenata imao neki od “doktorskih rođaka” doktora matematike s oznakom `id`. “Doktorski rođaci” su oni matematičari koji imaju zajedničkog “prvog mentorskog pretka”.

Obavezno navedite i primjer poziva funkcije `max`.

## Programiranje 2 – prvi kolokvij, 13. travnja 2012.

Ime i prezime: \_\_\_\_\_ JMBAG: \_\_\_\_\_

**Upute:** Na kolokviju je dozvoljeno koristiti samo pribor za pisanje i brisanje, te službeni šalabahter. Kalkulatori, mobiteli, razne neslužbene tablice, papiri i sl., nisu dozvoljeni! 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.

**Rezultati i uvid u zadaće:** petak, 20. travnja u 14:00 sati.

### 1. zadatak

(15 bodova) Napišite program koji računa vrijednost funkcije  $f : \mathbb{N}_0 \rightarrow \mathbb{N}_0$ , zadane s

$$f(n) = \begin{cases} n(2n + 1), & n < 5, \\ g(f(n + 1)), & \text{treći bit slijeva u } n \text{ je jednak } 1, \\ f(g(2n)), & \text{inače,} \end{cases}$$

gdje je  $g(n) = \lfloor \log_2(n + 1) \rfloor$ .

Argument funkcije  $f$  prima se preko komandne linije. Ako u komandnoj liniji nije naveden točno jedan argument nakon imena programa, i to prirodan broj (niz decimalnih znamenki), program treba ispisati odgovarajuću poruku o grešci i vratiti neku vrijednost različitu od nule operacijskom sustavu. U suprotnom, treba ispisati poruku o rezultatu, oblika

$$f(11) = 5.$$

Prirodni brojevi predstavljeni su tipom `unsigned`.

## Programiranje 2 – prvi kolokvij, 13. travnja 2012.

Ime i prezime: \_\_\_\_\_

JMBAG: \_\_\_\_\_

### 2. zadatak

(15 bodova) Neka je  $N \leq 40$  prirodan broj i neka je  $D$  matrica dimenzije  $N \times N$  čiji je svaki element neko od slova a, b, c, d, e, f. Na temelju podataka iz matrice  $D$  gradimo novu matricu  $W$  čiji su elementi znakovi: prebrojimo koliko puta se javlja koje slovo u matrici  $D$ , te za svako slovo formiramo jedan redak od  $W$ . Prvi element tog retka je to slovo, a desno od njega je onoliko znakova > koliko se ono puta javlja. Eventualne nepopunjene elemente u retku treba postaviti na znak : (vidi primjer).

- Napišite funkciju koja prima matricu  $D$ , te vraća matricu  $W$ . Za matricu  $W$  potrebno je alocirati točno onoliko memorije koliko je minimalno potrebno. Funkcija smije primiti i dodatne (potencijalno “varijabilne”) argumente.
- Napišite program koji učitava prirodni broj  $N$  i matricu  $D$ , te poziva funkciju iz podzadatka (a). Program nakon toga treba ispisati dobivenu matricu  $W$ .

Dozvoljeno je korištenje **pomoćnih nizova**, ali **nije** dozvoljeno korištenje **globalnih** varijabli! Ako ne znate napisati funkciju iz (a), za 3 boda manje, proces formiranja matrice  $W$  možete uklopiti u glavni program. Primjer: ako je  $N = 4$  i

$$D = \begin{bmatrix} f & b & e & e \\ b & f & a & b \\ b & a & f & d \\ f & d & b & d \end{bmatrix}, \quad \text{onda je} \quad W = \begin{bmatrix} a & > & > & : & : & : \\ b & > & > & > & > & > \\ c & : & : & : & : & : \\ d & > & > & > & : & : \\ e & > & > & : & : & : \\ f & > & > & > & > & : \end{bmatrix}.$$

## Programiranje 2 – prvi kolokvij, 13. travnja 2012.

Ime i prezime: \_\_\_\_\_

JMBAG: \_\_\_\_\_

### 3. zadatak

(15 bodova) Opcija je niz znakova koji počinje nizom znakova "--", nakon čega slijedi jedan ili više znakova engleskog abecede (samo slova). Opcije mogu imati i argumente. Opcija ima argument ako neposredno nakon opcije slijedi znak '='. Niz znakova engleske abecede koji slijedi neposredno nakon znaka '=' nazivamo argument.

Napišite funkciju

```
char* nextopt(const char* str);
```

koja će vratiti string koji u sebi sadržava prvu opciju, zajedno s eventualnim argumentom, koja se kao podstring nalazi u stringu `str`. Ako unutar stringa `str` ne postoji opcija ili je `str` zadan kao NULL pokazivač, funkcija treba vratiti NULL.

Primjeri:

- `nextopt("ls --color=auto --directory")` vraća `--color=auto`;
- `nextopt("blabla--opcija=argument=bla")` vraća `--opcija=argument`;
- `nextopt("ls --directory --color=auto")` vraća `--directory`;
- `nextopt("abc --123")` vraća NULL.

Za string koji vraćate iz funkcije obavezno rezervirajte točno onoliko memorije koliko je potrebno.

**Napomene:** Uočite da **ne smijete** mijenjati stringove koje dobijete kao argumente funkcije. Smijete koristiti funkcije iz `string.h`.

## Programiranje 2 – prvi kolokvij, 13. travnja 2012.

Ime i prezime: \_\_\_\_\_

JMBAG: \_\_\_\_\_

### 4. zadatak

(20 bodova) Naš stari znanac, zli znanstvenik dr. Kemije Epruvetić, nakon uspješne karijere virusologa, počeo se baviti toksičnim spojevima. Kombinirajući kemikalije i jednostavnije spojeve, gradi sve složenije spojeve, s tim da svaki sastojak (kemikaliju ili kreirani spoj) koristi kao dio u samo jednom novom spoju. Sastojke koje još nije iskoristio niti u jednom spoju, dr. Epruvetić naziva supertox-sastojci.

Sve korištene sastojke na neki je način pohranio u memoriji računala, a podatke o njima dobija preko funkcije

```
int kemije(int id, double *tox, int *novi, int *n, int sastojci[]);
```

koja, za zadani sastojak s oznakom `id` (pozitivni cijeli broj), vraća 1 ako takav postoji, a 0 ako ne postoji. Dodatno, preko varijabilnih argumenata funkcija, redom, vraća toksičnost sastojka, oznaku onog spoja kojem je jedan od sastojaka ovaj s oznakom `id` (s tim da argument `novi` poprima vrijednost 0 za supertox-sastojke), te polje oznaka sastojaka od koji je nasato zadani spoj (`n` je duljina tog polja).

Koristeći navedenu funkciju `kemije`, napišite:

- (a) [5 bodova] funkciju `int supertox(int id)` koja vraća oznaku supertox-sastojka čiji je jedan od sastojaka (ne nužno direktni!) onaj s oznakom `id`. Ako zadani sastojak ne postoji, funkcija vraća 0;
- (b) [15 bodova] funkciju `fail`, koja kroz argument (uz koji smiju postojati i drugi argumenti) prima oznaku nekog sastojka te ispisuje oznake neuspješnih spojeva (onih čija je toksičnost manja od toksičnosti nekog njihovog sastojka), a koji su indirektno korišteni zajedno sa zadanim sastojkom (tj. imaju isti supertox-sastojak).

Obavezno navedite i primjer poziva funkcije `fail`.