

Zadaća 3.

Ova zadaća nosi ukupno 4 poena, pri čemu prva dva zadatka nose po 1 poen, dok treći, četvrti i peti zadatak nose respektivno 0.7, 0.9 i 0.4 poena. Svi zadaci se mogu uraditi na osnovu gradiva sa prvih osam predavanja i pretpostavljenog predznanja iz predmeta "Osnove računarstva". Rok za predaju ove zadaće je petak 9. V 2014. (do kraja dana) i ne može se produžiti. Zadaće se predaju putem Zamgera.

1. U većini programa za obradu teksta postoji opcija za kreiranje *indeksa pojmova*, koji predstavlja popis pojmova koji se nalaze u tekstu, pri čemu se uz svaki pojam navode i pozicije (obično brojevi stranica) na kojima se taj pojam javlja u tekstu. U ovom zadatku ćete trebati uraditi nešto slično tome. Prvo ćete napraviti funkciju "KreirajIndeksPojmova" koja kao parametar prima neki string (tipa "string") koji predstavlja tekst koji se analizira. Funkcija kao rezultat treba vratiti mapu koja predstavlja traženi indeks pojmova. Ključna polja ove mape su tipa "string", a pridružene vrijednosti su skupovi cijelih brojeva. Vrijednosti ključnih polja predstavljaju različite riječi pronađene u analiziranom tekstu, dok su odgovarajuće pridružene vrijednosti skupovi čiji su elementi pozicije (tj. indeksi unutar razmatranog stringa) na kojima se odgovarajuća riječ nalazi unutar razmatranog stringa. Na primjer, ukoliko se kao parametar funkciji ponudi string "abc qwe stsda abc abc dhi qwe hrkw dhi", funkcija treba da vrati mapu u kojoj se nalazi 5 parova (toliko ukupno ima različitih riječi u stringu), čija su ključna polja stringovi "abc", "dhi", "hrkw", "qwe" i "stsda" (to su riječi koje se nalaze u stringu), dok su odgovarajuća ključna polja skupovi {0, 14, 18}, {22, 35}, {30}, {4, 26} i {8} (riječ "abc" se nalazi na pozicijama 0, 14 i 18 u razmatranom tekstu, riječ "dhi" na pozicijama 22 i 35, itd.). Sljedeća funkcija koju treba napraviti je "PretražiIndeksPojmova". Ova funkcija kao parametar prima neku riječ (tipa "string") i mapu koja predstavlja indeks pojmova, a koja kao rezultat vraća odgovarajući skup pozicija za datu riječ pronađen u datom indeksu pojmova, ili baca izuzetak tipa "logic_error" sa prpratnim tekstom "Pojam nije nađen!" u slučaju da data riječ nije nađena u indeksu pojmova. Konačno, posljednja funkcija koju treba napraviti je "IspisiIndeksPojmova". Ova funkcija kao parametar prima mapu koja predstavlja indeks pojmova, a ispisuje njen kompletan sadržaj na ekranu u obliku tako da se u svakom redu ispisuje prvo pojam, zatim dvotačka praćena razmakom, i na kraju, spisak pozicija međusobno razdvojenih zarezom (bez ikakvih razmaka). Na primjer, za indeks pojmova kreiran na osnovu stringa iz prethodnog primjera, ova funkcija bi trebala proizvesti ispis poput sljedećeg:

```
abc: 0,14,18
dhi: 22,35
hrkw: 30
qwe: 4,26
stsda: 8
```

Radi jednostavnosti, pretpostavite da string koji se analizira sadrži samo mala slova i ništa drugo (dakle, bez znakova interpunkcije), ali riječi mogu biti međusobno razdvojene sa više razmaka (također, tekst može počinjati i završavati razmacima).

Napisane funkcije demonstrirajte u glavnom programu u kojem se prvo sa tastature unosi tekst za analizu, nakon čega se na ekranu ispisuje kreirani indeks pojmova. Potom program ulazi u petlju u kojoj se za svaku riječ unesenu sa tastature ispisuje indeks pozicija na kojima se riječ nalazi u analiziranom tekstu, pri čemu su pozicije međusobno razdvojene razmacima (koristeći pri tome kreirani indeks pojmova), ili informaciju da unesena riječ nije nađena. Program treba da prekine rad kada korisnik unese znak "." (tačka sa tastature).

Napomena: Ukoliko efektivno iskoristite ono što tip mapa podržava, sve tražene funkcije mogu se napisati vrlo jednostavno.

2. Pod matricnim eksponencijalnim polinomom n -tog reda neke matrice A podrazumijeva se matrica definirana izrazom

$$\exp_n A = I + A + \frac{1}{2!} A^2 + \frac{1}{3!} A^3 + \dots + \frac{1}{n!} A^n = \sum_{k=0}^n \frac{1}{k!} A^k$$

gdje je I jedinična matrica istog formata kao i matrica A , dok je A^k klasični k -ti stepen matrice, tj. produkt matrice A k -puta sa samom sobom (razumije se da zbog toga A mora biti kvadratna matrica). Matricni eksponencijalni polinomi imaju veliku primjenu u raznim oblastima nauke i tehnike, a posebno u automatici (specijalno, za $n \rightarrow \infty$, izraz $\exp_n A$ konvergira ka tzv. matricnoj eksponencijalnoj funkciji $\exp A$, koja se obilježava i sa e^A). Vaš zadatak je da dopunite program za rad sa generičkom strukturom “Matrica” obrađen na Predavanju 8_b sa dvije nove funkcije “ProduktMatrica” i “MatricniEksponencijalniPolinom”. Funkcija “ProduktMatrica” prima dvije matrice kao parametre (matrice su definirane kao odgovarajuće generičke strukture tipa “Matrica”) i vraća njihov produkt kao rezultat, odnosno baca izuzetak tipa “domain_error” ukoliko matrice nisu saglasne za množenje. Funkcija “MatricniEksponencijalniPolinom” prima matricu A kao prvi parametar, a kao drugi parametar prirodan broj n , dok kao rezultat vraća vrijednost izraza $\exp_n A$ (za potrebe realizacije ove funkcije možete i trebate koristiti već napisanu funkciju “ProduktMatrica”). U slučaju da matrica nije kvadratna, funkcija treba baciti izuzetak, s obzirom da je izraz $\exp_n A$ definiran samo za kvadratne matrice. Izuzetak (također tipa tipa “domain_error”) treba baciti i ukoliko je n negativan.

Pored ove dvije funkcije, treba također proširiti funkciju “IspisiMatricu” sa dodatna dva parametra nazvana “preciznost” i “treba_brisati”. Parametar “preciznost” je cijeli broj koji određuje preciznost ispisa, odnosno broj tačnih cifara pri ispisu (njega zapravo treba proslijediti funkciji “cout.precision” odnosno manipulatoru “setprecision”. Ovaj parametar treba da ima podrazumijevanu vrijednost 6. Drugi parametar “treba_brisati” je tipa “bool”. Ukoliko ovaj parametar ima vrijednost “true”, funkcija treba da oslobodi prostor zauzet matricom koja joj je prosljeđena kao parametar, u suprotnom ne treba da radi ništa. Ovim se omogućava da možemo zadavati pozive poput

```
IspisiMatricu(ZbirMatrica(a, b), 10, 5, true);
```

tako da se oslobađanje memorije koju je zauzela pomoćna matrica koja predstavlja zbir matrica može obaviti bez korištenja pomoćne promjenljive (kao što smo morali u izvornom programu prikazanom na predavanjima). Pri tome, definirajte da parametar “treba_brisati” ima podrazumijevanu vrijednost “false”, tako da ga ne treba navoditi ukoliko nam brisanje ne treba.

Napisane funkcije testirajte u glavnom programu na primjeru matrica čije dimenzije i elemente unosi korisnik putem tastature. U glavnom programu predvidite hvatanje svih izuzetaka koji bi eventualno mogli nastupiti. Također, dobro pazite da nigdje ne dođe do curenja memorije, ni pod kakvim okolnostima (u suštini, to je i *osnovni cilj zadatka*). Kao primjer testa, ukoliko je matrica A formata 2×2 sa elementima $A_{1,1} = 1$, $A_{1,2} = 2$, $A_{2,1} = 3$ i $A_{2,2} = 4$ i ako je $n = 10$, ispis uz preciznost ispisa od 6 tačnih cifara trebao bi izgledati ovako:

```
50.8477    73.1025
109.654    160.501
```

Napomena 1: Kako faktorijel vrlo brzo raste sa porastom n , da biste mogli dobijati ispravne rezultate za iole veće vrijednosti n , morate predvidjeti da tip faktorijela bude “double”.

Napomena 2: U ovom programu je *izuzetno lako* napraviti curenje memorije (tačnije, u funkciji “MatricniEksponencijalniPolinom”). Dobro razmislite šta radite, i ne oslanjate se nasumice ni na kakve “testere curenja memorije” i slična pomagala, jer u suprotnom nećete ovladati tehnikama upravljanja memorijom (to i jeste glavni cilj ovog zadatka). Kad shvatite koliko treba razumijevanja da se u ovom zadatku izbjegne curenje memorije, tek tada ćete *zaista cijeniti destruktore*, koji ovakve probleme rješavaju praktično automatski.

Napomena 3: Da, dobro ste primijetili, ovaj zadatak je *već bio za zadaću u prethodnoj generaciji*. Naravno, možete ga “pođoniti”, ili Vam ga može uraditi neko drugi (kao što uostalom nekima od Vas drugi rade i ostale zadatke). Ali da znate, što bi rekao Balašević, *neko to od gore vidi sve*, i prije ili kasnije to će Vam se od glavu razbiti. Izuzetno je važno da barem ovaj zadatak *zaista uradite sami* (naravno, samostalno iste trebali uraditi i sve ostale zadatke, ali je izuzetno važno da baš ovaj zadatak ne prepisete ni .po koju cijenu). Ukoliko ne želite samostalno da uradite ovaj zadatak, *radije ga nemojte ni predavati*.

3. U plemenu Wabambe poglavica se bira razbrajalicom. Svi punoljetni članovi plemena osim bivšeg poglavice poredaju se u krug, a bivši poglavica saopštava plemenu neki prirodni broj M . Nakon toga, počinje razbrajanje. Razbrajanje se vrši tako što se svaki M -ti član plemena po redu odstranjuje iz kruga, dok u krugu ne ostane samo jedan čovjek. Taj čovjek će postati novi poglavica. Na primjer, ukoliko u krugu ima 11 ljudi (sa rednim brojevima 1 – 11) i ukoliko je $M=4$, redosljed ispadanja je 4, 8, 1, 6, 11, 7, 3, 2, 5 i 10 (nacrtajte sliku), tako da na kraju ostaje samo čovjek sa rednim brojem 9, koji će postati novi poglavica.

Mandat poglavice traje tačno godinu dana. Za vrijeme trajanja mandata poglavica ima apsolutnu moć u plemenu i uživa sve privilegije koje se mogu zamisliti. Po isteku mandata, poglavicu njegovi saplemenici skuhanju i pojedu. Svi članovi plemena doživljavaju izbor za poglavicu kao veliku čast. Međutim, Mbebe Mgogo, koji upravo ove godine postaje punoljetan, nikako ne želi postati poglavica. Ideja o neograničenim privilegijama zapravo i ne zvuči loše, ali mu se ideja da bude skuhan i pojeden nakon isteka mandata baš i ne sviđa osobito. Zbog toga, Mbebe Mgogo po svaku cijenu želi da izbjegne da bude izabran za poglavicu. On je uspio podmititi postojećeg poglavicu da mu oda koji će broj M biti saopšten. Međutim, Mbebe ne zna koliki je tačan broj punoljetnih osoba u plemenu, ali zna da taj broj nije manji od N_1 niti veći od N_2 gdje su N_1 i N_2 neki prirodni brojevi.

Vaš zadatak je da napravite program koji će Mbebeu pomoći da ne postane poglavica. U programu treba implementirati dvije funkcije, “Poglavica” i “SigurnoMjesto”. Funkcija “Poglavica” prima kao parametre broj punoljetnih članova plemena N i korak razbrajanja M i kao rezultat daje redni broj čovjeka koji postaje poglavica. Pri tome, funkcija treba biti zasnovana na tipu “list”, koji se često primjenjuje upravo za rješavanje problema srodnih opisanom problemu. Naime, funkcija “Poglavica” će prvo napuniti listu rednim brojevima od 1 do N . Nakon što je formirana lista, vršimo kretanje kroz listu, pri čemu se nakon svakih M napravljenih koraka iz liste odstranjuje onaj element na kojem se trenutno nalazimo (čime se odgovarajuća osoba efektivno odstranjuje iz kruga), nakon čega prelazimo na sljedeću listu. Pri tome, kad god dostignemo kraj liste, vraćamo se ponovo na početak, čime simuliramo “kružnu” listu. Postupak se ponavlja dok u listi ne ostane samo jedan broj, koji upravo predstavlja redni broj novog poglavice (i njega treba vratiti kao rezultat iz funkcije). Funkcija “Poglavica” treba da ima i treći parametar logičkog tipa, koji služi za lakše testiranje. Ukoliko je taj parametar “**true**”, funkcija vrši tokom rada ispis na ekran rednog broja svake odstranjene osobe (u redosljedu odstranjivanja), dok se ukoliko je taj parametar “**false**”, nikakav ispis ne vrši (“**true**” se zadaje samo za potrebe testiranja). Taj parametar treba da ima podrazumijevanu vrijednost “**false**”, tako da se ne mora zadavati (naravno, u tom slučaju se podrazumijeva rad bez ispisa).

Već je rečeno da program pored funkcije “Poglavica”, treba implementirati i funkciju “SigurnoMjesto”. Ova funkcija kao parametre prima M , N_1 i N_2 a kao rezultat daje redni broj sigurnog mjesta u krugu, tj. mjesta na koje treba stati da se sigurno ne bi postalo poglavica za ma kakav N u opsegu od N_1 do N_2 (ukoliko takvih mjesta ima više, funkcija vraća redni broj onog sigurnog mjesta sa najmanjim rednim brojem). Ukoliko slučajno sigurno mjesto za zadane vrijednosti M , N_1 i N_2 ne postoji, funkcija kao rezultat vraća 0. Rad ove funkcije zasniva se na pozivanju funkcije “Poglavica” za sve vrijednosti N u opsegu od N_1 do N_2 i bilježenja ko će biti izabran za poglavicu u svim tim slučajevima (mjesto je sigurno ukoliko čovjek koji stoji na tom mjestu neće biti poglavica ni za kakvo N u opsegu od N_1 do N_2). Napisane funkcije demonstrirajte u testnom programu koji za zadane vrijednosti M , N_1 i N_2 ispisuje na koju poziciju Mbebe Mgogo treba da stane da garantirano neće postati poglavica (ili ispisuje informaciju da takva garancija ne postoji za zadane ulazne podatke).

4. Riješite ponovo prethodni zadatak, ali tako što ćete za realizaciju funkcije “Poglavica” umjesto bibliotečki definiranog tipa podataka “list” koristiti ručno kreiranu povezanu listu čvorova (dakle, bez upotrebe ikakvih bibliotečki definiranih tipova). To ćete izvesti ovako. Prvo ćete definirati čvornu strukturu “Clan” koja predstavlja jedan element liste, odnosno jednog člana plemena. Ona treba sadržavati polje “redni_broj” tipa “int” i polje “sljedeci” koje je po tipu pokazivač na strukturu tipa “Clan”. Polje “redni_broj” sadržavaće redni broj člana plemena u krugu, dok će polje “sljedeci” pokazivati na sljedeću osobu u krugu. Funkcija “Poglavica” treba kreirati povezanu listu ljudi (tj. čvorova tipa “Clan”) čiji će redni brojevi biti postavljeni redom na vrijednosti od 1 do N, pri čemu će svaki član pokazivati na člana plemena sa narednim rednim brojem, osim posljednje osobe o krugu koja će pokazivati ponovo na prvu osobu, čime se zapravo kreira krug članova plemena (takve povezane liste u kojima posljednji čvor u listi pokazuje nazad na prvi čvor nazivaju se *kružne* ili *cirkularne liste*). Nakon što je formirana tražena lista, vrši se kretanje kroz listu, pri čemu se nakon svakih M napravljenih koraka odstranjuje ona osoba iz liste na kojoj se trenutno nalazimo. Odstranjivanje se izvodi tako što se pokazivač “sljedeci” osobe koja prethodi osobi na kojoj se trenutno nalazimo preusmjerava tako da ne pokazuje više na osobu na kojoj se trenutno nalazimo nego na osobu koja slijedi iza nje (čime se osoba efektivno odstranjuje iz kruga) i prelazimo na sljedeću osobu. Tom prilikom, pomoću operatora “delete” potrebno je izbrisati čvor koji odgovara odstranjenj osobi, da ne zauzima više memoriju. Postupak ćete ponavljati dok se ne eliminira svih N osoba, pri čemu je posljednja odstranjena osoba upravo novi poglavica (redni broj te osobe treba vratiti kao rezultat iz funkcije). U svim ostalim detaljima (osim u načinu realizacije funkcije “Poglavica”), ovaj program treba biti identičan programu iz prethodnog zadatka.
5. Riješite ponovo prethodni zadatak, ali tako što će polje “sljedeci” u čvornoj strukturi “Clan” biti “pametni” umjesto običnog pokazivača. Suštinska razlika je što Vam ovaj put neće trebati operator “delete”, jer će svaki čvor automatski nestati čim se isključi iz lanca, s obzirom da tada niko neće pokazivati na njega. Međutim, morate paziti da kada ostane samo jedan čvor, njegovo polje “sljedeci” će pokazivati na njega samog (tj. upravo na taj jedini preostali čvor). Ovu vezu *obavezno morate raskinuti* prije nego što napustite funkciju, inače će doći do curenja memorije (jer će taj čvor nastaviti da “čuva samog sebe” čak i kad nestanu drugi pokazivači koji su na njega pokazivali).