

Programiranje 2

Vežbanje za kolokvijum: zadaci sa prethodnih kolokvijuma i ispita

ZADACI SA ČASA

Zadatak 1 Napisati rekurzivnu funkciju `int izmeni(int n)` koja svaku parnu cifru c u broju n zamenjuje sa $c/2$. Napisati glavni program koji kao argument komandne linije dobija broj n , a na standardni izlaz ispisuje novi broj. U slučaju greške na standardni izlaz za greške ispisati `-1`.

Primer 1:	Primer 2:	Primer 3	Primer 4:
<code>./a.out 1234</code>	<code>./a.out -8888</code>	<code>./a.out 1111</code>	<code>./a.out</code>
1132	-4444	1111	-1

Zadatak 2 Sa standardog ulaza se zadaje ime tekstualne datoteke koja sadrži podatke o artiklima prodavnice. Jedna linija datoteke je u formatu: `bar_kod ime_artikla proizvođač cena`. Datoteka je sortirana prema bar kodu. Bar kod može imati najviše 20 cifara. Ime artikla i proizvođač su niske i mogu imati najviše 50 karaktera. Cena je realan broj. Pretpostaviti da su podaci u datoteci ispravni (ne treba proveravati ispravnost bilo kog od navedena 4 podatka). Nije unapred poznat broj artikala u datoteci. Sa standardnog ulaza unose bar kodovi artikla sve dok se ne unese 0. Izračunati ukupnu cenu unetih proizvoda. (koristiti ugrađenu f-jku `bsearch` za traženje artikla sa datim bar kodom). U slučaju greške, na standardni izlaz za greške ispisati `-1`. U slučaju da korisnik unese bar kod koji se ne nalazi u datoteci, ispisati mu poruku "Nepostojeci bar kod" i omogućiti mu da nastavi unosenje.

```
Primer 1:
ulaz.txt:
0123456 Smoki Stark 51.5
0123777 Jafa Crvenka 60
1112222 Napolitanke200 Stark 95
1112223 Napolitanke400 Stark 190
4567891 Bajadera Kras 200
6718357 Noblice Banini 130

Standardni ulaz:
ulaz.txt
0123777
1112222
0123456
1112222
0

Standardni izlaz:
301.5
```

Zadatak 3 Napisati funkciju `int veci_od_proseka()` koja učitava cele brojeve sa standardnog ulaza sve dok se ne učitava 0. Broj brojeva nije unapred poznat. Funkcija vraća broj brojeva većih od proseka. Ako nema unetih

brojeva funkcija treba da vrati 0. Testirati napisanu funkciju pozivom iz glavnog programa i na standardni izlaz ispisati rezultat izvršavanja funkcije. U slučaju greške, na standardni izlaz za greške ispisati -1.

Primer 1: 1 2 3 4 5 0 2	Primer 2: -1 2 -3 4 -5 6 9 9 1 1 6 6 0 8	Primer 3: 0 0
--------------------------------------	---	----------------------------

Zadatak 4 Argumenti komadne linije su opcija (-m, -v ili -mv) i reč. Ukoliko je opcija -m pretvoriti sva slova reči u mala slova, ukoliko je opcija -v pretvoriti sva slova u reči u velika slova, a ukoliko je opcija -mv pretvoriti sva mala slova u velika, a sva velika slova u mala slova. Ukoliko opcija nije zadata ili je netačno navedena ispisati -1 na standardni izlaz za greške.

Primer 1: ./a.out -m Dan28Mesec2 dan28mesec2	Primer 2: ./a.out -v DanDanas;Pamtim DANDANAS;PAMTIM	Primer 3: ./a.out -mv VarljivoLeto68 vaRLJIVOlET068	Primer 4: ./a.out -nn greska -1
---	---	--	--

Zadatak 5 Napisati funkciju `void svaka_n_ta(char* ime, int n)` koja prima ime datoteke i ceo broj `n` i na standardni izlaz ispisuje svaku `n`-tu reč iz datoteke. Napisati program koji testira napisanu funkciju. Ime datoteke i broj `n` se zadaju kao argumenti komandne linije. U slučaju greške na standardni izlaz za greške ispisati -1.

Primer 1: ./a.out ulaz.txt 2 ulaz.txt: cao danas je lep dan mi mnogo volimo da programiramo Standardni izlaz: danas lep mi volimo programiramo

Zadatak 6 Napisati funkciju `float nula(float a, float b, float eps)` koja računa nulu funkcije $f(x) = 5 * \sin(x) * \ln(x)$ na intervalu (a, b) sa tačnošću `eps`. Brojevi `a`, `b` i `eps` unose se sa standardnog ulaza. Napomena1: koristiti algoritam binarne pretrage
Napomena2: u `math.h` nalaze se `float sin(float)` za računanje sinusa i `float log(float x)` za računanje prirodnog logaritma.

Testirati funkciju pozivom u `main-u`. U slučaju greške na standardni izlaz za greške ispisati -1.

Primer 1: 0.9 2 0.01 0.998828	Primer 2: 2 4 0.000001 3.141593	Primer 3: 4 10 0.001 6.283142	Primer 4: 10 20 0.001 12.566376
--	--	--	--

Zadatak 7 Napisati program koji ispisuje na standardni izlaz koliko elemenata u njegovoj komandnoj liniji jesu palindromi (reči koje su iste kada se čitaju sa leva na desno, i sa desna na levo).

Primer 1: ./a.out ana_ana Milan MAMA TegeT W 3	Primer 2: ./a.out MaMa anka 0	Primer 3: ./a.out 0	Primer 4: ./a.out Ana H202H 1
---	--	----------------------------------	--

Zadatak 8 U datoteci **matrica.txt** nalaze se podaci o kvadratnoj matrici. U prvom redu datoteke data je njena dimenzija, a potom slede elementi matrice. Napisati program koji alocira memorijski prostor za matricu i potom je učitava. Ispisati indekse onih redova matrice u kojima su elementi sortirani neopadajuće (redovi se indeksiraju počevši od nule). U slučaju greške ispisati -1 na standardni izlaz za greške.

Primer 1: matrica.txt: 2 -1 3 2 1 0	Primer 2: matrica.txt: 3 1 3 2 -5 2 2 2 4 5 1 2	Primer 3: matrica.txt: -5 -1	Primer 4: matrica.txt: 3 4 4 3 1 0 -3 5 4 3
---	--	--	---

Zadatak 9 U svakom redu datoteke **saldo.txt** nalazi se identifikacija (niska maksimalne dužine 20) korisnika banke i iznos novca koji korisnik trenutno ima (ceo broj). Svaki korisnik se pojavljuje tačno jednom i njegov saldo je predstavljen celim brojem (negativan - korisnik je zadužen, pozitivan - korisnik ima pozitivan saldo i nije zadužen). Izračunati koliko su prosečno zaduženi korisnici (pri računanju proseka ne računati one korisnike koji nisu zaduženi). Na standardni izlaz ispisati identifikacioni broj onih korisnika koji su zaduženi više od proseka. Maksimalan broj korisnika nije unapred poznat. U slučaju greške na standardni izlaz za greške ispisati -1.

Primer 1: saldo.txt abc123 -60 abc321 100 aat987 -100000 dfa451 1000000 xyz111 -110000 qwe120 100000 poi333 -130000 ugh245 120 ert671 -8000 das -9000 Standardni izlaz: aat987 xyz111 poi333
--

Zadatak 10 Argumenti komandne linije su celi, pozitivni brojevi. Napisati program koji ispisuje broj elemenata komandne liniji čije su cifre uredjene strogo rastuće.

Primer 1: ./a.out 26 13 468 3	Primer 1: ./a.out 2 14 41 2	Primer 1: ./a.out 0	Primer 1: ./a.out 423 189 243 117 258 2
--	--	--------------------------------------	--

Zadatak 11 Sa standardnog ulaza se učitava ceo broj x , ceo broj n , a potom i niz od n celih brojeva. Napisati rekurzivnu funkciju **void f11(int* a, int n, int x)**, koja u nizu a poslaavlja na nulu sve parove susednih elementa čiji je zbir (u početnom nizu) jednak x . Rezultujući niz ispisati na standardni izlaz. U slučaju greške na standardni izlaz za greške ispisati -1.

Primer 1: 5 10 1 2 3 2 5 1 5 0 4 6 1 0 0 0 5 1 0 0 4 6
--

Zadatak 12 Sa standardnog ulaza se učitava ceo broj n , a potom i niz od n celih brojeva. Napisati rekurzivnu funkciju `void f12(int *a, int n)` koja u nizu a postavlja na nulu sve elemente koji su (u početnom nizu) jednaki zbiru svojih suseda. Prvi i poslednji element ne treba menjati (jer nemaju oba suseda). Rezultujući niz ispisati na standardni izlaz.

```
Primer 1:
10
-1 2 3 1 5 1 -4 0 4 6
-1 0 0 1 5 0 -4 0 4 6
```

Zadatak 13 Sa standardnog ulaza se učitava ceo broj k , ceo broj n , a potom i niz od n celih brojeva, uredjenih rastuće. Napisati funkciju koja binarnom pretragom nalazi indeks prvog k -tocičnog elementa (element sa najmanjim indeksom koji ispunjava dato svojstvo). Ispisati dobijeni indeks na standardni izlaz (u slučaju da takav broj ne postoji ne pisati ništa). U slučaju greške ispisati -1 na standardni izlaz za greške.

Primer 1:	Primer 2:	Primer 3:	Primer 4:
2 10	3 10	4 5	1 1
1 4 7 10 12 300 350 600 700 1000	1 4 7 10 12 300 350 600 700 1000	300 350 600 700 1000	20
3	5	4	

Zadatak 14 Ceo broj x se učitava sa standardnog ulaza. Napisati rekurzivnu funkciju `int f14(int x)` koja u datom broju x uklanja sve cifre koje su (u početnom broju) jednake zbiru svojih suseda. Rezultat funkcije ispisati na standardni izlaz.

Primer 1:	Primer 2:	Primer 3:	Primer 4:	Primer 5:	Primer 6:	Primer 7:
1214	8880	-121220	1234	2	11	-11
114	880	-1120	1234	2	11	-11

Zadatak 15 Sa standardnog ulaza se učitava ceo broj x , ceo broj n , a potom i niz od n celih brojeva, uredjenih rastuće. Napisati funkciju koja u rastuće uredjenom nizu celih brojeva pronalazi broj koji je najbliži datom broju x . Ukoliko ima više takvih brojeva pronalazi onaj sa najmanjim indeksom. Funkcija vraća vrednost pronađenog broja i treba da radi u vremenu $O(\log(n))$. Ispisati dobijeni broj na standardni izlaz. U slučaju greške ispisati -1 na standardni izlaz za greške.

Primer 1:	Primer 2:	Primer 3:
11 10	3 10	-5 5
1 4 7 10 12 300 350 600 700 1000	1 4 7 10 12 300 350 600 700 1000	300 350 600 700 1000
10	4	300

Zadatak 16 Sa standardnog ulaza se učitava ceo broj n , a potom i niz od n celih brojeva. Napisati rekurzivnu funkciju `int f16(int a[], int n)` koja računa proizvod svih neparnih brojeva prosledjenog celobrojnog niza. Rezultat funkcije ispisati na standardni izlaz. U slučaju greške ispisati -1 na standardni izlaz za greške.

Primer 1:	Primer 2:	Primer 3:	Primer 4:
10	4	4	4
1 4 7 10 12 300 350 600 700 1000	1 3 5 7	2 4 6 8	1 2 4 6
28	105	-	1

Zadatak 17 Kao argument komadne linije zadaje se jedna reč. Ispisati na standardni izlaz reč koja se dobije od zadate reči tako što se prvo slovo ponovi jednom, drugo dva puta, ..., n-to n puta. U slučaju greške ispisati -1 na standardni izlaz za greške.

Primer 1: ./a.out cao	Primer 2: ./a.out garfield	Primer 3: ./a.out
caaooo	gaarrrrffffiiiiilllll11111ddddd	-1

Zadatak 18 Kao argument komadne linije zadaju se tri parametra – reč, slovo, broj. Izmeniti reč tako da se između prva dva pojavljivanja datog slova u reči svaki karakter uveća za dati broj. U slučaju greške ispisati -1 na standardni izlaz za greške.

Primer 1: ./a.out danas a 3	Primer 2: ./a.out danas n 3	Primer 3: ./a.out oktobar 50	Primer 4: ./a.out proGramiraNjer r 5
daqas	danas	-1	prrLramiraNjer