

Zadatak 1. (6 poena) Napisati funkciju

```
int prepis(char a[][21], int na, char b[][21])
```

koja iz niza reči **a** dužine **na** prepisuje u niz **b** reči koje su zapisane samo malim ili samo velikim slovima. Informaciju o dužini niza **b** (broj reči koje zadovoljavaju prethodni uslov) vratiti kao povratnu vrednost funkcije.

Zadatak 2. Napisati program koji sa standardnog ulaza učitava prvo broj reči (strogo veći od nule, manji od 50), a zatim i same reči razdvojene blanko znakom (smatrati da reči koje se unose sa ulaza neće biti duže od 20 karaktera - ovaj uslov ne proveravati). Za slučaj kada je broj reči izvan traženog opsega ispisati -1 i prekinuti izvršavanje programa. Korišćenjem prethodno definisane funkcije `prepis`, odrediti sve reči koje su zapisane samo malim ili samo velikim slovima. Rezultat ispisati na standardni izlaz. Napomena: Ukoliko se pri rešavanju zadatka ne bude koristila funkcija `prepis`, zadatak neće biti pregledan i nosiće nula poena.

Primer 1:	Primer 2:	Primer 3:	Primer 4:
3	4	2	-3
abc ABC aBc	2abc AVF\$ abc AV4	mnB RGa	-1
abc ABC	abc		

Zadatak 3. Sa standardnog ulaza unose se neoznačeni brojevi **N** i **A**, a potom i **N** celih brojeva. Odrediti koliko ima parova uzastopnih brojeva koji se nalaze na rastojanju **A**.

Primer 1:	Primer 2:	Primer 3:	Primer 4:
5	3	10	6
2	19	5	20
2 3 5 1 -1	-30 1 12	-3 6 11 -20 -25 -8 42 37 1 6	1 -19 2 7 27 3
2	0	4	2

Zadatak 4. Napisati funkciju `void min_razlika(char s[], char s1[], char s2[])` koja u datotoj nisci **s** pronalazi dve reči koje imaju minimalnu razliku između svojih samoglasnika. (Reč je niz karaktera između dve praznine; razmak između samoglasnika reči **danas** i **jutro** je 2, a razmak između **sutrk** i **mnozenje** je 5). Testirati pozivom u `main-u`. Maksimalna dužina niske je 20 karaktera.

Zadatak 5. Napisati funkciju `int f4(int a[], int n, int b[], int m)` koja modifikuje prvi niz tako da ne sadrži ni jedan element drugog niza. Funkcija vraća novu dimenziju niza. Testirati pozivom u `main-u`. Maksimalna dužina niza je 100 elemenata.

Zadatak 6. Napisati funkciju `int pp(char s[], char t[])` koja određuje poziciju poslednjeg karaktera niske **s** sadržanog u okviru niske **t**, zanemarujući pri tom razliku između velikih i malih slova, ili -1 ako takvog karaktera nema. Testirati pozivom u `main-u`. Maksimalna dužina niske je 20 karaktera.

pp("a4BA3Bc", "A3b") = 5

Zadatak 7. Napisati funkciju `int f1(char s[])` koja prihvata tu nisku i proverava da li niska sadrži veliko slovo. Funkcija vraća 1 ako sadrži veliko slovo, inače 0. Testirati pozivom u main-u. Maksimalna dužina niske je 20 karaktera.

Zadatak 8. Napisati funkciju `void ukloniSlova(char s[])` koja iz niske `s` uklanja sva mala i velika slova. Testirati pozivom u main-u. Maksimalna dužina niske je 20 karaktera.

Zadatak 9. Napisati funkciju `int f3(int a[], int n, int b[], int m)` i ispituje da li prvi sadrži bar dva broja koji se pojavljuju u drugom nizu. Povratna vrednost je dakle, 0, ili 1. Testirati pozivom u main-u. Maksimalna dužina niza je 100 elemenata.

Zadatak 10. Napisati funkciju `unsigned btoi(char* s, unsigned char b)` koja određuje vrednost zapisa datog neoznačenog broja `s` u datoj osnovi `b`. Napisati funkciju `void itob(unsigned n, unsigned char b, char* s)` koja datu vrednost `n` zapisuje u datoj osnovi `b` i smešta rezultat u nisku `s`. Napisati zatim program koji čita liniju po liniju sa standardnog ulaza i obrađuje ih sve dok ne naiđe na praznu liniju. Svaka linija sadrži jedan dekadni, oktalni ili heksadekadni broj (zapisan kako se zapisuju konstante u programskom jeziku C). Program za svaki uneti broj ispisuje njegov binarni zapis. Pretpostaviti da će svi uneti brojevi biti u opsegu tipa `unsigned`.

Primer 1:	Primer 2:	Primer 3:	Primer 4:
0x49	012	123	981
1001001	1010	1111011	1111010101
0x1ABC	435	0777	
1101010111100	110110011	111111111	
	0x64FE		
	110010011111110		

Zadatak 11. Sa standardnog ulaza se unosi ceo neoznačeni broj `n`, a zatim i niz od `n` elemenata celobrojnog tipa (njih najviše 100). Napisati program koja kvadrira one elemente niza koji su veći od aritmetičke sredine niza, a zatim štampa modifikovani niz. U slučaju neregularnog unosa štampa se poruka: GRESKA.

Primer 1:	Primer 2:
ulaz: 5 1 6 7 2 3	ulaz: 7 23 4 67 -34 56 12 89
izlaz: 1 36 49 2 3	izlaz: 23 4 4489 -34 3136 12 7921
Primer 3:	Primer 4:

ulaz: 4 12 789 134 390

ulaz: 102

izlaz: 12 622521 134 152100

izlaz: GRESKA

Zadatak 12. Implementirati funkciju `int str_str(char s[], char t[])` koja proverava da li niska `s` sadrzi nisku `t`. Zatim napisati program koji sa standardnog ulaza učitava pet redova (svaki red ima najviše 100 karaktera) i koji ispisuje sve redne brojeve linija koje sadrže nisku `program` (linije se numerišu od broja 1). Ukoliko ne postoji red sa niskom `program` ispisati `-1`.

Primer 1:

Ulaz:

```
novi red*program
c prog. jezik
c? programskih jezik
Programski odbor
<b>program</b>
```

Izlaz: 1 3 5

Primer 2:

Ulaz:

```
PROGRAMIRANJE
program1.c
grupa2
Programator
Paralelno programiranje
```

Izlaz: 2 5

Primer 3:

Ulaz:

```
Programiranje
algebra
linearno programiranje
prvi_program.c
progres americkih trupaca
```

Izlaz: -1

Zadatak 13. Napisati C funkciju koja u prosleđenom nizu eliminiše sve brojeve koji nisu deljivi svojim indeksom (vrednost na indeksu 0 zadržati, jer nije dozvoljeno deljenje sa 0). Niz reorganizovati, tako da nema rupa koje su nastale eliminacijom elemenata. Kao rezultat funkcije vratiti novu dimenziju niza.

Na primer:

niz $a = \{4, 2, 1, 6, 7, 8, 10, 2, 16, 3\}$ se transformiše u niz $a = \{4, 2, 6, 16\}$ i vraća se vrednost 4.

Zadatak 14. Napisati funkciju `int novi_niz(int a[], int b[], int n, int c[])` koja na osnovu nizova `a` i `b` pravi niz `c` tako što se naizmenično raspoređuju elementi nizova `a` i `b`, tj. $a_0, b_0, a_1, b_1, a_2, b_2, \dots$. Nizovi `a` i `b` su iste dimenzije i njihova dimenzija je `n`. Funkcija vraća dimenziju niza `c`. Funkciju testirati pozivom u `main-u`, pri čemu se sa standardnog ulaza učitava `n`, potom nizovi `a` i `b` dužine `n`. Nakon poziva funkcije na standardni izlaz ištampati niz `c`.

Zadatak 15. Napisati funkciju `int VратиPoziciju(int a[], int n, int b[], int m)` koja vraća prvu poziciju u nizu `a` na kojoj se pojavljuje broj iz niza `b` ili `-1` ako `a` ne sadrži ni jedan broj iz `b`.

Napr.

Ako je $a = \{3, -5, 7, 11\}$ i $b = \{2, 4, 3, 11\}$ onda funkcija treba da vrati poziciju 0.

Ako je $a = \{3, -5, 7, 11\}$ a $b = \{11, -7, 2, 8\}$ onda funkcija treba da vrati poziciju 3.

Program testirati pozivom funkcije iz `main` programa i ispisom rezultata na standardni izlaz, pri čemu korisnik sa standardnog ulaza učitava broj `n` potom niz `a` dužine `n`, potom učitava broj `m` i potom niz `b` dužine `m`.

Zadatak 16. Implementirati funkciju `int min_max(int a[], int n)` koja prihvata celobrojni niz, pronalazi indekse najmanjeg i najvećeg elementa tog niza koristeći samo jedan prolaz (jednu petlju), a zatim kao povratnu vrednost vraća manji od ta dva indeksa.

Program testirati pozivom funkcije iz main programa i ispisom rezultata na standardni izlaz, pri čemu korisnik sa standardnog ulaza unosi niz dužine 10 elemenata.

Zadatak 17. Napisati funkciju `void brojanje(int a[], int brojac[], int N)` čiji su argumenti `a` i `brojac` celobrojni nizovi dimenzije `N`. Vrednosti elemenata niza `a` su između 0 i `N - 1`. Funkcija izračunava elemente niza `brojac` tako da je `brojac[i]` jednak broju pojavljivanja broja `i` u nizu `a`. Program testirati pozivom funkcije iz main programa - korisnik učitava broj `N` i potom niz `a` dužine `N`, potom poziva funkciju i potom na standardnom izlazu izpisuje dobijeni niz.

Zadatak 18. Napisati funkciju `int ind(int a[], int n)` koja kao povratnu vrednost ima indeks onog elementa niza koji je po vrednosti najbliži srednjoj vrednosti onih elemenata niza brojeva koji su deljivi sa 3.

Program testirati pozivom funkcije iz main programa i ispisom rezultata na standardni izlaz, pri čemu korisnik sa standardnog ulaza unosi broj `n`, a zatim niz od `n` celih brojeva (maksimalna dimenzija niza je 100 elemenata). Napr.

za ulaz: `n = 5, a = 1, 2, 3, 4, 5` ispis: 2

za ulaz: `n = 5, a = 3, 6, 2, 4, 7` ispis: 3

Zadatak 19. Napisati funkciju `void sifrat(char* rec, char* kljuc)` koja šifruje `rec` na sledeći način: za svako slovo reči `rec` i odgovarajuće slovo ključa određuje koliki je (alfabetski) razmak između njih i označimo taj broj sa `k`. Potom to slovo reči zamenjuje `k`-tim slovom alfabeta. Podrazumeva se da je ključ duži od reči.

PRIMER:

za `rec = bac` i `kljuc = dfge` rezultat je `bed`

Zadatak 20. Napisati funkciju `void obrni(char rec[], int k)` koja rotira reč za `k` mesta ulevo, napr. za reč `sveska` i `k=2`, rezultat treba da bude `eskasv`.

Zadatak 21. Napisati sledeće funkcije:

```
int poredjenje(char* s1, char* s2);  
// vraća 1 ako su s1 i s2 iste niske, 0 u suprotnom
```

```
void uVelikaSlova(char* s);
```

```
// pretvara sva slova niske s u velika, ostale znakove ne menja
```

Napisati program koji sa standardnog ulaza učitava dve reči (dužine najviše 20 znakova) i, koristeći ove dve funkcije, ispisuje da li su one jednake ako se sva slova pretvore u velika slova. Na primer, reči "isPit2010" i "IsPiT2010" su jednake ako se sva slova pretvore u velika.