Univerzitet u Sarajevu
Elektrotehnički fakultet Sarajevo
Osnove računarstva

Laboratorijska vježba 10

Cilj ove vježbe je vježbanje dodatnih zadataka iz oblasti stringova kao priprema za ovaj tip zadataka na Drugom parcijalnom ispitu.

I UVODNI DIO VJEŽBE

1. Unesite funkciju sa predavanja i program (za funkciju unesi imate prototip, a implementacija se nalazi u lab vježbi 9):

```
#include <stdio.h>
void unesi(char niz[], int velicina);
void zamijeni1(char* s) {
	int duzina = 4;
	while (*s != '\0') {
		if (*s == '1') {
			char* kraj = s;
			while (*kraj != '\0') kraj++;
			while (kraj > s) {
				*(kraj+duzina) = *kraj;
				kraj--;
			}
			*s++ = 'j'; *s++ = 'e'; *s++ = 'd';
			*s++ = 'a'; *s = 'n';
		}
		s++;
	}
}
int main() {
	char tekst[100];
	printf("Unesite tekst: ");
	unesi(tekst, 100);
	zamijeni1(tekst);
	printf("Nakon zamjene: %s", tekst);
	return 0;
}
```

a) Analizirajte program.
b) Kompajlirajte program.
c) Testirajte program tako što ćete tri puta unositi različite riječi i znakove koji se prebrojavaju.

II ZADACI ZA SAMOSTALNU VJEŽBU

2. Napišite funkciju pod nazivom `rot13` koja vrši šifrovanje i ujedno dešifrovanje primljenog stringa po ROT13 algoritmu.

Pod pojmom "ROT13 algoritam" misli se na to da se svako slovo u primljenom tekstu zamjenjuje slovom koje je udaljeno za 13 mjesta u engleskoj abecedi. Npr. slovo A se zamjenjuje slovom N, slovo B slovom O itd. Ako se pri brojanju mjesta dođe do slova Z, brojanje se nastavlja od početka, pa se npr. slovo M zamjenjuje slovom Z, a slovo N se zamjenjuje slovom A jer smo krenuli od početka abecede, slovo O se zamjenjuje slovom B itd. Iz ovoga vidimo da ista funkcija radi i šifrovanje i dešifrovanje teksta pošto engleska abeceda ima tačno 26 slova.

Pazite da se u funkciji velika slova zamjenjuju velikim a mala malim, dok ostale znakove (razmake, brojeve itd.) ostavljate nepromijenjenim. Radi lakšeg lančanog pozivanja, funkcija treba da vraća pokazivač na prvo slovo primljenog stringa.

Napišite i glavni program u kojem se unosi string, poziva funkcija i zatim ispisuje (de)šifrovani string.

3. Napravite funkciju `izbaci_rijec` koja prima string koji predstavlja rečenicu i neki cijeli broj **n**, a zatim iz rečenice izbacuje **n**-tu po redu riječ. Ako je **n** manje od jedan ili veće od broja riječi u stringu, funkcija ne treba uraditi ništa. Radi lakšeg lančanog pozivanja, funkcija treba vraćati pokazivač na početak primljenog stringa.

Riječ se u ovom zadatku definiše kao neprekinuti niz velikih i malih slova. Svaki drugi karakter u stringu (tačka, zarez, minus itd.) smatraće se prekidom riječi. Npr. ako je dat string "Auto-Stop", smatraće se da se on sastoji iz dvije riječi od po četiri slova. Znakove koji ne pripadaju riječi (razmaci i sl.) ne treba izbacivati.

Napišite i kraći glavni program koji demonstrira korištenje ove funkcije.

4. Napravite funkciju sa prototipom:

```
char* whitespace(char* s)
```

koja u datom stringu zamjenjuje sve "whitespace" znakove jednim razmakom. Dakle, novi redovi (`\n`) i tabovi (`\t`) trebaju biti zamijenjeni razmacima. Također, ukoliko se javlja više razmaka zaredom, treba ih zamijeniti jednim razmakom. Whitespace znakove na početku ili kraju stringa treba izbaciti. Ovo uključuje i kombinacije razmak-tab ili novi red-razmak itd. Funkcija treba vratiti pokazivač na prvi znak primljenog stringa radi lakšeg lančanog pozivanja. Napravite i kraći program koji demonstrira korištenje ove funkcije.

U zadatku je zabranjeno korištenje funkcija iz biblioteke **string.h**.

Primjer: ako string glasi

```
Ovo
      je
neki                tekst
```

potrebno je zamijeniti ga sljedećim stringom:

```
Ovo je neki tekst
```

5. Napišite funkciju `ukloni_komentare` koja iz datog stringa, koji predstavlja C kod, izbacuje sve komentare. Funkcija treba da podržava komentare u C stilu:

```
/* komentar */
```

i u C++ stilu:

```
// komentar.....
```

Napravite i kraći program koji demonstrira korištenje ove funkcije.

U zadatku je zabranjeno korištenje funkcija iz biblioteke **string.h**.

6. Napraviti funkciju `tritacke` koja u datom stringu sve riječi dužine 10 ili više slova pretvara u skraćeni oblik - prva tri slova i znak tačka, te vraća pokazivač na prvo slovo stringa. Primjer:

Ulaz:

```
Za vrijeme promjena specifikacija moze doci do poremecaja
tolerancije izlaznog proizvoda
```

Izlaz:

```
Za vrijeme promjena spe. moze doci do por. tol. izlaznog
proizvoda
```

Radi jednostavnosti, ne trebate posebno tretirati znakove interpunkcije. Riječ je svaki dio teksta koji je ograničen sa jednim ili više razmaka.

Izmjene:

-