

# Ispit iz Programiranja 2

Ispit traje 120 minuta

## Napomene:

- a) Pažljivo proučite Uputstvo pre popunjavanja Obrasca za odgovore.  
 b) Vrednost odgovora: tačan = **5**; netačan = **-1.25**; nevažeći (nula ili više zacrnjenih kružića) = **0**.  
 c) Na pitanjima se može osvojiti najviše **15** poena. Prvi zadatak nosi **25** poena, dok drugi nosi **30** poena.

## I ZADACI

**1)** Napisati program na programskom jeziku C koji obrađuje tekst učitani sa standardnog ulaza. Tekst se čita iz proizvoljnog broja redova sve dok se ne dođe do praznog reda. Svakim učitani red se potom ispisuje na standardni izlaz uz prelamanje (ako ima više od N reči). Redove teksta koji imaju više od N reči potrebno je prelomiti u više manjih redova, tako da svaki manji red ima tačno N reči, osim poslednjeg u kojem može biti najviše N reči. Maksimalne dužine reči i reda nisu unapred poznate. Poznato je da se u ulaznom tekstu pojavljuju samo slova i blanko znakovi. Reči su razdvojene sa jednim ili više blanko znakova. Parametar N se učitava sa standardnog ulaza na početku programa. Voditi računa o korektnoj upotrebi resursa.

**2)** U svakom redu datoteke **korisnici.txt** zapisani su podaci o jednom korisniku u formatu: korisničko ime i e-mail adresa, razdvojeni jednim blanko znakom. Korisničko ime, kao i e-mail adresa su jedinstveni podaci i pojavljuju se u najviše jednom redu. U svakom redu datoteke **log.txt** nalaze se podaci o istoriji prijavljivanja i odjavljivanja korisnika na sistem u formatu: e-mail, akcija, vreme. Akcija može imati oznaku „p” ili „o” (p – prijava, o – odjava), dok je vreme ceo broj i predstavlja trenutak u kome se akcija izvršila. Jedan korisnik se može više puta prijaviti i odjaviti sa sistema. Napisati **potprogram** koji formira listu korisnika tako što u listu dodaje svakog pročitano korisnika iz datoteke **korisnici.txt**. Zatim napisati **potprogram** za učitavanje podataka iz datoteke **log.txt** i ažuriranje liste na osnovu e-mail adrese, tako da na kraju rada potprograma za svakog korisnika bude izračunato ukupno vreme koje je proveo prijavljen na sistem (za potrebe računanja u listi je dozvoljeno čuvati i druge podatke). Smatrati da je datoteka **log.txt** ispravna i da su svi korisnici odjavljeni sa sistema. Koristeći implementirane potprograme napisati **program** koji ispisuje ime korisnika koji je bio najviše vremena prijavljen na sistem. Ukoliko ima više takvih korisnika, dovoljno je ispisati jednog od njih. Voditi računa o korektnoj upotrebi resursa.

korisnici.txt	log.txt	Izlaz
milan milan@etf.rs zoran zoran@gmail.com ana ana@etf.rs	milan@etf.rs p 10 zoran@gmail.com p 20 milan@etf.rs o 30 zoran@gmail.com o 35 milan@etf.rs p 40 milan@etf.rs o 60 ana@etf.rs p 50 ana@etf.rs o 90	Milan

## II PITANJA

**1)** Šta ispisuje sledeći program na programskom jeziku C? Smatrati da je podatak tipa short smešten na širini od 16 bitova, kao i da program pri izvršavanju niži bajt smešta na nižu adresu, a viši bajt na višu adresu.

<pre>#include &lt;stdio.h&gt;  void f(short num, short* res) {     if (num) {         *res = *res * 10 + num%10;         f(num / 10, res);     } }</pre>	<pre>typedef union {     short br;     struct { char a, b; } str; } un;  void main() {     short res = 0; un x;     x.str.a = 4; x.str.b = 2;     f(x.br, &amp;res); x.br = res;     printf("%d %d %d\n", x.br, x.str.a, x.str.b); }</pre>
--	--

A) 6201 57 24

B) -516 4 2

(C) 615 103 2

2) Šta radi funkcija **rebus** napisana na programskom jeziku C nad nizom **arr** dužine **n**? Smatrati da je **d** manje ili jednako **n**.

<pre>void tumble(int *p, int *q) {     int temp;     while (q - p &gt; 0) {         temp = *p;         *p++ = *q;         *q-- = temp;     } }</pre>	<pre>void rebus(int *arr, int d, int n) {     tumble(arr + 0, arr + d - 1);     tumble(arr + d, arr + n - 1);     tumble(arr, arr + n - 1); }</pre>
--	---

A) Obrće poredak prvih **d** elementa niza **arr**.

(B) Rotira niz **arr** za **d** mesto ulevo (od viših ka nižim indeksima).

C) Izbacuje iz niza **arr** element sa pozicije **d**.

3) Šta ispisuje sledeći program na programskom jeziku C ukoliko jednostruko ulančana lista sadrži redom brojeve 3 3 8 5 9 7? Smatrati da funkcija **ucitajlistu()** ispravno formira, a funkcija **pisilistu()** ispravno ispisuje sadržaj jednostruko ulančane liste redom od početka.

<pre>#include &lt;stdio.h&gt; typedef struct elem {     int broj; struct elem *sled; } Elem; Elem* obradi (Elem*); Elem* ucitajlistu(void); void pisilistu(Elem*); void main () {     Elem *lst = ucitajlistu();     lst = obradi(lst);     pisilistu(lst); }</pre>	<pre>Elem* obradi (Elem* lst) {     Elem* t , *p = lst;     t = lst-&gt;sled;     while(t) {         Elem *s = t;         p-&gt;sled = t-&gt;sled;         p = t = t-&gt;sled;         if (p) t = p-&gt;sled;         free(s);     }     return lst; }</pre>
---	--

(A) 3 8 9

B) 3 3 9 7

C) 3 5 7

4) Šta ispisuje sledeći program na programskom jeziku C? Pretpostaviti da je program pozvan komandom **prog programiranje matematika elektrotehnika engleski**, gde je **prog** naziv programa.

<pre>#include &lt;stdio.h&gt; #include &lt;string.h&gt; void main(int argc, char* argv[]) {     int i;     int j;     i = 0;</pre>	<pre>while(i &lt; argc){     if(i%2){         for(j=0; j &lt; strlen(*(argv+i)); j += 2)             printf("%c", *((argv+i)+j));     }     i++; }</pre>
--	--

(A) pormrneeetoenk

B) .pomtmtkegek

C) rgaiajllkrthia