

Treći kolokvijum iz Programiranja 2

Kolokvijum traje 90 minuta

Napomene:

- a) Pažljivo proučite Uputstvo pre popunjavanja Obrasca za odgovore.
 b) Vrednost odgovora: tačan = **15**; netačan = **-3.75**; nevažeći (nula ili više zacrtnjenih kružića) = **0**.
 c) Na pitanjima se može osvojiti najviše **45** poena.
 d) Zadatak nosi **55** poena.

I ZADACI

1) 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 napisan na programskom jeziku C?

<pre>#include <stdio.h> int f1(int *p) { int d; d = *p; *p *= 1 + d; return *p /= 3; } typedef int (*tf) (int *); int x = 0; int f2(tf f, int *p) { int d; x++; if (*p < 4 && x < 2) { d = f2(f, p); </pre>	<pre> return (*f)(p) + d; } else if (*p < 4) { d = (*f)(p); return f2(f, p) + d; } return (*f)(p); } void main() { int a = 3; a = f2(f1, &a); printf("%d\n", x); }</pre>
---	--

(A) 3

B) 4

C) 5

2) 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 <stdio.h> 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, &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

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

(A) pormrneeetoenk

B) .pomtmtkegek

C) rgaiajkrthia