

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 pronalazi ljude koji neće dobiti poruku. Ljudi su međusobno povezani, a svaki čovek može imati najviše dva prijatelja od poverenja kojima može preneti poruku. Sa standardnog ulaza se učitava broj ljudi N ($N \leq 100$). Ljudi su numerisani brojevima od 0 do $N-1$. Zatim se za svakog čoveka unose dva redna broja prijatelja od poverenja sa kojima je u vezi i kojima posmatrani čovek može poslati poruku. Ukoliko neki prijatelj ne postoji, unosi se vrednost -1 . Uneti podaci moraju biti takvi da ako je osoba A prijatelj sa osobom B, onda je i osoba B prijatelj sa osobom A. Nakon unosa sa standardnog ulaza se unosi broj čoveka kojem se daje poruka da je razglasi. Čovek koji primi poruku, posleduje je prijateljima kojima može da pošalje poruku i oni dalje prosleđuju poruku prijateljima kojima mogu i tako dalje. Svaki čovek samo jednom prosleđuje poruku. Program treba da odredi i ispiše sve ljude koji neće primiti poruku. Učitavanje ljudi i obradu realizovati kao zasebne potprograme koji sa glavnim programom komuniciraju isključivo putem argumenata i povratnih vrednosti.

2) Napisati program na programskom jeziku C koji filtrira oglase za posao. U svakom redu datoteke `poslovi.txt` nalaze se informacije o jednom oglasu za posao u sledećem formatu: naziv posla (niz znakova, koji može sadržati blanko znake, od najviše 30 karaktera) praćen dvotačkom, naziv kategorije kojoj posao pripada (niz znakova, koji može sadržati blanko znake, od najviše 20 karaktera) praćen dvotačkom i plata na tom poslu (ceo broj). Poslovi su navedeni proizvoljnim redosledom. Broj poslova i broj kategorija nije unapred poznat. Smatrati da ne postoje dve kategorije, kao ni dva posla sa istim nazivom. Korisnik sa standardnog ulaza unosi željenu kategoriju posla u jednom redu, a u narednom redu minimalnu odgovarajuću platu. Program treba da učita podatke o poslovima u jednostruko ulančanu listu poslova koja u svakom trenutku treba da bude uređena nerastuće po iznosu plate, nađe odgovarajuće poslove u zavisnosti od korisničkog unosa i ispiše ih na standardnom izlazu. Pretraga poslova se nastavlja dok god korisnik ne unese prazan red. Voditi računa o korektnoj upotrebi resursa.

II PITANJA

1) Šta ispisuje sledeći program napisan na programskom jeziku C?

```
#include <stdio.h>
#define SIZE 3
typedef int (*FP) (int *);
int f1(int *a) { return *(a - 1) = *a >> 2 - 1; }
int f2(int *b) { return --*b++; }
int f3(int *c) { int a = *c; return *c = a++, a % 2 ? a : 0; }
void main() {
    int arr[] = { 7, 3, 4 }, i = 0;
    FP fun[] = { f2, f1, f3 };
    while (i++, fun[i%SIZE](arr + i%SIZE));
    for (i = 0; i < SIZE; printf("%d", arr[i++]));
}
```

(A) 034

B) 035

C) -124

2) Šta ispisuje sledeći program napisan na programskom jeziku C? Funkcija `strstr(d, r)` vraća pokazivač na prvu pojavu podstringa `r` u stringu `d`, ili `NULL` ukoliko se `r` ne pojavljuje u `d`.

<pre>#include <stdio.h> #include <stdlib.h> #include <string.h> void main() { char *p = "abcdedef", *d = "defghi", *t = calloc(strlen(p) + strlen(d) + 1, sizeof(char)), *r, *q, *s; r = p; q = t; </pre>	<pre>while (*r) { while (*r != *d) *q++ = *r++; s = strstr(d, r); if (s) { s += strlen(r) - 1; strcat(t, s); break; } else *q++ = *r++; } printf("%s", t); free(t); }</pre>
---	---

(A) abcdcddefghi

B) abcdcdedfghi

C) abcdcdedfdefghi

3) Šta radi sledeći program napisan na programskom jeziku C nad argumentima koji se zadaju putem komandne linije?

<pre>#include <stdio.h> #include <stdlib.h> #include <string.h> void main(int argc, char *argv[]) { char **s; int i, j, k = 0, d; s = malloc((argc-1)*sizeof(char*)); for (i = 1; i < argc; i++) { d = strlen(argv[i]); for (j = k - 1; j >= 0; j--) if (strlen(s[j]) > d) s[j+1] = s[j]; else break; </pre>	<pre>s[j+1] = malloc((d+1) * sizeof(char)); strcpy(s[j+1], argv[i]); k++; } for (i = 0; i < k; i++) { puts(s[i]); free(s[i]); } free(s); }</pre>
---	---

A) Formira i ispisuje niz stringova zadatih putem komandne linije u leksikografskom poretku.

(B) Formira i ispisuje niz stringova zadatih putem komandne linije neopadajuće uređen po dužinama stringova.

C) Formira i ispisuje niz svih stringova zadatih putem komandne linije dužih od imena programa.

4) Šta ispisuje sledeći program na programskom jeziku C ukoliko su u jednostruko ulančanu listu `lst1` redom uneseni brojevi 9 7 5 2 1, a u jednostruko ulančanu listu `lst2` redom uneseni brojevi 8 5 4? Smatrati da funkcija `citaj` ispravno formira, a funkcija `psi` ispravno ispisuje sadržaj zadate liste redom od početka.

<pre>#include <stdio.h> #include <stdlib.h> typedef struct elem { int broj; struct elem *sled; } Elem; Elem *f(Elem *lst1, Elem *lst2); void main () { Elem *lst, *lst1, *lst2; lst1 = citaj(); lst2 = citaj(); lst = f(lst1, lst2); psi(lst); brisi(lst); }</pre>	<pre>Elem *f(Elem *lst1, Elem *lst2) { Elem *lst = NULL, *tek; while (lst1 && lst2) { if (lst1->broj > lst2->broj) { tek = lst1; lst1 = lst1->sled; } else { tek = lst2; lst2 = lst2->sled; } tek->sled = lst; lst = tek; } brisi(lst1); brisi(lst2); return lst; }</pre>
--	---

(A) 4 5 5 7 8 9

B) 1 2 4 5 7 8 9

C) 9 8 7 5 4 2