

## Ispit iz Programiranja 2

Ispit traje 150 minuta.

**Napomene:**

- Pažljivo proučite Uputstvo pre popunjavanja Obrasca za odgovore.
- Vrednost odgovora: tačan = 5; netačan = -1.25; nevažeći (nula ili više zacrnjenih kružića) = 0.
- Na pitanjima se može osvojiti najviše 25 poena.
- Zadatak nosi 20 poena.

## I ZADACI

1) Napisati program na programskom jeziku C koji vrši određenu obradu nad listom tačaka u ravni sa realnim koordinatama. Program treba da utvrdi i na standardni izlaz ispiše sve one trojke tačaka koje čine jedan trougao, svaku pronađenu trojku u zasebnom redu. Tri tačke se ne nalaze na istoj pravoj ukoliko važi formula:  $x_1(y_2 - y_3) + x_2(y_3 - y_1) + x_3(y_1 - y_2) \neq 0$ . Program treba da učitava elemente jednostruko ulančane liste sve dok se ne unese tačka koja označava koordinatni početak, a zatim izvrši zahtevanu obradu. Učitavanje i obradu liste implementirati kao zasebne potprograme koji sa glavnim programom komuniciraju isključivo preko liste argumenata i povratne vrednosti. Voditi računa o korektnoj upotrebi dinamičke memorije.

2) Napisati program na programskom jeziku C koji određuje osvajače medalja na atletskim trkama. Informacije o trkama se nalaze u datoteci `trke.txt`. Za svaku trku postoji informacija o broju učesnika i nakon toga u sledećem redu su redom navedeni učesnici. Za svakog učesnika su navedeni prijavni broj učesnika i njegovo prolazno vreme u formatu `MM:SS` (`MM` minuti i `SS` sekunde). Program treba da odredi tri najbolja učesnika u svakoj trci i da ih upiše u datoteku `medalje.txt`. Za porednika trke program treba da upiše prijavni broj učesnika i njegovo prolazno vreme, a za učesnike koji su osvojili drugo i treće mesto treba da upiše prijavni broj i vreme zaostataka za porednikom u formatu `+MM:SS`. Pretpostaviti da ni na jednoj trci ne učestvuje više od 20 učesnika.

Primer <b>trke.txt</b>	Primer <b>medalje.txt</b>
4	4 00:47 1 +00:01 5 +00:01
1 00:48 2 00:49 5 00:48 4 00:47	4 00:21 5 +00:03
2	
5 00:24 4 00:21	

## II PITANJA

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

```
#include <stdio.h>
#include <ctype.h>
#include <string.h>
int fja(char *s1, char *s2);
void main () {
    char str[] = "prog 2 jul", *s;
    s = str + strlen(str) - 1;
    printf("%d", fja(str, s));
}

int fja(char *s1, char *s2) {
    int r;
    if (!isspace(*s1)) {
        r = *s1 >= *s2;
        if (r) s2--;
        return r + fja(++s1, s2);
    } else return 0;
}
```

- (A) 1                      B) 2                      C) 4

2) Šta ispisuje sledeći program na programskom jeziku C ukoliko jednostruko ulančana lista sadrži redom brojeve 1 3 2 5 3 2 1? Smatrati da funkcija `ucitajlistu()` ispravno formira, a funkcija `pisilistu()` ispravno ispisuje sadržaj jednostruko ulančane liste redom od početka.

<pre> typedef struct elem {     int broj; struct elem *sled; } Elem; Elem* obradi (Elem* lst); Elem* pomeri(Elem *lst){     Elem *p = lst, *e = lst;     while (e-&gt;sled)         { p = p-&gt;sled; e = e-&gt;sled-&gt;sled; }     return p; }  void main (){     Elem *lst = ucitajlistu();     obradi(lst); pisilistu(lst); } </pre>	<pre> Elem* obradi (Elem* lst) {     Elem *t1 = lst, *t2, *p = NULL, *s;     t2 = pomeri(lst);     while (t2) {         if (t1-&gt;broj == t2-&gt;broj) {             s = t1; t1 = t1-&gt;sled;             if (!p) lst = t1;             else p-&gt;sled = t1;             free(s);         } else { p = t1; t1 = t1-&gt;sled; }         t2 = t2-&gt;sled;     }     return lst; } </pre>
--	--

- (A) 1 5 3 2 1                      B) 1 3 5 3 1                      C) 2 5 3 2 1

3) Koja od sledećih deklaracija ispravno deklarise pokazivač na niz pokazivača na funkcije koje prihvataju argument tipa pokazivača na `char`, a vraćaju podatak tipa `int`?

- A) `int *(*p[])(char*);`      (B) `int ((*p)[])(char*);`      C) `int ((*(*p)(char*))[]);`

4) Koje od ponuđenih tvrdnji su tačne za programski jezik C?

A) Datoteka se otvara za čitanje i pisanje pomoću režima **"rw"** u funkciji **fopen()**.

(B) U prototipu funkcije ne moraju se navoditi imena argumenata.

C) Prilikom prenosa niza u funkciju pravi se kopija niza kojoj se pristupa u samoj funkciji.

5) Šta treba da stoji umesto \*\*\* da bi funkcija `replace` vršila zamenu podstringa, koji počinje i završava se znakom #, u stringu koji se prosledi kao prvi argument, sa stringom koji se prosledi kao drugi argument? Smatrati da u stringu koji se prosledi kao prvi argument ima tačno dva znaka #.

<pre> char* replace(char* prvi, char* drugi){     int n1 = strlen(prvi), n2 =strlen(drugi);     char* ret = calloc((n1 + n2 + 1), sizeof(char));     char *p,*q; </pre>	<pre>         strcpy(ret, prvi);         ***         strcat(ret,q);         return ret;     } </pre>
---	--

- |  |   |   |
|--|---|---|
| <b>(A)</b> p = ret;<br>while (*p != '#') p++;<br>q = prvi + n1;<br>while (*(q - 1) != '#') q--;<br>strcpy(p, drugi); | <b>B)</b> p = ret;<br>while (*p != '#') p++;<br>q = prvi + n1 - 1;<br>while (*q != '#') q--;<br>strcat(p, drugi); | <b>C)</b> p = ret;<br>while (*p != '#') p++;<br>q = prvi + n1 - 1;<br>while (*(q - 1) != '#') q--;<br>strcat(p, drugi); |
|--|---|---|

6) Šta ispisuje sledeći program napisan na programskom jeziku C, ako se pozove komandom `prog.exe 0 1`?

<pre>#include &lt;stdio.h&gt; #include &lt;stdlib.h&gt; void main(int argc, const char *argv[]) {     int mat[3][3] =         {{4, 3, 2}, {5, 4, 1}, {1, 3, 4}};     int i, j, s, r = 0, p = 0;     r = atoi(argv[1]); p = atoi(argv[2]);</pre>	<pre>for (i = 2; i &gt;= 0; i--){     for (j = s = 0; j &lt; 3; j++)         if (r) s += (*(mat + i) + j);         else s += (*(mat + j) + i);     }     printf("%d", s + p); }</pre>
---	---

- A) 8                                      B) 10                                      (C) 11